

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

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Інтелектуальна система моніторингу дорожнього
трафіку у Smart City»

на здобуття освітнього ступеня бакалавра
зі спеціальності 126 Інформаційні системи та технології
освітньо-професійної програми Інформаційні системи та технології

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

(підпис)

Ім'я, ПРІЗВИЩЕ здобувача

Виконав:

здобувачка вищої освіти гр. ІСД-41

Марія ФЕДОРЧЕНКО

Ім'я, ПРІЗВИЩЕ

Керівник:

науковий ступінь,
вчене звання

Ірина СРІБНА, д. т. н., доцент

Ім'я, ПРІЗВИЩЕ

Рецензент:

науковий ступінь,
вчене звання

Ім'я, ПРІЗВИЩЕ

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут Інформаційних технологій

Кафедра Інформаційних систем та технологій

Ступінь вищої освіти бакалавр

Спеціальність 126 Інформаційні системи та технології

Освітньо-професійна програма Інформаційні системи та технології

ЗАТВЕРДЖУЮ

Завідувач кафедру ІСТ

_____ Каміла СТОРЧАК

«____» _____ 2025 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Федорченко Марія Олександрівна

(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: Інтелектуальна система моніторингу дорожнього трафіку у Smart City

керівник кваліфікаційної роботи Ірина СРІБНА, д. т. н., доцент,

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

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

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

3. Вихідні дані до кваліфікаційної роботи: науково-технічна література за тематикою диплому, відкриті дані досліджень, офіційна документація мов програмування та моделей машинного навчання, офіційна документація технологій IoT.

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

1. Огляд сучасних систем моніторингу дорожнього трафіку

2. Аналіз методів застосування машинного навчання та штучного інтелекту для моніторингу дорожнього трафіку

3. Проектування системи моніторингу трафіку з використанням технологій розумного міста

5. Перелік ілюстративного матеріалу: *презентація*

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

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Підбір науково-технічної літератури	01.03.2025 р.	
2	Дослідження існуючих систем керування транспортним трафіком у розумних містах	15.03.2025 р.	
3	Огляд моделей машинного навчання, які доцільно використовувати для моніторингу трафіку	17.03.2025 р.	
4	Постановка технічного завдання, формування функціональних та нефункціональних вимог до системи, проектування архітектури	4.04.2025 р.	
5	Розробка системи моніторингу дорожнього трафіку, тестування та оптимізація	16.05.2025 р.	
6	Розробка демонстраційних матеріалів	26.05.2025 р.	
7	Попередній захист дипломної роботи	27.05.2025 р.	
8	Подання роботи в деканат	30.05.2025 р.	

Здобувач(ка) вищої освіти

(підпис)

Марія ФЕДОРЧЕНКО

(Ім'я, ПРІЗВИЩЕ)

Керівник
кваліфікаційної роботи

(підпис)

Ірина СРІБНА

(Ім'я, ПРІЗВИЩЕ)

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня бакалавра: 54 стор., 32 рис., 6 табл., 32 джерел.

Мета роботи – аналіз та практична реалізація інтелектуальної системи моніторингу безпеки із використанням технологій машинного навчання та комп'ютерного зору для автоматичного розпізнавання номерних знаків транспортних засобів.

Об'єкт дослідження – процес моніторингу безпеки в умовах транспортної інфраструктури розумного міста.

Предмет дослідження – методи та інструменти машинного навчання і комп'ютерного зору, що застосовуються в системах розпізнавання номерних знаків у режимі реального часу.

Короткий зміст роботи:

У першому розділі проаналізовано проблеми дорожнього трафіку в умовах урбанізації, класифіковано інтелектуальні транспортні системи та визначено переваги технологій комп'ютерного зору.

У другому розділі досліджено технічні засоби реалізації системи (Python, OpenCV, YOLOv4-tiny, Tesseract, EasyOCR), визначено технічні вимоги, обґрунтовано вибір інструментів та виконано моделювання структури.

У третьому розділі реалізовано програмну систему для детекції транспортних засобів і розпізнавання номерів із відеопотоку, проведено її тестування та визначено напрями вдосконалення.

КЛЮЧОВІ СЛОВА: МАШИННЕ НАВЧАННЯ, КОМП'ЮТЕРНИЙ ЗІР, РОЗПІЗНАВАННЯ НОМЕРІВ, СИСТЕМА МОНИТОРИНГУ, OPENCV, YOLO.

ABSTRACT

Textual part of the qualification work for obtaining a bachelor's degree: 54 pages, 32 figs., 6 tables, 32 sources.

The purpose of the work is to analyze and implement an intelligent security monitoring system using machine learning and computer vision technologies for automatic recognition of vehicle license plates.

The object of research is the process of security monitoring in the context of the transport infrastructure of a smart city.

The subject of research is methods and tools of machine learning and computer vision used in real-time license plate recognition systems.

Summary of the work:

The first section analyzes the problems of traffic in urbanization, classifies intelligent transportation systems, and identifies the advantages of computer vision technologies.

The second section investigates the technical means of implementing the system (Python, OpenCV, YOLOv4-tiny, Tesseract, EasyOCR), defines technical requirements, justifies the choice of tools, and performs structure modeling.

In the third section, a software system for vehicle detection and license plate recognition from a video stream is implemented, tested, and areas for improvement are identified.

KEYWORDS: MACHINE LEARNING, COMPUTER VISION, LICENSE PLATE RECOGNITION, MONITORING SYSTEM, OPEN SOURCE, YOLO.

ЗМІСТ

ВСТУП	9
РОЗДІЛ 1 АНАЛІЗ СУЧАСНИХ СИСТЕМ МОНІТОРИНГУ ДОРОЖНЬОГО ТРАФІКУ В УМОВАХ РОЗУМНОГО МІСТА	12
1.1 Концептуальні засади та ключові характеристики розумного міста	12
1.2 Огляд проблематики транспортного менеджменту у сучасних містах ...	16
1.3 Існуючі системи моніторингу заповненості доріг та їх функціональні обмеження.....	25
1.4 Переваги інтеграції інтелектуальних систем у міську інфраструктуру...	30
2 ВИКОРИСТАННЯ НЕЙРОІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ У СИСТЕМАХ РОЗПІЗНАВАННЯ ОБРАЗІВ.....	35
2.1 Ключові поняття та принцип роботи нейромереж.....	35
2.2 Популярні інструменти машинного навчання для розпізнавання зображень.....	41
2.3 Вибір технічних інструментів для створення розумної системи моніторингу автомобілів	48
3 СТВОРЕННЯ СИСТЕМИ МОНІТОРИНГУ ТРАФІКУ НА ДОРОГАХ.....	55
3.1 Постановка задачі та технічні вимоги до системи	55
3.2 Реалізація моделі для розпізнавання номерних знаків	57
3.3 Тестування роботи нейромережі та програмного функціоналу	64
ВИСНОВКИ	69
ПЕРЕЛІК ПОСИЛАНЬ.....	71
ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація)	74

ВСТУП

У сучасних умовах стрімкої урбанізації, зростання кількості транспортних засобів і підвищення навантаження на дорожню інфраструктуру актуалізується необхідність упровадження інтелектуальних систем моніторингу дорожнього трафіку. Автоматизація процесів контролю за рухом транспортних потоків, зокрема через використання технологій комп'ютерного зору та нейромереж, є ключовим напрямом розвитку транспортної інфраструктури в межах концепції «розумного міста». Проблема виявлення та розпізнавання номерних знаків транспортних засобів має велике значення для забезпечення безпеки, покращення транспортного менеджменту, боротьби з порушеннями правил дорожнього руху, а також організації систем автоматизованого доступу та аналітики. В Україні, як і в усьому світі, підвищується попит на високоточні, масштабовані та доступні системи моніторингу, що робить дану тему особливо актуальною з огляду на соціальну, інженерну та інформаційну значущість.

Аналіз останніх наукових досліджень і публікацій свідчить про активне використання глибоких згорткових нейронних мереж (CNN) для задач комп'ютерного зору, зокрема в межах архітектур YOLO, які забезпечують високу швидкодію та точність виявлення об'єктів у реальному часі. Серед відомих рішень у цій сфері слід відзначити проєкти з відкритим кодом, такі як License Plate Recognition YOLOv7, що демонструють практичну ефективність CNN у задачах розпізнавання номерів. Разом із тим, більшість наявних систем орієнтовані на специфічні умови використання або потребують значних обчислювальних ресурсів, що знижує їх доступність для впровадження у малих громадах або локальних мережах відеоспостереження. Таким чином, досі залишаються відкритими питання адаптації подібних систем до змінного освітлення, зниження похибки OCR-модуля, забезпечення стабільної роботи в умовах реального часу та масштабованості рішення. Цим невирішеним аспектам і присвячена дана кваліфікаційна робота.

Метою дипломної роботи є розробка та реалізація автоматизованої системи розпізнавання номерних знаків транспортних засобів на основі глибокого навчання та технологій комп'ютерного зору з використанням Python, OpenCV, YOLO та OCR. Робота спрямована на створення ефективного програмного рішення для моніторингу транспортного трафіку в умовах розумного міста, що забезпечує точне розпізнавання номерів у відеопотоці в режимі квазіреального часу.

У процесі дослідження було здійснено аналіз проблематики транспортного менеджменту в умовах урбанізації та визначено вимоги до систем розпізнавання номерних знаків; розглянуто сучасні інтелектуальні транспортні системи (ITS), їх архітектури, функціональні можливості та обмеження; обґрунтовано вибір технічних інструментів для реалізації програмного модуля, зокрема Python, OpenCV, YOLO та Tesseract/EasyOCR; розроблено архітектуру програмного рішення, що забезпечує виявлення транспортних засобів і розпізнавання номерних знаків у відеопотоці; проведено тестування працездатності створеної системи в умовах, наближених до реального використання, з подальшим аналізом отриманих результатів і визначенням перспектив для подальшого вдосконалення.

Об'єктом дослідження є процес інтелектуального моніторингу дорожнього трафіку в умовах розумного міста.

Предметом дослідження є програмна модель автоматичного розпізнавання номерних знаків транспортних засобів на основі відеопотоку.

У роботі застосовано методи системного аналізу, моделювання, комп'ютерного зору, глибокого навчання, обробки відеоданих, а також порівняльний аналіз ефективності програмних бібліотек для розпізнавання зображень.

Наукова новизна отриманих результатів полягає в створенні модульної системи, яка поєднує алгоритми детекції YOLOv4-tiny з оптичним розпізнаванням Tesseract для забезпечення обробки відеопотоку в режимі реального часу з мінімальними апаратними вимогами. Вперше запропоновано архітектуру, оптимізовану для локального застосування з можливістю масштабування на кілька відеоканалів і збереження результатів на NAS-сервері.

Практична значущість полягає в можливості застосування створеного програмного модуля в муніципальних системах відеоспостереження, на підприємствах із закритим доступом, у платних паркінгах, а також для забезпечення громадської безпеки. Реалізована система може бути легко адаптована для використання в інших проєктах розумної інфраструктури.

Апробація дослідження здійснювалась у формі участі в фахових конференціях:

1. II всеукраїнська науково-технічна конференція «Технологічні горизонти: дослідження та застосування інформаційних технологій для технологічного прогресу України і світу» з поданням тез на тему «застосування штучного інтелекту для вивчення Адаптаційних властивостей сільськогосподарських Культур».

2. VII міжнародна науково-технічна конференція «Сучасний стан та перспективи розвитку IoT» з поданням тез на тему «Використання технологій машинного навчання та комп'ютерного зору в інтелектуальних системах моніторингу безпеки».

Теоретична і методична значущість роботи проявляється у формалізації підходів до побудови модульної архітектури систем комп'ютерного зору з підтримкою обробки в реальному часі та використанням відкритих інструментів Python/OpenCV. Запропонований алгоритм роботи системи може бути адаптований для інших задач виявлення і розпізнавання об'єктів.

Структурно робота складається зі вступу, трьох розділів, висновків та списку використаних джерел. У першому розділі розглянуто теоретичні засади розумного міста та проблематику транспортного менеджменту. Другий розділ присвячено аналізу нейронних мереж і вибору інструментів. У третьому розділі викладено процес реалізації, тестування та вдосконалення системи.

РОЗДІЛ 1 АНАЛІЗ СУЧАСНИХ СИСТЕМ МОНІТОРИНГУ ДОРОЖНЬОГО ТРАФІКУ В УМОВАХ РОЗУМНОГО МІСТА

1.1 Концептуальні засади та ключові характеристики розумного міста

У сучасному світі розумне місто (Smart City) є однією з найактуальніших тем у контексті урбаністичного розвитку. Стрімке зростання інформаційно-комунікаційних технологій (ІКТ), обумовлене прогресом у галузі апаратного і програмного забезпечення, стало основою для підвищення ефективності міських операцій. Різноманітне використання ІКТ для вирішення міських завдань призвело до появи таких термінів, як «кібермісто», «цифрове місто», «електронне місто», «гнучке місто», «інформаційне місто» та «підключене місто». Серед цих визначень термін «розумне місто» має найбільший обсяг і є абстракцією, яка об'єднує всі інші поняття.

Розумне місто, попри широку популярність, залишається концепцією без чіткого й усталеного визначення, що викликає дискусії як серед науковців, так і практиків. У спрощеному вигляді його можна описати як урбаністичний простір, у якому традиційні мережі та послуги стають більш гнучкими, ефективними й стійкими завдяки використанню цифрових і телекомунікаційних технологій. Головною метою є оптимізація міських операцій задля покращення якості життя мешканців, ефективнішого використання ресурсів та зменшення впливу на довкілля. Наприклад, одне з формальних визначень описує розумне місто як простір, що «поєднує фізичну, інформаційну, соціальну та бізнесову інфраструктури для використання колективного інтелекту міста». Інше визначення підкреслює, що розумне місто — це «інноваційне місто, яке використовує ІКТ та інші засоби для підвищення якості життя, ефективності міських операцій і конкурентоспроможності, забезпечуючи задоволення потреб нинішніх і майбутніх поколінь у соціальному, економічному й екологічному аспектах».

Кількість компонентів (рис.1), необхідних для реалізації концепції розумного міста, залежить від доступних технологій та фінансових можливостей. До таких

компонентів можуть належати розумна інфраструктура, розумні будівлі, транспорт, енергетика, охорона здоров'я, управління, освіта та свідомі громадяни. Водночас місто не обов'язково повинно мати всі ці компоненти, щоб вважатися розумним. Наприклад, впровадження окремих рішень, таких як інтелектуальні транспортні системи чи автоматизація енергоспоживання, уже може зробити місто «розумним».

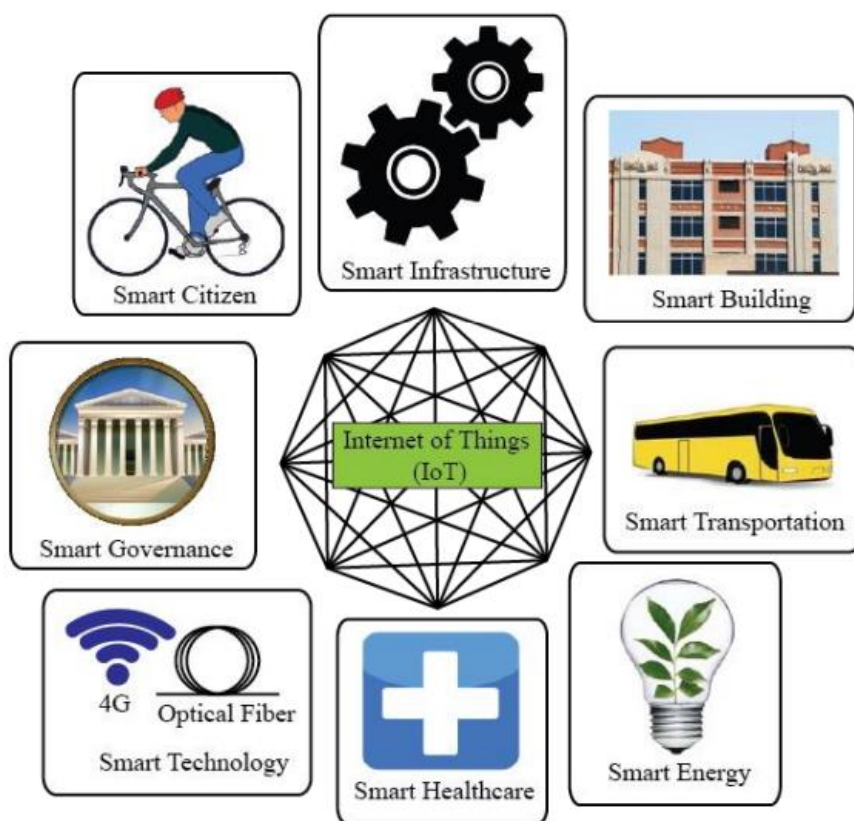


Рис. 1.1 Компоненти розумного міста [1]

Швидке зростання міського населення та рівня урбанізації, що, згідно з прогнозами, досягне 70% до 2050 року, створює серйозні виклики для екології. Сьогодні міста споживають 75% світових ресурсів і енергії та генерують 80% парникових газів, що значно впливає на глобальне потепління. Таким чином, концепція розумного міста стає необхідною стратегією для пом'якшення негативних наслідків урбанізації. Незважаючи на високі витрати, пов'язані з її реалізацією, впровадження рішень розумного міста дозволяє суттєво скоротити енергоспоживання, викиди вуглецю, транспортні витрати та кількість відходів.

Стандартизація відіграє ключову роль у розвитку розумних міст. Міжнародні організації, такі як ISO, пропонують стандарти, які дозволяють забезпечити якість, ефективність і безпеку впроваджуваних рішень. Наприклад, стандарт ISO 37120 включає 100 індикаторів ефективності міст, таких як економіка, освіта, енергетика й навколишнє середовище. Ці індикатори дозволяють порівнювати міста між собою, обмінюватися найкращими практиками та моніторити результати впроваджених рішень. Інші стандарти, такі як IEEE, охоплюють окремі компоненти розумних міст, наприклад, смарт-мережі, інтернет речей (IoT), електронне здоров'я та інтелектуальні транспортні системи.

Розумне місто (Smart City) являє собою комплексну систему, що включає в себе низку важливих характеристик, які працюють разом для покращення якості життя його мешканців та забезпечення сталого розвитку. Ці характеристики (рис.1.2) можна поділити на такі основні: розумна інфраструктура, розумні будівлі, розумний транспорт, розумна енергетика, розумне охорону здоров'я, розумні технології, розумне управління, розумна освіта та розумні громадяни. Усі ці складові є важливими для ефективного функціонування та розвитку розумних міст, і їх рівень впровадження може варіюватися в залежності від стратегічного фокусу конкретного міста.

У цьому контексті важливо виокремити базові характеристики, які визначають концептуальну сутність розумного міста та формують основу для його ефективного функціонування. Вони охоплюють широкий спектр компонентів, що взаємодіють між собою з метою підвищення якості життя, сталого розвитку та цифрової трансформації міського простору. З урахуванням міждисциплінарного підходу, ці характеристики структуруються за ключовими напрямками, які відображають як технологічні, так і соціально-економічні аспекти. Їх узагальнене відображення наведено на рисунку нижче, який демонструє системну взаємопов'язаність елементів розумного міста.

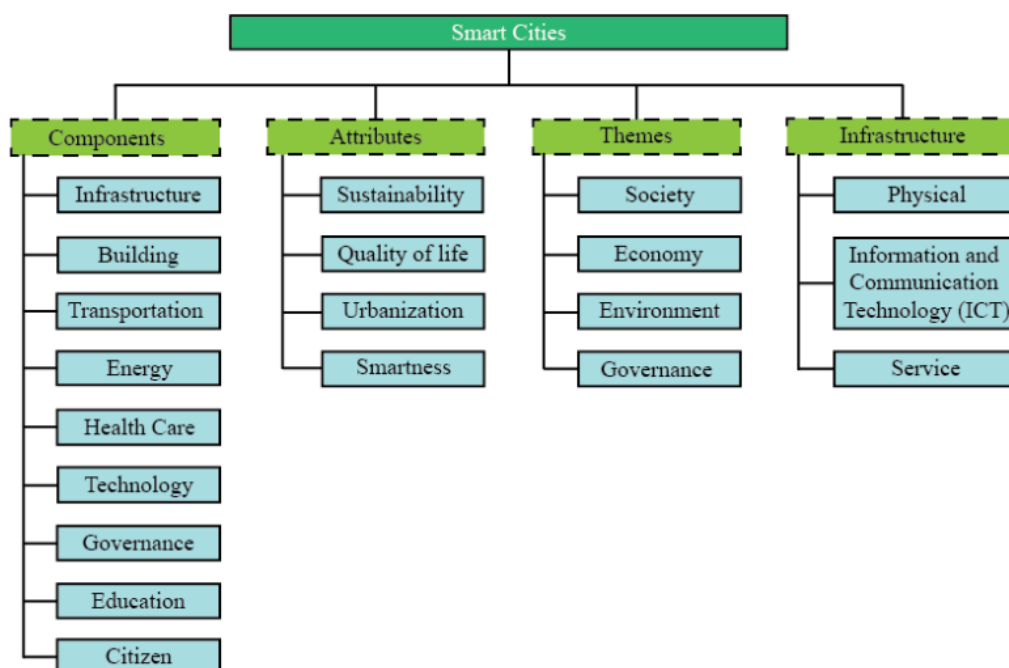


Рис. 1.2 Характеристики розумних міст [1]

Крім компонентів, розумні міста характеризуються певними атрибутами, такими як стійкість, якість життя (Quality of Life, QoL), урбанізація та "розумність". Сійкість міста визначається через його інфраструктуру та управління, енергетику та зміни клімату, боротьбу з забрудненням і відходами, а також через соціальні, економічні та здоров'я аспекти. Якість життя оцінюється через емоційне та фінансове благополуччя громадян. У контексті урбанізації акцент робиться на таких індикаторах, як технології, інфраструктура, управління та економіка. Що стосується "розумності" міста, то вона концептуалізується як прагнення покращити економічні, соціальні та екологічні стандарти міста та його мешканців. Основні складові розумності міста включають розумну економіку, розумних людей, розумне управління, розумну мобільність та розумне життя.

Розумне місто можна розглядати через чотири основні тематичні напрямки: суспільство, економіка, довкілля та управління. Тема суспільства підкреслює, що місто повинно бути орієнтоване на своїх мешканців, забезпечуючи комфорт, доступність послуг та залучення громадян до прийняття рішень. Тема економіки фокусується на здатності міста до сталого економічного зростання, створення нових робочих місць та розвитку інноваційної екосистеми. Тема довкілля

відображає важливість збереження екологічної рівноваги та забезпечення сталого функціонування міста для нинішніх і майбутніх поколінь. Тема управління вказує на здатність міста ефективно адмініструвати політики, інтегрувати різні компоненти у єдину систему та забезпечувати прозорість і підзвітність.

Інфраструктура розумного міста складається з трьох основних елементів: фізичної інфраструктури, інформаційно-комунікаційних технологій (ІКТ) та сервісної інфраструктури. Фізична інфраструктура включає реальні фізичні об'єкти, такі як будівлі, дороги, залізничні колії, лінії електропостачання та водопостачання. Це звичайні, не "розумні" компоненти міста. ІКТ інфраструктура є основним елементом "розумного" компонента міста, оскільки вона забезпечує інтеграцію та функціонування всіх інших компонентів, виконуючи роль нервової системи. Сервісна інфраструктура, побудована на фізичній основі, може включати елементи ІКТ. Прикладом такої інфраструктури є системи масового транспорту або смарт-мережі [1].

Важливим аспектом розумного міста є впровадження систем контролю та безпеки, що забезпечують захист його мешканців та інфраструктури. Такі системи включають мережу камер відеоспостереження, датчики руху, контролери доступу до громадських зон, а також алгоритми на основі штучного інтелекту для моніторингу та аналізу потенційних загроз. Це дозволяє своєчасно реагувати на надзвичайні ситуації, запобігати злочинам і забезпечувати стабільну роботу міста. Розвинуті системи безпеки не лише підвищують рівень комфорту життя, але й створюють довіру до технологій, які інтегровані в міське середовище.

1.2 Огляд проблематики транспортного менеджменту у сучасних містах

Сучасні міста стикаються з численними викликами у сфері транспортного управління, що обумовлені як зростанням інтенсивності дорожнього руху, так і структурною недостатністю міської інфраструктури. Для комплексного розуміння масштабів і природи цих проблем доцільно виокремити основні аспекти, які формують поточну ситуацію в транспортному секторі. Узагальнене представлення

ключових проблем транспортного менеджменту наведено на рисунку, який дозволяє візуалізувати основні напрями дисфункцій, що потребують негайного вдосконалення з боку органів міського управління та систем інтелектуального контролю.

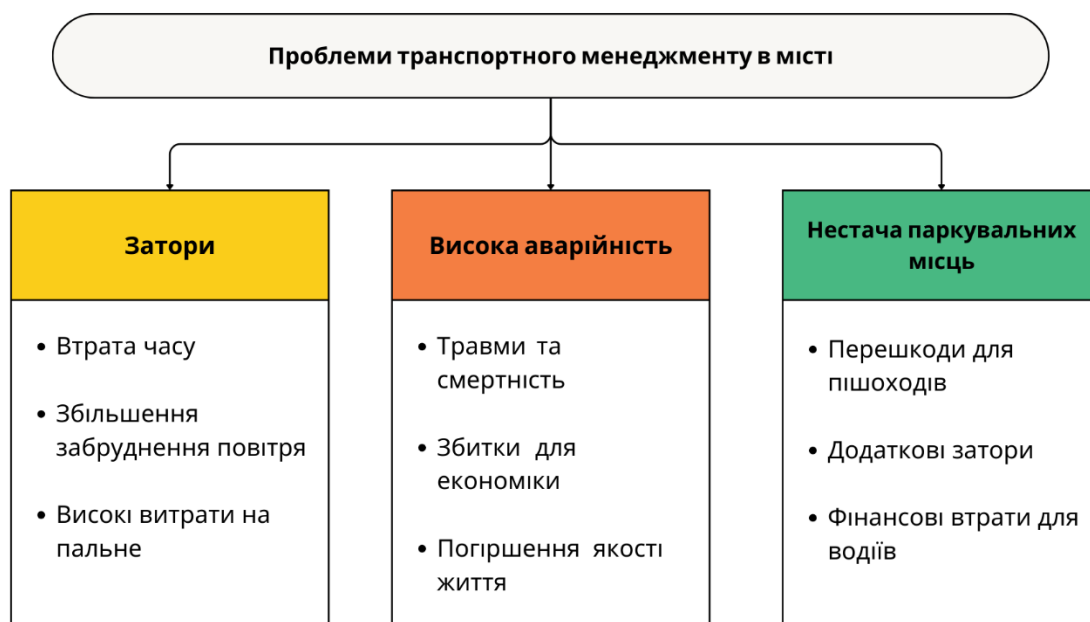


Рис. 1.3 Проблематика транспортного менеджменту.

Транспортний менеджмент є однією з ключових складових сучасного міського управління, яка впливає на соціально-економічний розвиток, екологічну стабільність та загальну якість життя мешканців. У контексті стрімкої урбанізації та зростання населення міст, ефективне управління транспортними системами стає все більш актуальним завданням. Проблематика транспортного рис (1.3) менеджменту зосереджується на подоланні основних викликів, зокрема заторів, високого рівня аварійності та нестачі паркувальних місць. Недосконалість існуючих підходів у цій сфері призводить до збільшення витрат часу та ресурсів, погіршення екологічної ситуації та зниження рівня комфорту життя громадян.

Однією з ключових передумов виникнення цих проблем є дисбаланс між інтенсивним зростанням кількості транспортних засобів та недостатнім розвитком міської інфраструктури. Урбаністичні простори багатьох міст не адаптовані до зростаючого транспортного навантаження, що спричиняє нераціональне

використання дорожнього простору, уповільнення руху та перевантаження магістралей. Відсутність комплексного планування транспортних потоків та недосконалість систем управління дорожнім рухом додатково ускладнюють ситуацію, сприяючи формуванню постійних заторів, які, у свою чергу, мають негативний вплив на економіку, екологію та соціальну сферу.

Транспортні затори є однією з найбільш актуальних проблем сучасних міст, що має значний вплив на економіку, екологію та соціальну сферу. В Україні спостерігається тенденція до зростання кількості автомобілів, яка супроводжується недостатнім темпом розвитку дорожньої інфраструктури. За даними Державної служби статистики України, з кожним роком кількість асфальтованих доріг збільшується непропорційно малими темпами, що спричиняє зростання заторів на транспортних шляхах. Аналогічна ситуація спостерігається і в глобальному масштабі, де значні фінансові втрати виникають через простой транспорту в заторах. Наприклад, економічні втрати(табл.1.1)[2] від заторів у США склали 305 мільярдів доларів, тоді як у Великій Британії — 52,01 мільярда доларів. Серед найбільш постраждалих міст виділяються Нью-Йорк, Лос-Анджелес, Маніла та інші.

Таблиця 1.1

Економічні втрати через затори у найбільш завантажених регіонах світу

Area	Losses, bln. dollars
USA	305
UK	52,01
NYC	33,7
Los Angeles	19,2
Manila	18,615
Bangladesh	11,4

Продовження таблиці 1.1

Area	Losses, bln. dollars
San Francisco	10,6
Atlanta	7,1
Jakarta	5
Dhaka	4,463
Greater Toronto and Hamilton Area (GTHA)	3,3

Затори на дорогах супроводжуються цілою низкою проблем, включаючи підвищений рівень стресу серед мешканців, зниження продуктивності підприємств через запізнення працівників, а також підвищене споживання пального і, як наслідок, зростання шкідливих викидів. Як показують дані(табл.1.2)[2], у Києві протягом 2009–2018 років кількість автомобілів зросла з 906 тисяч до 1,06 мільйона, що призвело до значного збільшення викидів вуглецю (з 6064 тонн у 2009 році до 8692 тонн у 2018 році), оксидів азоту, сірчаного ангідриду та інших шкідливих речовин.

Таблиця 1.2

Статистичні дані викидів транспортних засобів у Києві

Year	Emissions from traffics in Kyiv, t							Automobiles in Kyiv, thousand units
	CO	Hydrocarbons	Nitrogen oxides	Sulfuric acid anhydride	Aldehydes	Soot	Plumbum	
2009	6064	340	407	117	22	71	7	906
2010	6312	354	424	122	23	74	7	936
2011	6569	369	441	127	23	77	7	951
2012	6838	384	459	132	24	82	8	1026
2013	7117	399	478	138	25	84	8	1064
2014	7407	415	497	143	26	87	8	1055
2015	7709	432	518	149	28	91	9	1048
2016	8024	450	539	155	29	94	9	1051
2017	8352	468	561	161	30	98	9	1052
2018	8692	488	584	168	31	102	10	1060

Затори також мають суттєвий економічний вплив. За даними(табл.1.3), середній водій у Києві у 2018 році витратив близько 350 годин на рік у заторах, що еквівалентно втраті понад 28 350 гривень у вигляді упущеної вигоди, якщо враховувати середню погодинну зарплату в місті. Крім того, додаткові витрати на пальне склали приблизно 9296 гривень для автомобілів на бензині та 6657 гривень для дизельних автомобілів.[2]

Таблиця 1.3

Грошовий вираз втрат від заторів залежно від виду пального (на одного водія)

The monetary expression of losses from traffic jams depending on fuel per 1 motorist				
Fuel	Time spent in jams, hours/year	1 hour expenditure in jams, liters	Cost, UAH/1 liter	Total, UAH
Gasoline	350	0.8	33,2	9296
Diesel	350	0.6	31,7	6657

Однією з ключових проблем транспортного менеджменту є висока частота транспортних аварій і пов'язаних із ними порушень правил дорожнього руху. Водіння у стані алкогольного сп'яніння, перевищення швидкості, проїзд на червоне світло, незаконні маневри та інші порушення щорічно призводять до численних дорожньо-транспортних пригод, що супроводжуються значними людськими втратами та травмами. Зростання кількості автомобілів на дорогах у поєднанні з недостатньо ефективними заходами безпеки створює умови, що сприяють збільшенню аварійності. Це не лише негативно впливає на загальну безпеку дорожнього руху, а й ускладнює управління транспортними потоками, збільшуючи затримки та навантаження на інфраструктуру.

Аналіз статистичних даних за 2013–2023 роки свідчить про значний рівень транспортних аварій та пов'язаних із ними жертв у національному масштабі. За цей період було зафіксовано понад 400 тисяч дорожньо-транспортних пригод категорій А1 і А2, що супроводжувались щорічно в середньому 1751 смертельним випадком та 442601 травмою. Як показано на діаграмі (рис. 1.4), кількість жертв, що траплялися протягом перших 24 годин після аварії, зменшувалася з 2013 по 2018

роки, але почала зростати з 2019 року. Ця тенденція свідчить про необхідність впровадження ефективних заходів безпеки для зменшення ризиків на дорогах.[3]

З огляду на стрімке зростання інтенсивності руху та кількості транспортних засобів, проблема аварійності на дорогах залишається одним із найгостріших викликів для транспортного менеджменту. Поширені порушення правил дорожнього руху, як-от перевищення швидкості, керування у нетверезому стані чи проїзд на заборонний сигнал світлофора, прямо впливають на рівень дорожньо-транспортних пригод. Для обґрунтованого формування заходів безпеки та удосконалення політики в галузі управління трафіком доцільним є аналіз статистичних даних щодо аварійності та кількості порушень. У цьому контексті доцільним є звернення до узагальненої динаміки дорожньо-транспортних пригод і зареєстрованих порушень за останнє десятиріччя, що дозволяє ілюструвати масштаби проблеми та її стійке загострення у часовій перспективі.

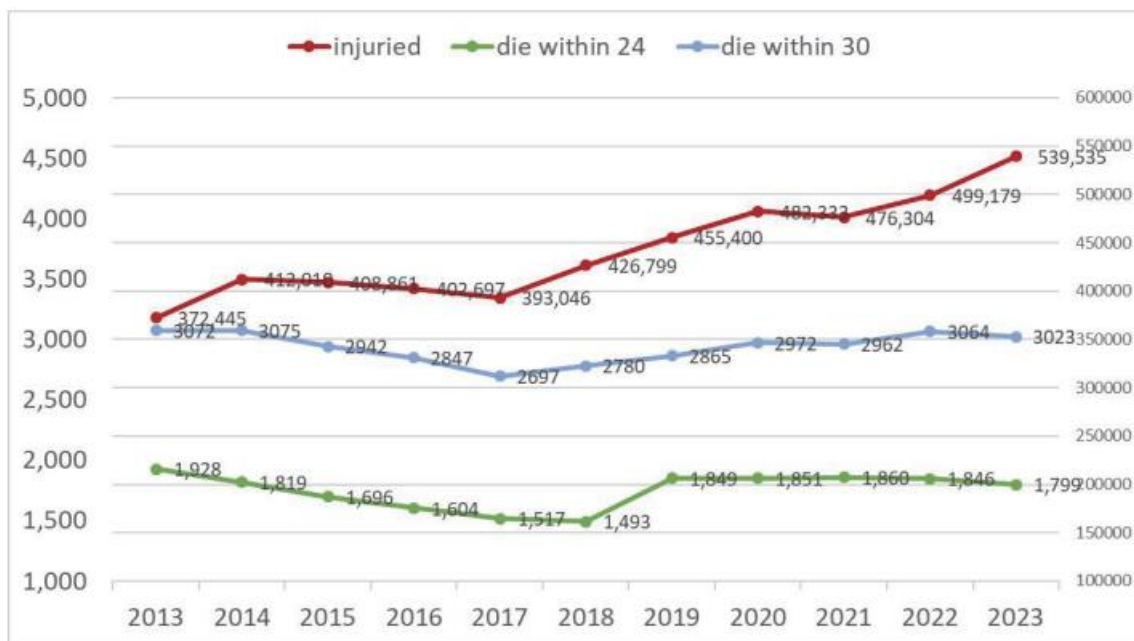


Рис. 1.4 Динаміка дорожньо-транспортних пригод, постраждалих і травм з 2013 по 2023 рік [3]

Щороку в період із 2013 по 2023 роки в середньому реєструвалося 11,92 мільйона порушень правил дорожнього руху. За даними (рис. 1.5), кількість

порушень зростала стабільно протягом усього періоду, збільшившись із 8,05 мільйона в 2013 році до 17,55 мільйона в 2023 році. Така динаміка свідчить про необхідність посилення контролю за дотриманням правил дорожнього руху.[3]

Окрім підвищеного рівня аварійності, додатковим системним викликом для міської транспортної інфраструктури є проблема обмеженої кількості паркувальних місць. Існуючий просторовий ресурс більшості урбанізованих територій не здатен задовольнити попит з боку власників приватного транспорту, що призводить до несанкціонованого паркування, перешкоджання руху, зниження пропускної здатності доріг та підвищення ризиків аварійних ситуацій. Вивчення закономірностей попиту на паркування упродовж доби дозволяє виявити пікові навантаження на інфраструктуру та сформулювати оптимальні підходи до планування та регулювання паркувального простору. Відповідні графічні дані ілюструють залежність між добовою активністю та середнім часом, витраченим на пошук місця, що є важливим для підвищення ефективності транспортного планування.

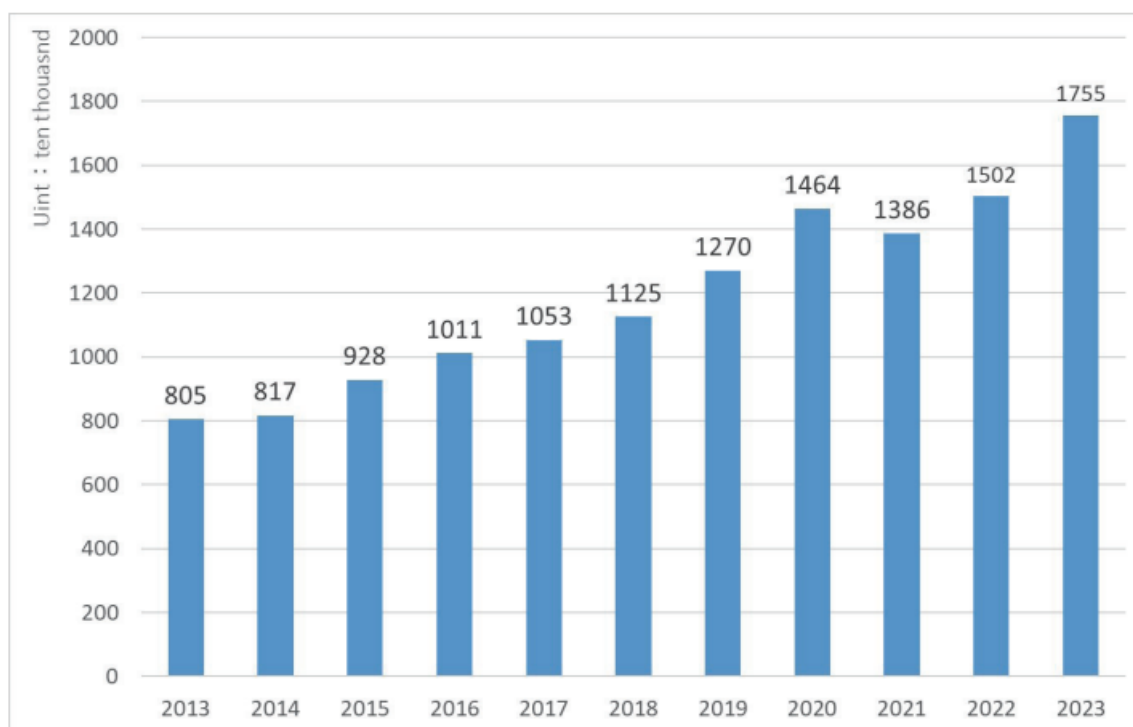


Рис. 1.5 Порушення правил дорожнього руху з 2010 по 2023 рік [3]

Ще однією важливою проблемою є обмежений простір для паркування в сучасних містах. Зростання урбанізації та кількості приватних автомобілів значно перевищує можливості міської інфраструктури, що призводить до перевантаження вуличних паркувальних місць і виникнення хаотичного паркування. Водії нерідко залишають автомобілі у невстановлених місцях, блокуючи смуги руху та зменшуючи пропускну здатність доріг. Це, у свою чергу, сприяє зростанню заторів, зниженню ефективності транспортних систем і погіршенню якості життя міських мешканців.

На основі представлених даних графіка (рис1.5), що відображає попит на паркування та середній час, витрачений на пошук місця і подальшу пішохідну прогулянку, видно, що попит на паркування змінюється протягом доби. Максимальний попит на паркування спостерігається у пікові години між 8:00 та 18:00, що відповідає основним робочим і діловим активностям мешканців міст. У ці періоди середній час, витрачений на пошук місця для паркування, також зростає, досягаючи найвищих показників у ранковий та вечірній час, що створює додаткові незручності для водіїв.

Одним із важливих аспектів сталого міського розвитку є ефективне управління паркувальними ресурсами. У сучасних містах зростання кількості транспортних засобів значно перевищує темпи розширення інфраструктури, що зумовлює дефіцит місць для стоянки. Це спричиняє неорганізоване паркування, ускладнює дорожній рух, знижує його пропускну здатність і створює додаткові затримки. У сукупності такі явища погіршують транспортну ситуацію та комфорт проживання в містах. Рациональне використання паркувального простору стає пріоритетом у системах міського управління.

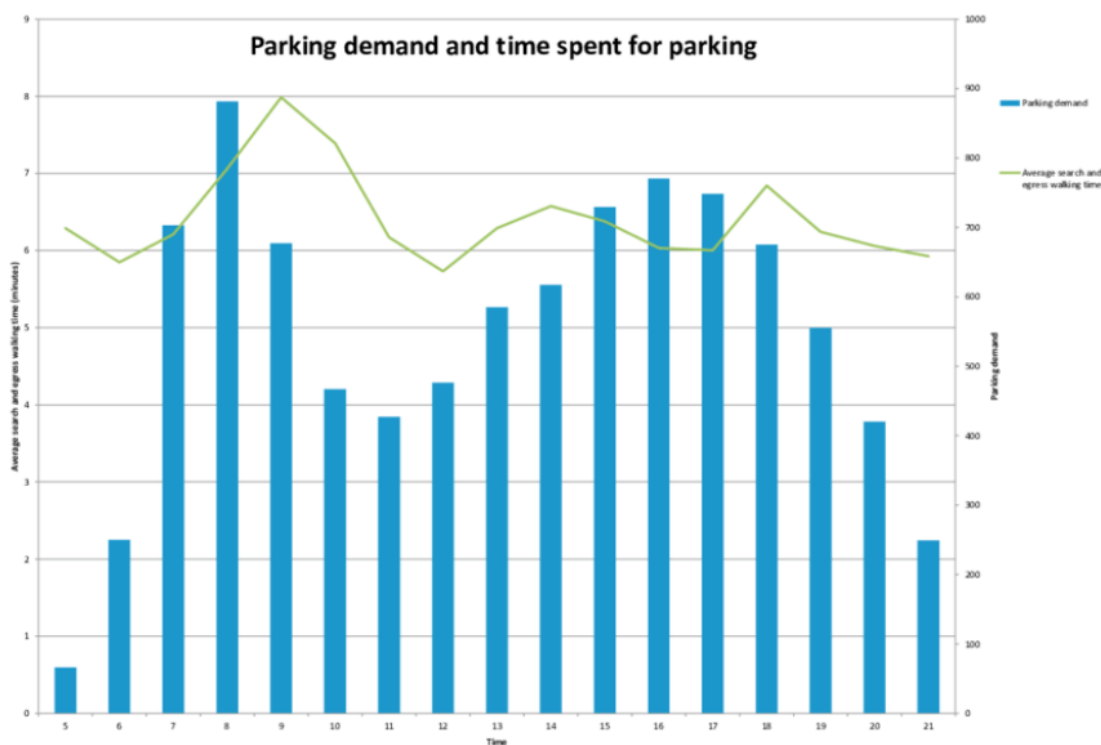


Рис. 1.6 Попит на паркування та середній час, витрачений на пошук паркінгу та виїзд протягом дня [4]

Аналіз даних свідчить, що найбільше навантаження на паркувальну інфраструктуру припадає на період з 7:00 до 9:00 ранку та з 15:00 до 17:00 вечора. У цей час попит на паркувальні місця сягає свого піку, а середній час пошуку місця для паркування може досягати понад 8 хвилин. Увечері цей час поступово скорочується, коли загальна активність мешканців знижується. Така тенденція демонструє важливість ефективного управління паркувальними зонами, особливо у ділових районах міст, і необхідність впровадження сучасних інтелектуальних систем паркування, які допоможуть зменшити затримки та оптимізувати використання паркувальних площ.

Сучасні міста стикаються з низкою транспортних проблем, серед яких затори, високий рівень аварійності та нестача паркувальних місць. Зростання кількості автомобілів перевищує можливості інфраструктури, що посилює ці виклики. Ефективним рішенням може стати впровадження систем моніторингу трафіку, які в реальному часі аналізують транспортні потоки, прогнозують затори та оптимізують рух. Використання інтелектуальних датчиків, камер і GPS-трекерів

дозволяє зменшити затримки, знизити аварійність і покращити використання міського простору. Інтеграція таких технологій сприятиме сталому розвитку транспортної інфраструктури та підвищенню якості життя в містах.

1.3 Існуючі системи моніторингу заповненості доріг та їх функціональні обмеження

Інтелектуальні системи управління дорожнім рухом (Intelligent Traffic Management Systems, ITMS) (рис.1.7) відіграють важливу роль у сучасному управлінні транспортними потоками, сприяючи зниженню рівня заторів, підвищенню безпеки на дорогах та ефективному використанню дорожньої інфраструктури. Впровадження таких систем базується на комплексному використанні інноваційних технологій, зокрема датчиків, камер відеоспостереження, інтелектуального аналізу даних та комунікаційних мереж, що дозволяє здійснювати безперервний контроль дорожньої ситуації в режимі реального часу.

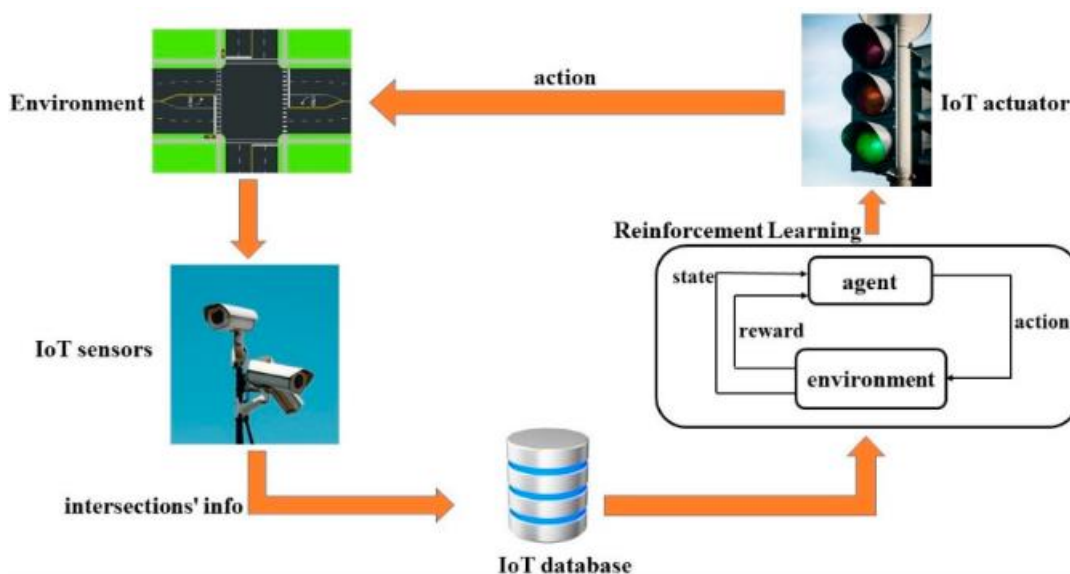


Рис. 1.7 Інтелектуальні системи управління дорожнім рухом[5]

Сучасні ITMS інтегрують сенсорні та комунікаційні технології, механізми контролю та аналізу даних у режимі реального часу. Одним із ключових елементів таких систем є розумне відеоспостереження, що забезпечує детальний моніторинг дорожньої ситуації. Камери спостереження, оснащені штучним інтелектом та комп'ютерним зором, здатні розпізнавати транспортні засоби, пішоходів, виявляти порушення правил дорожнього руху та прогнозувати потенційні аварійні ситуації.

ITMS виконують низку ключових функцій, зокрема збір даних про поточний стан руху, його інтенсивність, швидкість транспортних засобів, аналіз отриманої інформації для виявлення закономірностей та прогнозування заторів, інформування учасників дорожнього руху про можливі маршрути, дорожні роботи та аварійні ситуації. Завдяки інтеграції штучного інтелекту та автоматизованих систем управління можливе динамічне регулювання руху, що включає зміну світлофорних циклів, регулювання в'їзду на автомагістралі та оптимізацію маршрутів.

Запропонована архітектура інтелектуальної системи управління трафіком (рис.1.8) передбачає використання Low Power Wide Area Networks (LPWAN) та механізмів машинного навчання. Одним із найбільш перспективних рішень у цій сфері є технологія LoRa (Long Range Wide Area Network), що забезпечує енергоефективний та довготривалий зв'язок між сенсорами та хмарною платформою. Використання LoRa дозволяє знизити витрати на підтримку інфраструктури та забезпечити безперервний збір даних навіть у віддалених або густонаселених районах.

З огляду на зазначені функціональні можливості, доцільно розглянути типову архітектуру інтелектуальної системи управління дорожнім рухом, яка поєднує енергоефективні сенсорні мережі та хмарні обчислення для обробки потоків даних у режимі реального часу. Така архітектура (рис. 1.7) базується на використанні Low-Power Wide Area Networks, зокрема технології LoRa, що забезпечує надійний довготривалий зв'язок між розподіленими IoT-датчиками й центральною платформою аналізу. Інтеграція LPWAN-комунікацій із модулями машинного навчання та інтелектуального відеоспостереження створює єдине середовище,

здатне безперервно збирати, обробляти й інтерпретувати дані про дорожню ситуацію, тим самим підвищуючи адаптивність і ефективність транспортної інфраструктури.

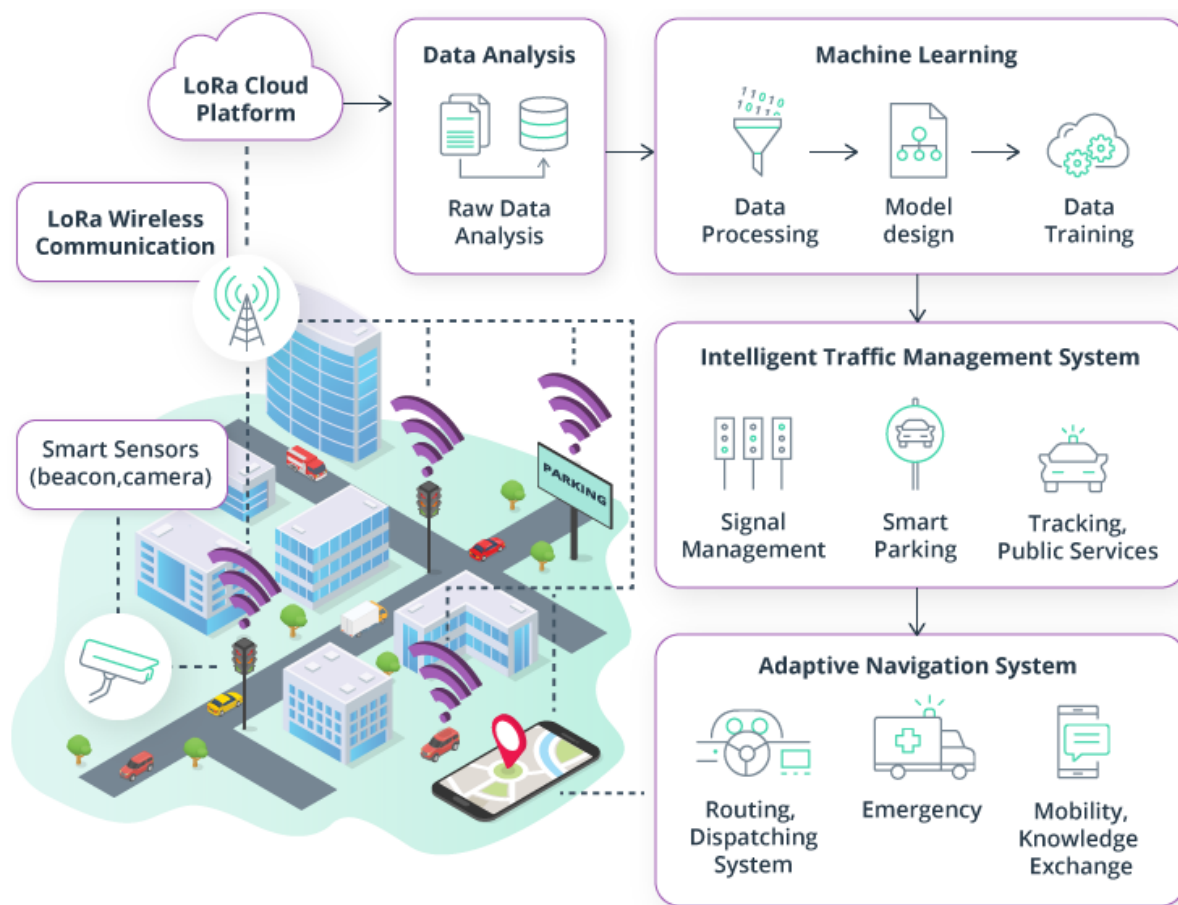


Рис. 1.8 Огляд інтелектуальної системи управління дорожнім рухом на основі технології LoRa [6]

Зібрані розумними датчиками дані передаються на хмарну платформу для обробки та аналізу. Використовуючи алгоритми машинного навчання, система здатна прогнозувати затори, визначати аварійні ситуації та автоматично налаштовувати параметри дорожнього руху для підвищення його ефективності. Одним із ключових елементів такої системи є механізм логістичної регресії, який дозволяє моделювати дорожні умови та пропонувати оптимальні маршрути в режимі реального часу. [7]

Основними компонентами(табл.1.4) [6] інтелектуальних систем управління дорожнім рухом є апаратне та програмне забезпечення, що працює в комплексі. До апаратної частини входять IoT-датчики, зокрема RFID-мітки (радіочастотна ідентифікація) та системи автоматичного збору даних, датчики температури та якості повітря, а також підключені камери відеоспостереження. Програмне забезпечення включає обчислювальні технології хмарного та граничного (edge) обчислення, платформи для зберігання та аналізу трафіку, геоінформаційні системи (GIS), а також додатки для управління транспортом.

Таблиця 1.4

Основні елементи інтелектуальної системи руху

Апаратне забезпечення (Hardware)	Програмне забезпечення (Software)
IoT-датчики на дорогах, зокрема: а) RFID або AIDC теги; б) Датчики температури.	Хмарні обчислення та обробка на периферії: а) Платформи/сховища транспортних даних; б) Хмарні системи керування трафіком; с) Геоінформаційні системи (GIS);
Датчики якості повітря	Усі транспортні застосунки
Підключені камери спостереження (CCTV)	Аналітика великих даних і прогнозне моделювання
Електронні пункти оплати / системи ціноутворення на дорогах	Послуги на основі місцезнаходження
Підключені системи світлофорів	AI/ML (штучний інтелект та машинне навчання), комп'ютерний зір, OCR: а) Комп'ютерний зір; б) Оптичне розпізнавання символів (OCR); с) Навчання з підкріпленням.

Інтелектуальні світлофорні системи, великі дані та предиктивна аналітика відіграють ключову роль у регулюванні дорожнього руху. Використання штучного

інтелекту та машинного навчання, зокрема комп'ютерного зору, оптичного розпізнавання символів (OCR) та методів навчання з підкріпленням, дозволяє значно підвищити ефективність управління транспортними потоками. Додаткові компоненти, такі як системи електронного збору плати за проїзд, геолокаційні сервіси та пристрої граничних обчислень, забезпечують швидку обробку даних та ефективну координацію руху. Таким чином, розширення та вдосконалення ITMS сприяє не лише підвищенню ефективності транспортної мережі, а й створенню більш безпечного та екологічно чистого міського середовища.

Інтелектуальні системи управління дорожнім рухом (ITMS) мають значний потенціал для оптимізації транспортних потоків, проте їх впровадження та експлуатація супроводжуються низкою обмежень та викликів. Однією з ключових проблем є висока вартість розгортання та підтримки такої інфраструктури. Інтеграція IoT-датчиків, камер відеоспостереження, обчислювальних потужностей та комунікаційних мереж вимагає значних фінансових вкладень, що може бути непосильним для малих міст або країн з обмеженими бюджетними ресурсами.

Окрім фінансових бар'єрів, критичним фактором є безпека та захист даних. Оскільки ITMS використовують великі обсяги даних у реальному часі, вони стають вразливими до кібератак, що може призвести до несанкціонованого втручання у роботу системи та навіть повного виходу з ладу окремих її компонентів. Захист персональних даних водіїв та пасажирів також є викликом, оскільки зібрана інформація про маршрути пересування, швидкість транспортних засобів та інші параметри може використовуватися для ідентифікації осіб, що викликає питання конфіденційності.

Ще одним обмеженням є складність інтеграції ITMS із застарілою інфраструктурою міського транспорту. У багатьох містах дорожня мережа та світлофорні системи були спроектовані без урахування сучасних цифрових технологій, що ускладнює їх модернізацію. Крім того, відсутність єдиних стандартів у сфері інтелектуального управління транспортом створює проблеми сумісності різних компонентів системи, що може призводити до неефективного функціонування.

Фактор людського сприйняття також впливає на ефективність ITMS. Водії та пішоходи не завжди готові довіряти автоматизованим системам управління дорожнім рухом, що може спричиняти труднощі у впровадженні нових технологій. Також поведінка учасників дорожнього руху є динамічною і не завжди піддається точному прогнозуванню навіть за допомогою передових методів машинного навчання та штучного інтелекту.

Нарешті, функціональність ITMS значною мірою залежить від якості та доступності зв'язку. Впровадження таких технологій, як LPWAN та LoRa, дозволяє забезпечити стабільний зв'язок між елементами системи, проте у зонах зі слабким покриттям або нестабільною мережею ефективність роботи ITMS може суттєво знижуватися. Також слід враховувати ризики, пов'язані зі збоєм у роботі апаратного або програмного забезпечення, що може призвести до некоректного функціонування системи та створити додаткові загрози безпеці дорожнього руху.

Таким чином, впровадження інтелектуальних систем управління дорожнім рухом супроводжується низкою викликів, включаючи фінансові витрати, питання кібербезпеки, проблеми інтеграції з існуючою інфраструктурою, людський фактор та технічні обмеження. Незважаючи на ці труднощі, подальший розвиток технологій, зокрема вдосконалення алгоритмів штучного інтелекту, впровадження більш надійних комунікаційних мереж та створення уніфікованих стандартів, сприятиме підвищенню ефективності та масштабованості таких систем. Це дозволить значно покращити керованість транспортними потоками, знизити рівень заторів та підвищити безпеку дорожнього руху, що є ключовими пріоритетами сучасних урбаністичних стратегій.

1.4 Переваги інтеграції інтелектуальних систем у міську інфраструктуру

У контексті зростаючих викликів урбанізації впровадження інтелектуальних транспортних систем (ITS) стає важливою складовою модернізації міської інфраструктури. Завдяки комплексному використанню цифрових технологій,

автоматизованого аналізу даних та штучного інтелекту, такі системи здатні суттєво підвищити ефективність управління дорожнім рухом. До основних переваг інтеграції ITS належать підвищення безпеки на дорогах, зменшення заторів, скорочення викидів шкідливих речовин, оптимізація маршрутів та підтримка стратегічного планування міського простору. Ці переваги (рис.1.9) створюють основу для формування сталої, адаптивної та технологічно розвиненої транспортної системи.

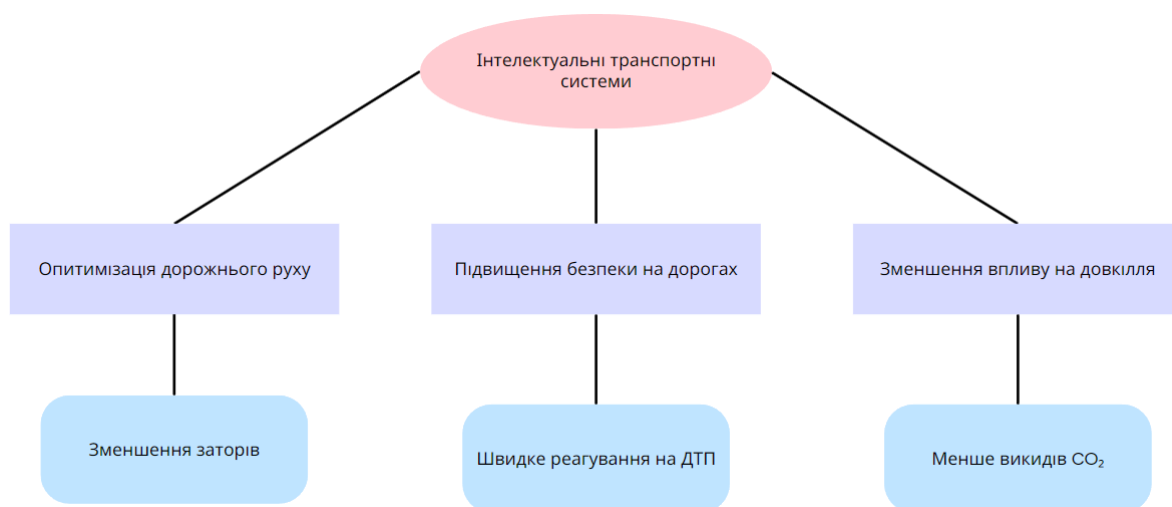


Рис. 1.9 Переваги інтеграції інтелектуальних транспортних систем

Інтеграція інтелектуальних систем у транспортну інфраструктуру є важливим інструментом для забезпечення сталого розвитку міст та підвищення ефективності управління дорожнім рухом. Однією з ключових переваг таких систем є здатність відстежувати рух транспортних засобів у режимі реального часу, що дозволяє отримувати актуальні дані про трафік та оперативно реагувати на його зміни. На основі цих даних інтелектуальні алгоритми аналізують завантаженість доріг та адаптують режими роботи світлофорів, забезпечуючи пріоритет для основних транспортних потоків або громадського транспорту. Це сприяє мінімізації часу очікування на перехрестях, зменшенню кількості заторів та більш рівномірному розподілу транспортних потоків у межах міської мережі. Унаслідок цього відбувається скорочення середньої тривалості поїздок, що позитивно впливає

як на водіїв, так і на пасажирів, сприяючи підвищенню рівня комфорту та мобільності населення.

Окрім оптимізації дорожнього руху, інтеграція інтелектуальних транспортних систем має значний екологічний та економічний ефект. Зниження кількості заторів призводить до зменшення споживання пального, а отже, і до скорочення викидів парникових газів та інших забруднювальних речовин в атмосферу. Це особливо актуально для великих мегаполісів, де транспорт є одним із головних джерел забруднення повітря.

Інтелектуальні транспортні системи відіграють ключову роль у виявленні та запобіганні нещасним випадкам завдяки безперервному моніторингу дорожньої ситуації. Використання датчиків, камер і алгоритмів аналізу дозволяє оперативно виявляти аварії, небезпечні маневри або інші ризики на дорозі. Швидка реакція на такі інциденти сприяє зменшенню кількості жертв і мінімізує ризик подальших аварій. Окрім цього, системи аналізують поведінку водіїв, визначаючи фактори, що можуть призвести до небезпечних ситуацій, тим самим запобігаючи виникненню аварійних випадків ще на етапі їхнього зародження.

Інтелектуальні системи значно підвищують ефективність роботи правоохоронних органів у сфері забезпечення дотримання правил дорожнього руху. Технології відеоспостереження та автоматичного розпізнавання номерних знаків дозволяють виявляти порушення, такі як перевищення швидкості, проїзд на червоне світло або рух смугою громадського транспорту. Це дає змогу правоохоронцям оперативно реагувати на порушення та накладати відповідні штрафи, що не лише підвищує рівень дисципліни серед водіїв, але й сприяє зниженню аварійності на дорогах.

Важливим аспектом використання інтелектуальних транспортних систем є сприяння ефективному міському плануванню. Аналіз даних про рух транспорту, пікові години навантаження та маршрути дозволяє міським планувальникам ухвалювати обґрунтовані рішення щодо будівництва нових доріг, реконструкції існуючих транспортних розв'язок або розвитку громадського транспорту. Завдяки

цьому створюється більш збалансована та ефективна транспортна інфраструктура, яка відповідає потребам населення та сприяє зменшенню заторів у майбутньому.

Обмежений простір для паркування є однією з актуальних проблем сучасних міст, і інтеграція інтелектуальних систем допомагає вирішувати це питання. Завдяки датчикам і камерам, що контролюють наявність вільних паркувальних місць у режимі реального часу, водії можуть швидко знаходити місце для паркування, знижуючи рівень заторів, спричинених пошуком паркувального простору. Окрім цього, зібрані дані дозволяють міським адміністраціям аналізувати потребу в додаткових паркувальних зонах та планувати їхній розвиток відповідно до реальних потреб населення.

Для оцінки ефективності різних методів управління транспортними потоками було проведено аналіз практичного застосування ITS у кількох мегаполісах світу. Наведені дані демонструють зміни в обсягах трафіку, часу подорожі та популярності громадського транспорту після впровадження інтелектуальних систем управління.

Таблиця 1.5

Вплив ITS на трафік і транспортні показники мегаполісів

Місто	Зміни у транспортному потоці (%)	Скорочення часу подорожі (%)	Збільшення використання громадського транспорту (%)
Лондон	-16; -30	-21; -30	+25; +15; +49; +18
Сінгапур	-44; -10; -15	20–30 км/год; 45–65 км/год	N/A
Стокгольм	-20	-33	+5
Мілан	-34; -49	-17; +7; +4.7	N/A
Гетеборг	-10; -2.5	-10; -20	+6
Рим	-20	+4; +5	+5

Як видно з табл.1.5 [7], впровадження інтелектуальних систем управління рухом дозволяє суттєво зменшити затори та покращити швидкість транспортних

засобів. Успішність таких ініціатив залежить від регулювання вартості проїзду, технічної інфраструктури та державної політики у сфері транспорту.

Інтеграція інтелектуальних систем у транспортну інфраструктуру міст має численні переваги, зокрема оптимізацію трафіку, зменшення заторів, підвищення рівня безпеки та покращення екологічної ситуації. Аналіз практичного застосування цих технологій у різних країнах показує їхню ефективність у скороченні часу поїздок та збільшенні використання громадського транспорту. Подальший розвиток та впровадження таких систем сприятиме формуванню сталого міського середовища та підвищенню якості життя мешканців.

2 ВИКОРИСТАННЯ НЕЙРОІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ У СИСТЕМАХ РОЗПІЗНАВАННЯ ОБРАЗІВ

2.1 Ключові поняття та принцип роботи нейромереж

Штучні нейронні мережі (ШНМ) — це обчислювальні моделі, натхненні біологічними нейронними структурами, які здатні навчатися на основі вхідних даних і виконувати завдання класифікації, прогнозування та розпізнавання. У своїй основі ШНМ моделюють елементарні функціональні одиниці мозку — нейрони — і взаємозв'язки між ними. Вони імітують процес обробки сигналів, як це відбувається у нервовій системі людини, але в математично спрощеній формі.[8]

Концепція нейронних мереж була вперше формалізована у 1943 році Ворреном МакКаллоком і Волтером Піттсом, які розробили логічну модель нейрона. Ця модель описувала нейрон як бінарний пристрій, що активується, якщо зважена сума вхідних сигналів перевищує певний поріг. Цей підхід став фундаментальним для формування поняття про нейронні мережі як системи обробки інформації, що базуються на простих обчислювальних елементах і логічних операціях [9].

Пізніше, з розвитком кібернетики та інформатики, нейронні мережі еволюціонували в потужний інструмент машинного навчання. Як зазначає Саймон Гайкін [Haykin, 1999], нейронна мережа — це адаптивна система, яка змінює свою структуру на основі зовнішніх або внутрішніх інформаційних потоків. Основними властивостями ШНМ, що забезпечують їхню ефективність, є здатність до узагальнення, паралельність обчислень, відмовостійкість, а також здатність до самоорганізації та навчання з прикладів [8].

Завдяки цим характеристикам ШНМ широко застосовуються у сферах, де класичні алгоритми мають обмеження: це розпізнавання образів і мовлення, обробка природної мови, фінансове прогнозування, діагностика у медицині, контроль складних технічних систем і багато інших прикладних задач.

Архітектура штучної нейронної мережі визначає спосіб організації обчислювальних елементів і схему взаємодії між ними. Центальною складовою є штучний нейрон, що моделює біологічний нейрон у спрощеному вигляді. Він приймає множину входних сигналів, кожен з яких асоціюється з відповідним ваговим коефіцієнтом, підсумовує ці входні значення разом зі зміщенням (bias), а потім застосовує до результату функцію активації, що додає системі нелінійності. Всі нейрони організовані у шари: входний шар приймає зовнішню інформацію, приховані шари виконують обробку ознак, а вихідний шар видає результат. Передача сигналу в таких мережах зазвичай здійснюється в одному напрямку — від входу до виходу, без зворотних зв'язків.

Кожен штучний нейрон у мережі виконує обчислення, що моделюють процес прийняття рішення. На вхід нейрона подаються сигнали $x_1, x_2, \dots, x_n$, кожен з яких має відповідну вагу $w_1, w_2, \dots, w_n$, яка визначає силу впливу даного входу. Вихід нейрона обчислюється за формулою:

$$y = f \left(\sum_{i=1}^n w_i x_i + b \right) \quad (1)$$

де b — зміщення (bias), а f — функція активації, яка визначає кінцеву реакцію нейрона. Така модель (рис. 2.1) дозволяє мережі апроксимувати складні нелінійні функції, що є критично важливим у задачах класифікації, зокрема при обробці зображень та відео.

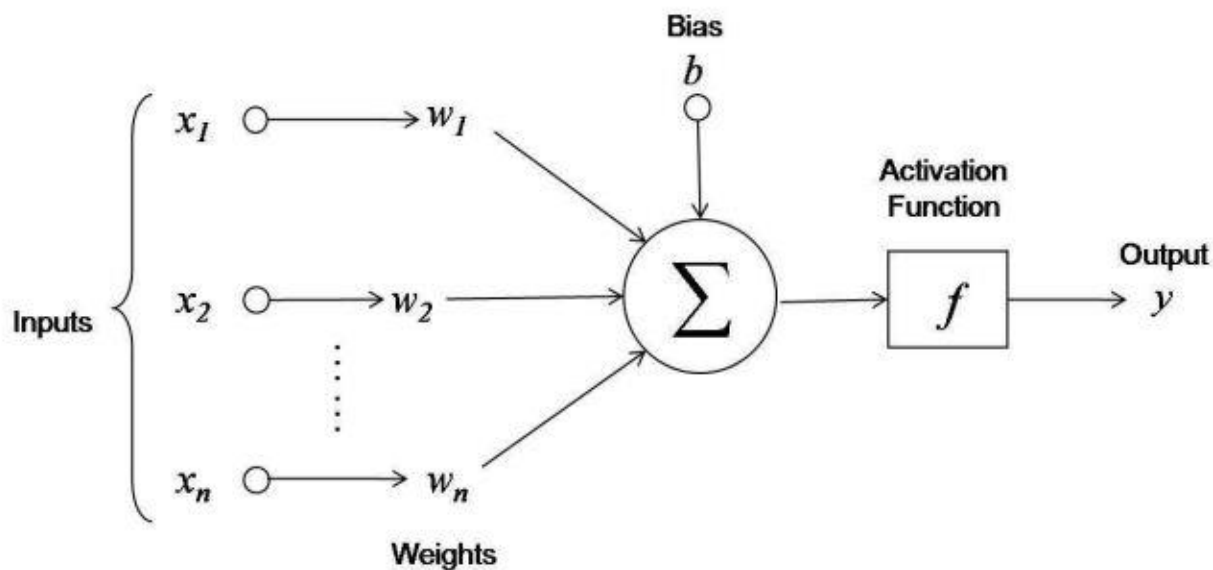


Рис. 2.1 Схема роботи штучного нейрона[10]

Нелінійна функція активації є ключовим компонентом нейронного обчислення, оскільки саме вона дозволяє мережі виявляти складні взаємозв'язки у даних. Без неї мережа складалась би лише з послідовних лінійних перетворень, що зводились би до одного шару. Використання таких функцій як сигмоїда, гіперболічний тангенс або ReLU забезпечує необхідну гнучкість для побудови моделей, здатних відображати складні залежності у вхідних даних. Згідно з теоремою про універсальну апроксимацію, навіть одна прихована шарова мережа з достатньою кількістю нейронів і нелінійною активацією здатна апроксимувати будь-яку неперервну функцію на компактній множині. У глибоких мережах, де використовується багато шарів, досягається ієрархічне представлення даних: нижчі шари відповідають за прості ознаки, такі як контури або текстури, тоді як вищі шари опрацьовують дедалі складніші структури та абстракції. Саме така багаторівнева обробка робить глибокі нейронні мережі особливо ефективними у задачах комп'ютерного зору, обробки природної мови, звуку та інших складних типів даних. У результаті, архітектура мережі визначає її здатність до навчання, генералізації й адаптації до широкого спектра прикладних завдань [10].

Однією з ключових класифікацій нейронних мереж є поділ на мілкі (shallow) та глибокі (deep) моделі. Мілкі нейронні мережі зазвичай містять лише один прихований шар і здатні реалізовувати прості функції перетворення вхідних даних. Така архітектура (рис. 2.2) є придатною для задач із низькою складністю, де обсяг даних або рівень абстракції ознак невисокий. Проте зі зростанням складності задач, зокрема пов'язаних із обробкою зображень, мови або часових рядів, ефективність мілких моделей суттєво знижується через обмежену здатність формувати ієрархії ознак.

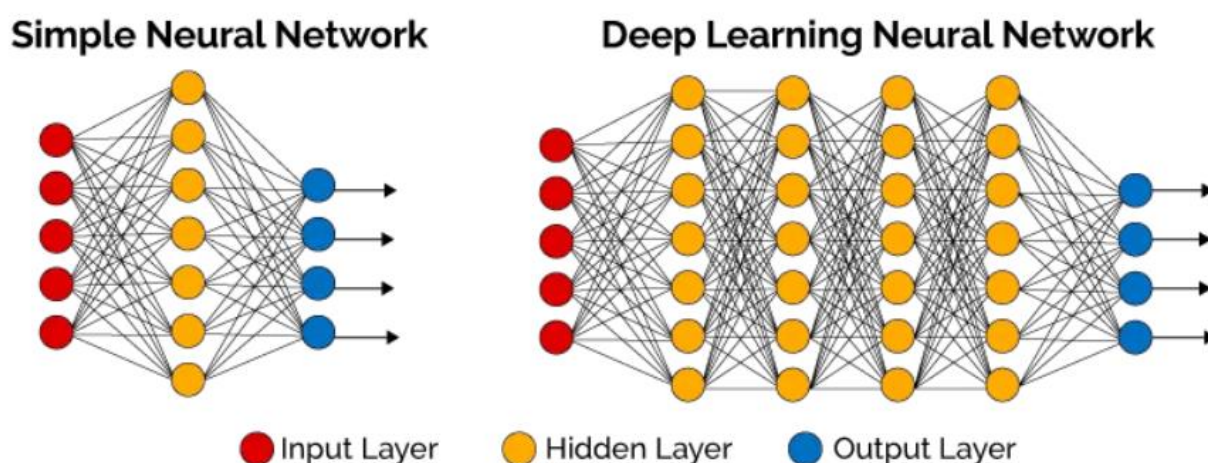


Рис. 2.2 Схема мілких та глибоких нейромереж [12]

Глибокі нейронні мережі, навпаки, складаються з двох і більше прихованих шарів, що дозволяє їм поступово формувати складні абстракції через багаторівневу обробку вхідної інформації. На кожному наступному рівні мережа трансформує дані у все більш узагальнене представлення, що підвищує точність класифікації та зменшує потребу в ручному створенні ознак. З технічної точки зору, кожен нейрон виконує лінійну комбінацію вхідних сигналів, до якої додається зміщення, після чого результат передається через нелінійну активаційну функцію. Навчання глибоких моделей відбувається за допомогою алгоритму зворотного поширення помилки з оптимізацією ваг методом градієнтного спуску.

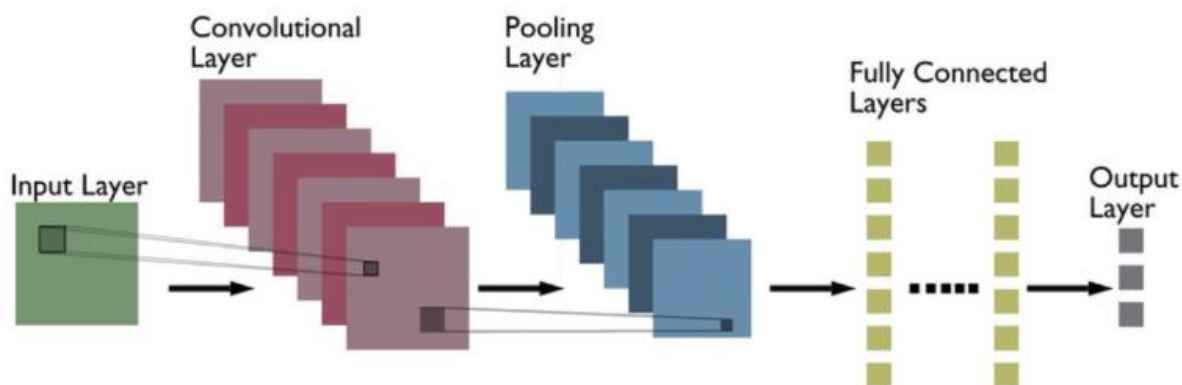


Рис. 2.3 Архітектура згорткової нейронної мережі(CNN) [13]

Одним із найважливіших представників глибоких архітектур є згорткова нейронна мережа (CNN), спеціально розроблена для обробки багатовимірних даних, насамперед зображень. CNN наслідують біологічні механізми зорової кори мозку й дозволяють виділяти просторові залежності у вхідних даних. Основу архітектури(рис. 2.3) складають згорткові шари, кожен елемент яких з'єднаний лише з обмеженою ділянкою вхідного простору, що називається рецептивним полем. Застосування згортки полягає у ковзанні фільтра (ядра) по всьому зображенню для виявлення локальних шаблонів, таких як краї, кути або текстури. Одні й ті ж фільтри застосовуються по всьому зображенню, що дозволяє моделі виявляти однакові ознаки незалежно від їхнього розташування.

Після згорткових операцій до структури мережі додаються шари субдискретизації (pooling), які зменшують розмірність простору ознак та виділяють найбільш релевантну інформацію. Це знижує обчислювальну складність та підвищує інваріантність моделі до незначних змін вхідного сигналу. В результаті багат шарового згорткового аналізу формується багатовимірне уявлення, яке передається до повнозв'язних шарів для остаточної класифікації. Така архітектура дозволяє будувати ефективні моделі для аналізу зображень, розпізнавання символів, обробки медичних знімків, детекції об'єктів та багатьох інших завдань, де важливо виявити локальні шаблони у складній структурі даних.

Завдяки використанню локальних з'єднань і спільних ваг, CNN значно скорочують кількість параметрів у порівнянні з класичними MLP, що дає змогу

тренувати великі мережі навіть на обмежених обчислювальних ресурсах. Крім того, сучасні реалізації CNN активно застосовують методи регуляризації (наприклад, Dropout), спеціальні активаційні функції (ReLU) та оптимізацію із використанням графічних процесорів, що робить їх винятково продуктивними в умовах обробки великих обсягів візуальної інформації.[11]

Застосування CNN у задачах розпізнавання номерних знаків транспортних засобів є ефективним завдяки здатності таких мереж працювати з відеопотоками в режимі реального часу, забезпечуючи високу точність і адаптивність до змінних умов. У таких системах згорткова нейронна мережа виконує класифікацію номерного знака після його локалізації та попередньої обробки зображення, яка може бути реалізована, наприклад, за допомогою бібліотеки OpenCV. CNN аналізує вхідні зображення, виділяючи релевантні ознаки для ідентифікації символів, навіть якщо зображення спотворене, затемнене або частково перекрите. [14]

Як приклад практичної реалізації, у відкритому проєкті License-Plate-Recognition-YOLOv7-and-CNN було створено конвеєр обробки відеопотоку, що включає виявлення транспортного засобу, корекцію положення номерного знака та його розпізнавання за допомогою CNN-моделі. Незважаючи на використання YOLOv7 для початкової детекції, ключова роль у розпізнаванні символів належить згортковій мережі, яка забезпечує точне дешифрування даних у реальному часі.



Figure. Input license plate



Figure. Final result

Рис. 2.4 Результати розпізнавання номерних знаків за допомогою згорткової нейронної мережі [15]

Окрема модель, орієнтована на відеоаналіз, продемонструвала здатність CNN забезпечувати високу продуктивність у складних умовах. Попередня сегментація дозволяє виділити цільові ділянки кадру, які потім обробляються CNN для отримання фінального результату(рис.2.4). Такий підхід забезпечує ефективну адаптацію до змін ракурсу, швидкості руху, фону та освітлення, що робить його придатним для використання в інтелектуальних транспортних системах, орієнтованих на моніторинг трафіку у розумному місті. [15]

2.2 Популярні інструменти машинного навчання для розпізнавання зображень

Комп'ютерний зір є міждисциплінарною галуззю, що досліджує методи автоматизованого аналізу та інтерпретації цифрових зображень і відео. Основна мета комп'ютерного зору полягає у створенні алгоритмів і систем, які здатні імітувати функції людського зору, розпізнавати об'єкти, визначати їхні характеристики та здійснювати відповідні дії на основі отриманих даних. Впровадження цієї технології у системи моніторингу дорожнього трафіку дозволяє автоматизувати процеси виявлення та класифікації транспортних засобів, що сприяє підвищенню ефективності управління дорожнім рухом.

Для розпізнавання об'єктів на зображеннях і відео комп'ютер використовує певні числові представлення даних. Зображення в цифровому форматі є матрицею пікселів, кожен із яких має певне значення яскравості або колірному каналу. У випадку кольорових зображень вони складаються з трьох основних каналів – червоного (Red), зеленого (Green) та синього (Blue), що дозволяє формувати кінцеве зображення через комбінацію інтенсивностей цих кольорів(рис.2.4). Для зображень у градаціях сірого використовується лише один канал яскравості. Таким чином, обробка зображень комп'ютером базується на математичних операціях над матрицями піксельних значень.

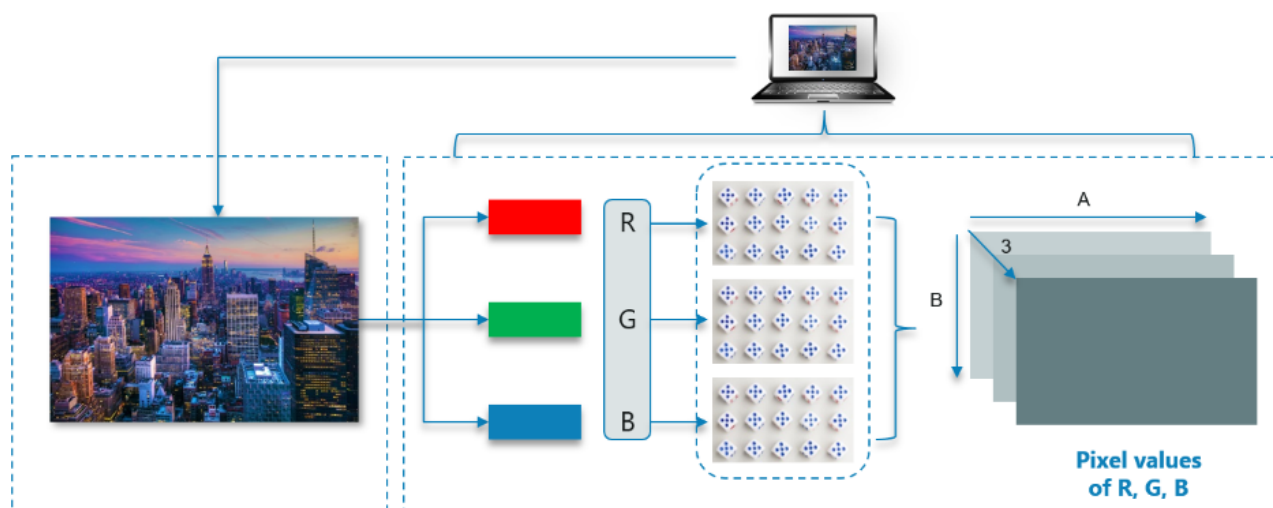


Рис. 2.4 Структура представлення зображення у форматі RGB [16]

Використання машинного навчання для розпізнавання об'єктів на зображеннях та відео значно покращує точність і швидкість обробки даних. Особливо ефективними в цьому контексті є згорткові нейронні мережі (CNN), які завдяки своїй архітектурі здатні автоматично виокремлювати релевантні ознаки транспортних засобів, дорожньої інфраструктури та інших об'єктів. Це дозволяє здійснювати детекцію автомобілів незалежно від змін освітлення, ракурсу чи погодних умов.

Процес розпізнавання зображень у таких системах складається з кількох основних етапів. Спочатку проводиться збір візуальної інформації за допомогою камер відеоспостереження, встановлених уздовж доріг або на транспортних вузлах. Далі застосовуються алгоритми попередньої обробки, які включають видалення шумів, покращення контрастності та нормалізацію зображень. Наступним етапом є сегментація та виділення ключових об'єктів, після чого відбувається класифікація транспортних засобів і визначення їхніх характеристик, таких як швидкість руху, тип автомобіля чи номерний знак.

Розвиток технологій розпізнавання зображень сприяє створенню більш ефективних систем управління дорожнім рухом, що дозволяє не лише оптимізувати транспортні потоки, а й підвищити безпеку на дорогах. Такі системи можуть використовуватися для автоматичного виявлення порушень правил дорожнього

руху, прогнозування заторів та аналізу транспортної інфраструктури. Інтеграція методів глибокого навчання з сучасними сенсорними технологіями відкриває широкі можливості для подальшого вдосконалення інтелектуальних транспортних систем.

TensorFlow є однією з найбільш популярних бібліотек для реалізації методів глибокого навчання, яка розроблена компанією Google Brain Team. Ця бібліотека надає широкий набір інструментів для створення, тренування та оптимізації моделей машинного навчання. Її використання охоплює завдання класифікації, регресії та розпізнавання зображень, що робить її універсальним рішенням для побудови інтелектуальних систем. TensorFlow відзначається масштабованістю та здатністю обробляти великі обсяги даних, що є ключовим фактором у розробці високопродуктивних систем аналізу дорожнього трафіку.

Архітектура TensorFlow(рис.2.5) має модульну структуру, що складається з кількох основних компонентів, які забезпечують ефективне виконання обчислень у моделях машинного навчання. Центральним елементом є обчислювальний граф (Graph), який представляє всі операції та взаємозв'язки між ними у процесі навчання та інференсу. Дані в моделі представлені у вигляді багатовимірних масивів, що називаються тензорами (Tensors), які є основною структурною одиницею для обміну інформацією між вузлами графа. Виконання обчислень у графі відбувається через операції (Operations), що охоплюють різноманітні математичні перетворення, необхідні для навчання моделі. Для запуску та керування виконанням графа використовується сесія (Session), яка ініціалізує операції та забезпечує виконання обчислень у визначеному середовищі. Завдяки такій архітектурі TensorFlow забезпечує гнучкість, масштабованість та ефективність у реалізації складних моделей машинного навчання.

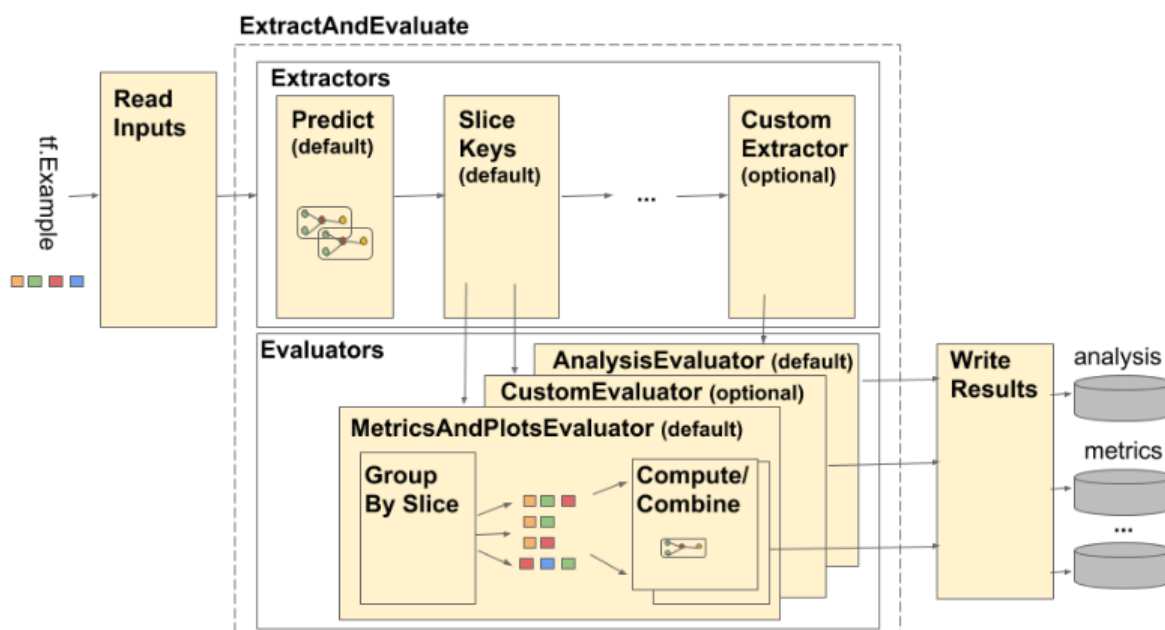


Рис. 2.5 Архітектура TensorFlow [17]

Однією з ключових особливостей TensorFlow є його здатність працювати на різних апаратних платформах, включаючи центральні процесори (CPU), графічні процесори (GPU) та тензорні процесори (TPU). Це дозволяє значно прискорювати обчислення та забезпечувати ефективне навчання глибоких нейронних мереж навіть при роботі з великими наборами даних. Окрім того, TensorFlow підтримує гнучке налаштування архітектур нейронних мереж та їх оптимізацію, що дозволяє адаптувати моделі до специфічних вимог системи моніторингу дорожнього руху.

Keras є високорівневою бібліотекою для створення нейронних мереж, яка побудована на основі TensorFlow. Вона спрощує процес розробки та навчання моделей завдяки зрозумілому та інтуїтивному API. Keras підтримує широкий спектр архітектур глибокого навчання, включаючи згорткові та рекурентні нейронні мережі, що робить її ефективним інструментом для задач розпізнавання зображень. Її перевагою є можливість швидкого створення прототипів моделей, що особливо важливо у контексті розробки інтелектуальних систем моніторингу.

PyTorch, розроблений Facebook, є ще однією потужною бібліотекою для машинного навчання, яка відзначається своєю гнучкістю та зручністю у використанні. Його головною перевагою є динамічний обчислювальний граф, який

дозволяє змінювати структуру моделі під час виконання, що є важливим для наукових досліджень та експериментів. PyTorch також забезпечує широкий набір інструментів для підготовки даних, навчання моделей та їх розгортання. Бібліотека активно використовується в академічному середовищі та промисловості завдяки своїй інтеграції з іншими інструментами аналізу даних, такими як NumPy та Pandas. PyTorch підтримує як центральні процесори, так і графічні прискорювачі, що дає змогу ефективно навчати складні моделі для розпізнавання зображень та обробки відеопотоків.[18]

Загалом, використання TensorFlow, Keras та PyTorch у розробці систем моніторингу дорожнього руху дозволяє реалізовувати високоточні алгоритми аналізу візуальних даних. Ці бібліотеки забезпечують широкий функціонал для роботи з глибокими нейронними мережами, що сприяє вдосконаленню технологій розпізнавання транспортних засобів та підвищенню ефективності управління трафіком.

OpenCV (Open Source Computer Vision)(рис.2.7) є потужною бібліотекою для обробки зображень, яка широко застосовується в задачах комп'ютерного зору. Вона дозволяє виконувати широкий спектр операцій, включаючи розпізнавання об'єктів, аналіз відеопотоків та ідентифікацію осіб. Спочатку розроблена Intel, OpenCV згодом підтримувалася компаніями Willow Garage та Itseez, а згодом знову повернулася під управління Intel. Бібліотека має відкритий вихідний код і випущена під ліцензією Apache 2.0, що робить її доступною як для академічного, так і для комерційного використання.

Серед бібліотек, що використовуються для реалізації систем комп'ютерного зору, OpenCV займає провідне місце завдяки своїй функціональності, відкритості та підтримці багатьох мов програмування. Вона надає розробникам інструменти для виконання базових і складних операцій з обробки зображень, що робить її універсальним рішенням для побудови інтелектуальних систем моніторингу. Її модульна архітектура дозволяє ефективно поєднувати обробку відеопотоків, аналіз об'єктів та інтеграцію з нейромережевими фреймворками. Це забезпечує високу

гнучкість при створенні рішень для розпізнавання транспортних засобів і подій у реальному часі.

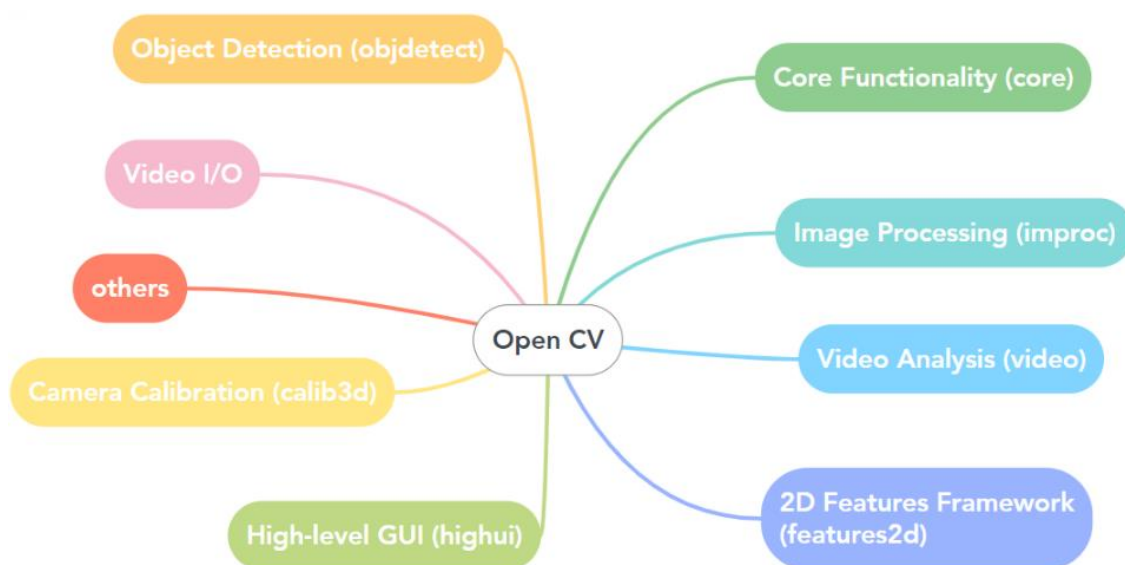


Рис. 2.7 Архітектура OpenCV[19]

OpenCV підтримує декілька мов програмування, зокрема C++, Python, Java та MATLAB, що дозволяє легко інтегрувати її в різні розробницькі середовища. Вона сумісна з основними операційними системами, такими як Windows, Linux, MacOS, iOS та Android. Завдяки можливості використання апаратного прискорення, включаючи GPU-обчислення, OpenCV забезпечує високу продуктивність у реальному часі. Бібліотека може працювати у зв'язці з іншими інструментами, такими як NumPy, що дозволяє ефективно виконувати математичні операції з векторними просторами для аналізу зображень. Вона є важливим інструментом для створення сучасних інтелектуальних систем моніторингу дорожнього руху.[20]

Рис 2.8 містить порівняльну характеристику чотирьох популярних інструментів машинного навчання та комп'ютерного зору: TensorFlow, Keras, PyTorch і OpenCV. Вона охоплює основні параметри, такі як розробник, призначення, гнучкість, продуктивність, підтримувані мови, переваги та недоліки кожної технології. Таблиця показує, що TensorFlow є потужною платформою для глибокого навчання, Keras забезпечує простоту у використанні, PyTorch вирізняється гнучкістю, а OpenCV спеціалізується на обробці зображень. Це

дозволяє визначити найкращий вибір інструменту залежно від конкретних завдань розпізнавання зображень.

Параметр	 TensorFlow	 Keras	 PyTorch	 OpenCV
Розробник	Google Brain	Google	Meta (Facebook)	Intel
Основне призначення	Глибоке навчання	Високорівневий API для TensorFlow	Гнучке машинне навчання	Обробка зображень
Гнучкість	Висока	Обмежена	Дуже висока	Висока
Продуктивність	Висока	Середня	Дуже висока	Висока
Підтримувані мови	Python, C++, Java	Python	Python, C++	Python, C++, Java
Переваги	Висока продуктивність, підтримка TPU, масштабованість	Простота використання, швидка побудова моделей	Гнучкість, зручний синтаксис, динамічний обчислювальний граф	Висока швидкість, широкий функціонал
Недоліки	Складний синтаксис	Обмежена гнучкість	Відносно низька продуктивність на великих даних	Відсутність вбудованих засобів для навчання моделей

Рис. 2.8 Порівняльна характеристика інструментів обробки зображень

Аналіз популярних інструментів машинного навчання для розпізнавання зображень демонструє, що вибір технології залежить від конкретних вимог проекту. TensorFlow є оптимальним вибором для продуктивних, масштабованих рішень, тоді як Keras підходить для швидкої розробки моделей з меншою складністю. PyTorch забезпечує максимальну гнучкість і зручність роботи з

динамічними нейронними мережами. OpenCV, хоча й не містить інструментів для навчання моделей, є ефективним засобом для попередньої обробки зображень та виконання комп'ютерного зору. Таким чином, ефективне використання зазначених бібліотек дозволяє розробляти високоточні та продуктивні системи розпізнавання зображень.

2.3 Вибір технічних інструментів для створення розумної системи моніторингу автомобілів

Успішна реалізація інтелектуальної системи моніторингу дорожнього трафіку безпосередньо залежить від правильного вибору програмного та апаратного забезпечення. Технічні інструменти визначають не лише функціональні можливості системи, але й такі ключові параметри, як швидкість обробки відеопотоку, точність розпізнавання об'єктів, стабільність роботи та можливість масштабування на більші ділянки міської інфраструктури.

Недостатньо потужне або неправильно підібране технічне середовище може призвести до затримок в обробці, помилок у розпізнаванні та загальної нестабільності системи. Тому вибір відповідних інструментів є критично важливим для забезпечення високої ефективності, надійності та подальшого масштабування системи у межах інтелектуальної транспортної інфраструктури міста.

Однією з ключових умов ефективного впровадження програмного модуля в системі інтелектуального моніторингу є вибір мови програмування, яка забезпечить баланс між продуктивністю, масштабованістю та простотою реалізації алгоритмів комп'ютерного зору. У межах цього проєкту як основну мову розробки обрано Python, що є домінуючим інструментом у сфері аналізу зображень, машинного навчання та автоматизованої обробки даних. Простота синтаксису Python істотно спрощує розробку та тестування прототипів, знижуючи вхідний поріг для розробників і дозволяючи швидко переходити від ідеї до реалізації [21].

Крім синтаксичної зручності, Python є надзвичайно поширеним інструментом у галузі комп'ютерного зору. Його активно використовують як в

академічному середовищі, так і в промислових рішеннях, що пов'язано з доступністю потужних фреймворків та бібліотек для обробки зображень, таких як OpenCV, NumPy, pytesseract, а також бібліотек для машинного навчання, таких як TensorFlow та PyTorch. Завдяки широкій екосистемі інструментів Python дозволяє реалізовувати як класичні алгоритми виявлення об'єктів, так і сучасні глибокі нейронні мережі [22].

Ще однією перевагою Python є потужна підтримка спільноти, яка забезпечує велику кількість відкритих рішень, документації, навчальних матеріалів та готових моделей для задач розпізнавання об'єктів. Це не лише пришвидшує розробку, а й підвищує надійність реалізованих рішень завдяки можливості адаптації перевірених підходів. Активне оновлення бібліотек і сумісність із сучасними апаратними платформами роблять Python універсальним інструментом для створення систем реального часу у сфері комп'ютерного зору [23].

Для реалізації комп'ютерного зору в системі моніторингу транспортних засобів важливо використовувати інструмент, здатний забезпечити ефективну обробку відеопотоку в реальному часі. У межах цього проєкту обрано бібліотеку OpenCV (Open Source Computer Vision Library), яка є однією з найпоширеніших у сфері комп'ютерного зору. Вона містить широкий набір функцій для обробки зображень, трансформацій та базових алгоритмів виявлення й відстеження об'єктів.

OpenCV підтримує роботу з попередньо навченими моделями нейронних мереж через модуль `cv2.dnn`, зокрема YOLO та Haar Cascade, що дозволяє поєднувати класичні та сучасні методи розпізнавання. Додатково, завдяки підтримці CUDA для графічних прискорювачів NVIDIA, бібліотека забезпечує високу швидкодію — ключовий параметр для завдань реального часу. Це робить OpenCV ефективним і гнучким інструментом для побудови системи розпізнавання номерних знаків[24].

Серед сучасних алгоритмів виявлення об'єктів особливої уваги заслуговує архітектура YOLO (You Only Look Once), яка забезпечує надзвичайно швидке та точне розпізнавання об'єктів у реальному часі. На відміну від багатоступеневих

підходів, YOLO виконує детекцію та класифікацію за один прохід нейронної мережі, що значно зменшує затримку обробки, критичну для задач відеомоніторингу. Новітні версії, зокрема YOLOv7, демонструють високу продуктивність і можуть бути інтегровані в системи на основі Python і OpenCV, що робить їх практичним вибором для проєктів автоматичного розпізнавання транспортних засобів і їхніх номерних знаків у міському середовищі [25].

Для розробки інтелектуальної системи моніторингу транспортних засобів було обрано апаратне забезпечення, що відповідає вимогам обробки відеоданих у режимі реального часу та забезпечує високу якість вхідної інформації. Основними критеріями при виборі апаратної частини стали надійність, продуктивність, сумісність із програмними модулями, а також можливість масштабування системи.



Camera	
Image Sensor	1/1.8" Progressive Scan CMOS
Max. Resolution	2688 × 1520
Min. Illumination	Color: 0.0001 Lux @ (F1.0, AGC ON), 0 Lux with light
Shutter Time	1 s to 1/100,000 s
Day & Night	IR cut filter
Angle Adjustment	Pan: 0° to 360°, tilt: 0° to 75°, rotate: 0° to 360°
Lens	
Lens Type	Fixed focal lens, 2.8 and 4 mm optional
Focal Length & FOV	2.8 mm, horizontal FOV 111.1°, vertical FOV 57.6°, diagonal FOV 138.5° 4 mm, horizontal FOV 95.2°, vertical FOV 48.3°, diagonal FOV 117.4°
Lens Mount	M16
Iris Type	Fixed
Aperture	F1.0
Depth of Field	2.8 mm: 2.5 m to ∞ 4 mm: 3.1 m to ∞
DORI	
DORI	2.8 mm: D: 61 m, O: 24 m, R: 12 m, I: 6 m 4 mm: D: 75 m, O: 30 m, R: 15 m, I: 7 m
Illuminator	
Supplement Light Type	White Light, IR
Supplement Light Range	Up to 30 m
Smart Supplement Light	Yes
IR Wavelength	850 nm
Video	
Main Stream	50 Hz: 25 fps (2688 × 1520, 1920 × 1080, 1280 × 720) 60 Hz: 30 fps (2688 × 1520, 1920 × 1080, 1280 × 720)
Sub-Stream	50 Hz: 25 fps (1280 × 720, 640 × 480, 640 × 360) 60 Hz: 30 fps (1280 × 720, 640 × 480, 640 × 360)
Third Stream	50 Hz: 10 fps (1920 × 1080, 1280 × 720, 640 × 480, 640 × 360) 60 Hz: 10 fps (1920 × 1080, 1280 × 720, 640 × 480, 640 × 360) *The third stream is supported under certain settings.
Video Compression	Main stream: H.265/H.264/H.264+, Sub-stream: H.265/H.264/MPEG, Third stream: H.265/H.264, *Third stream is supported under certain settings.
Video Bit Rate	32 Kbps to 8 Mbps
H.264 Type	Baseline Profile, Main Profile, High Profile
H.265 Type	Main Profile
Bit Rate Control	CBR, VBR
Scalable Video Coding (SVC)	H.264 and H.265 encoding
Region of Interest (ROI)	1 fixed region for main stream
Audio	
Audio Type	Mono sound
Audio Compression	G.711/G.722.1/G.726/MP2L2/PCM/MP3/AAC-LC

Рис. 2.9 Специфікації і вигляд камери Hikvision DS-2CD7A26G0/P-IZHS[26]

У якості джерела відеопотоку було обрано мережеві камери з підтримкою Full HD (1920×1080), нічного бачення та протоколів RTSP або ONVIF, що забезпечує стабільну інтеграцію з програмним середовищем на базі OpenCV. Зокрема, доцільним є використання моделі Hikvision DS-2CD7A26G0/P-IZHS(рис.

2.9), яка поєднує високу якість зображення з широким діапазоном фокусування та можливістю передачі відео в реальному часі по захищених каналах зв'язку [26].

До основних технічних характеристик камери, що забезпечують її ефективне застосування в системі моніторингу, належать:

а) Роздільна здатність: 2 МП (1920×1080) — забезпечує достатню деталізацію номерних знаків;

б) Інфрачервоне підсвічування (IR): до 100 метрів — дозволяє фіксувати зображення в умовах низької освітленості;

с) Оптичний зум: до 25× — дає змогу масштабувати об'єкти без втрати якості;

д) Підтримка WDR (Wide Dynamic Range): до 140 дБ — покращує зображення при різкому контрасті світла і тіні;

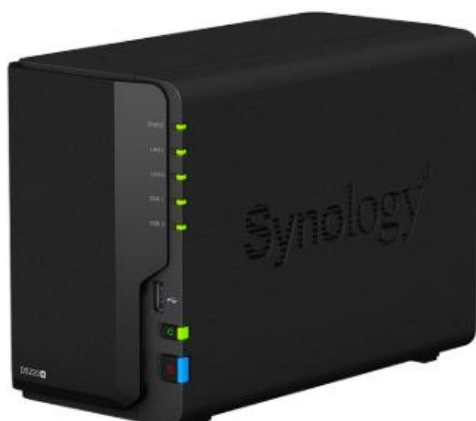
е) Протоколи зв'язку: RTSP, ONVIF — забезпечують сумісність із системами відеоаналітики;

ф) Формати стиснення відео: H.265/H.264 — зменшують навантаження на мережу та обчислювальний модуль.

Ці параметри дозволяють стабільно отримувати якісні зображення в різних умовах, що є критично важливим для точного виявлення та розпізнавання транспортних засобів[26].

Для виконання обчислювальних операцій було обрано персональний комп'ютер або вбудовану систему з процесором не нижче Intel Core i5 або AMD Ryzen 5, оперативною пам'яттю обсягом 8–16 ГБ, твердотілим накопичувачем (SSD) для прискореного зчитування даних та, за необхідності, графічним процесором NVIDIA з підтримкою CUDA для прискорення роботи нейронних мереж. Така конфігурація дозволяє обробляти відеопотік без затримок і здійснювати розпізнавання номерних знаків у режимі реального часу [27].

Для передачі відеоданих між камерою та обчислювальним модулем використовується мережеве з'єднання Gigabit Ethernet (GigE), що гарантує високу пропускну здатність і мінімальні затримки, необхідні для стабільної роботи системи [28].



Technical Specifications

Hardware

CPU	Intel Celeron J4025 2-core 2.0 GHz, burst up to 2.9 GHz
Hardware encryption engine	Yes (AES-NI)
Memory	2 GB DDR4 onboard (expandable up to 6 GB) ¹
Compatible drive type	2 x 3.5" or 2.5" SATA HDD/SSD (drives not included)
Hot swappable drive	Yes
External port	2 x USB 3.0 port
Size (HxWxD)	165 x 108 x 232.2 mm
Weight	1.3 kg
LAN	2 x Gigabit (RJ-45)
Wake on LAN/WAN	Yes
Scheduled power on/off	Yes
System fan	92 x 92 x 25 mm
AC input power voltage	100 V to 240 V AC
Power frequency	50/60 Hz, single phase
Operating temperature	0°C to 40°C (32°F to 104°F)
Storage temperature	-20°C to 60°C (-5°F to 140°F)
Relative humidity	5% to 95% RH
Maximum operating altitude	5,000 m (16,400 ft)

Рис 2.10 Специфікації та вигляд сховища Synology DiskStation DS220+[29]

Для централізованого зберігання відеоданих, зафіксованих транспортних порушень, та результатів розпізнавання номерних знаків у межах системи моніторингу обрано мережеве сховище Synology DiskStation DS220+(рис.2.10). Це двохдисковий NAS-сервер, призначений для безперервної роботи в умовах помірного навантаження, що ідеально підходить для локальних систем відеоспостереження та відеоаналітики на підприємствах або в муніципальних структурах.

Серед основних технічних характеристик моделі варто виділити:

- а) Процесор: двоядерний Intel Celeron J4025, до 2.9 ГГц (підтримка апаратного шифрування AES-NI);
- б) Оперативна пам'ять: 2 ГБ DDR4 (з можливістю розширення до 6 ГБ);
- с) Підтримка RAID: JBOD, RAID 0, RAID 1 — для підвищення надійності зберігання;
- д) Інтерфейси підключення: 2 × Gigabit Ethernet, 2 × USB 3.0 — забезпечують стабільну передачу даних;
- е) Максимальна ємність: до 32 ТБ (2 × 16 ТБ HDD);

Програмне середовище DSM (DiskStation Manager) з вбудованими інструментами моніторингу, резервного копіювання, прав доступу та інтеграції з IP-камерами через Surveillance Station.

Завдяки цим характеристикам NAS-сервер Synology DS220+ є надійною платформою для безперервного зберігання даних у реальному часі, підтримуючи стабільну взаємодію з модулями обробки Python + OpenCV через локальну мережу[29].

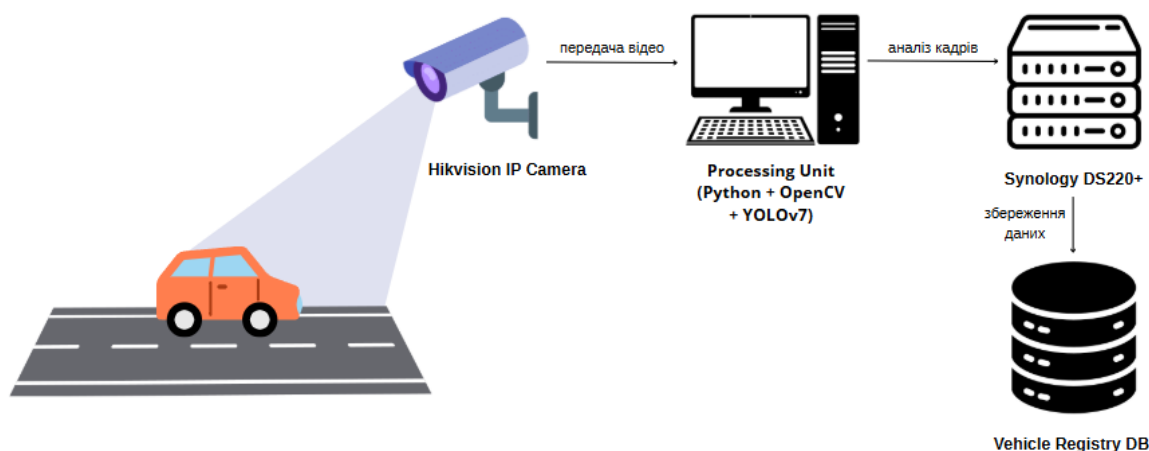


Рис. 2.11 Функціональна схема взаємодії компонентів

Функціональна схема системи (рис. 2.11) передбачає послідовну взаємодію апаратних і програмних компонентів для забезпечення автоматизованого розпізнавання номерних знаків у реальному часі. Відеопотік із мережевої камери (наприклад, Hikvision DS-2CD7A26G0/P-IZHS) передається по протоколу RTSP через Gigabit Ethernet до обчислювального модуля, де за допомогою Python, OpenCV та моделі YOLOv7 здійснюється виявлення транспортних засобів і розпізнавання номерних знаків. Результати обробки — зображення з метаданими — автоматично зберігаються на NAS-сервері (Synology DS220+), що забезпечує надійне зберігання інформації з можливістю подальшої обробки або передачі до зовнішніх баз даних.

Отже, комбінація мови програмування Python, бібліотеки OpenCV та підбраного апаратного забезпечення забезпечує оптимальні умови для створення

ефективної системи розпізнавання номерних знаків у межах інтелектуального моніторингу дорожнього трафіку. Python надає гнучкість у реалізації алгоритмів та інтеграції з неймережами, OpenCV — високопродуктивні інструменти обробки відео та зображень, а апаратна частина — достатній обчислювальний ресурс і надійність роботи в реальному часі. Така технологічна конфігурація є масштабованою, оскільки допускає підключення додаткових камер і серверів без радикальної перебудови системи; економічно доцільною, завдяки використанню відкритого програмного забезпечення та доступного обладнання; та функціонально повною, що дозволяє реалізувати повний цикл — від відеофіксації до зберігання та аналізу даних.

3 СТВОРЕННЯ СИСТЕМИ МОНІТОРИНГУ ТРАФІКУ НА ДОРОГАХ

3.1 Постановка задачі та технічні вимоги до системи

Із розвитком концепції «розумного міста» все більшої актуальності набувають автоматизовані засоби контролю транспортного трафіку. Однією з ключових задач у цьому напрямі є створення системи розпізнавання номерних знаків транспортних засобів у реальному часі з можливістю збереження даних для подальшої обробки та аналітики. Така система повинна інтегруватися в існуючу міську інфраструктуру відеоспостереження, працювати з високою точністю в умовах обмеженої пропускної здатності мережі, різноманітних погодних умов та змінного освітлення.

У межах реалізації проєкту передбачено виконання таких функціональних задач:

- а) забезпечення стабільного зчитування відеопотоку з IP-камери в режимі реального часу;
- б) виявлення присутності транспортних засобів у кадрі за допомогою нейромережевих методів;
- в) точна локалізація області номерного знака на транспортному засобі;
- г) оптичне розпізнавання символів номерного знака з використанням технологій OCR;
- д) фіксація супровідної інформації: дата, час, ID камери, скріншот кадру;
- е) збереження отриманих результатів у централізованому мережевому сховищі для подальшої аналітики;
- ж) забезпечення взаємодії з іншими інформаційними системами через API.

Для забезпечення ефективності функціонування розроблюваної системи критичною є формалізація вимог (табл.3.1) до її надійності та масштабованості. Зокрема, система повинна підтримувати стабільну роботу в режимі тривалого безперервного моніторингу без втрати цілісності потоку даних. У разі виникнення непередбачених обставин — таких як тимчасова недоступність відеопотоку або

збій з'єднання — необхідно реалізувати механізми автоматичного відновлення. Масштабованість системи забезпечується можливістю розширення до декількох IP-камер та збереженням архітектурної цілісності програмного забезпечення. При цьому кожен новий модуль обробки відео повинен бути незалежним та здатним інтегруватися з основним сховищем даних без необхідності повної реконфігурації системи.

Таблиця 3.1

Вимоги до системи розпізнавання номерних знаків

Категорія	Вимога
Мова розробки	Python 3.x
Бібліотеки	OpenCV, pytesseract, YOLOv7, NumPy
Камера	Hikvision DS-2CD7A26G0/P-IZHS
Обчислення	Комп'ютер з GPU (підтримка CUDA)
Зберігання даних	NAS Synology DS220+
Формат відео	RTSP-потік
Пропускна здатність	Gigabit Ethernet
Макс. час обробки	≤ 200 мс на кадр
Точність розпізнавання	$\geq 85\%$
Надійність	Автовідновлення після збоїв, стабільна робота 24/7
Масштабованість	Можливість підключення нових камер без зміни архітектури
Безпека	Шифрування даних, обмеження доступу, ведення журналів подій
Сценарії використання	Паркінги, контроль трафіку, безпека об'єктів, аналітика

Оскільки система має справу з персоналізованими даними — зокрема, з відеозаписами та символами номерних знаків — актуальним є питання забезпечення інформаційної безпеки. На всіх етапах обробки передбачено використання засобів шифрування даних при передачі по мережі, автентифікації користувачів, обмеження доступу до сховища та ведення журналу подій. Уся взаємодія з мережею має відбуватись через захищені канали, з урахуванням політик конфіденційності та нормативних вимог до обробки персональних даних, що діють у відповідному регіоні.

Згідно із практичним призначенням, система має забезпечувати широкий спектр сценаріїв застосування. До них належать: контроль проїзду на платні паркінги, облік транспортного потоку на міських магістралях, контроль доступу до закритих територій, фіксація фактів порушень ПДР або інтеграція з аналітичними платформами муніципалітетів. Результати розпізнавання — у вигляді текстових записів, скріншотів кадру, метаданих (дата, час, ідентифікатор камери) — мають архівуватись для подальшої перевірки або передачі до зовнішніх інформаційних систем через API.

Крім основних функцій, система повинна бути орієнтована на використання доступного й широко підтримуваного програмного забезпечення. У якості середовища розробки обрано Python 3.x — популярну мову програмування в галузі комп'ютерного зору, яка сумісна з провідними бібліотеками, такими як OpenCV, pytesseract, NumPy та YOLOv7. Це дозволяє реалізувати як детекцію транспортних засобів, так і розпізнавання символів номерного знака у межах єдиної програмної архітектури.

На рівні апаратного забезпечення система використовує IP-камеру Hikvision DS-2CD7A26G0/P-IZHS з підтримкою RTSP-протоколу, комп'ютер з GPU, сумісним із технологією CUDA для обчислювального прискорення, а також мережеве сховище Synology DS220+ для централізованого зберігання даних. Очікується, що час обробки одного кадру не перевищуватиме 200 мс, що забезпечує роботу в квазіреальному часі. Цільовий показник точності розпізнавання номерних знаків становить не менше 85% в умовах нормального освітлення, а також при наявності зовнішніх перешкод, таких як дощ або пил.

3.2 Реалізація моделі для розпізнавання номерних знаків

На етапі реалізації системи було розроблено архітектуру програмного рішення, яке забезпечує послідовну обробку відеопотоку з метою автоматичного виявлення транспортних засобів та розпізнавання їх номерних знаків. Для досягнення високої точності та швидкодії застосовано комбінацію методів

комп'ютерного зору та глибокого навчання з використанням бібліотек OpenCV, YOLOv4-tiny та модуля оптичного розпізнавання символів Tesseract OCR.

Основна логіка роботи системи поділяється на два умовні етапи: 1) ініціалізаційний, у межах якого здійснюється завантаження моделей, конфігурацій та запуск відеопотоку; 2) операційний, який охоплює обробку відеокадрів, виявлення транспортних засобів, локалізацію номерного знака та розпізнавання його символів. У результаті реалізовано замкнений цикл обробки вхідних даних у реальному часі.

Для наочного представлення структури та логіки роботи розробленого програмного рішення була створена архітектурна блок-схема. Вона демонструє взаємозв'язок між основними функціональними модулями, відображає послідовність обробки даних і вказує на використані технології та інструменти. Це дозволяє чітко зрозуміти алгоритм роботи системи, визначити ключові процеси та візуалізувати потік даних від початку надходження інформації до отримання кінцевого результату у вигляді розпізнаного номерного знака транспортного засобу.

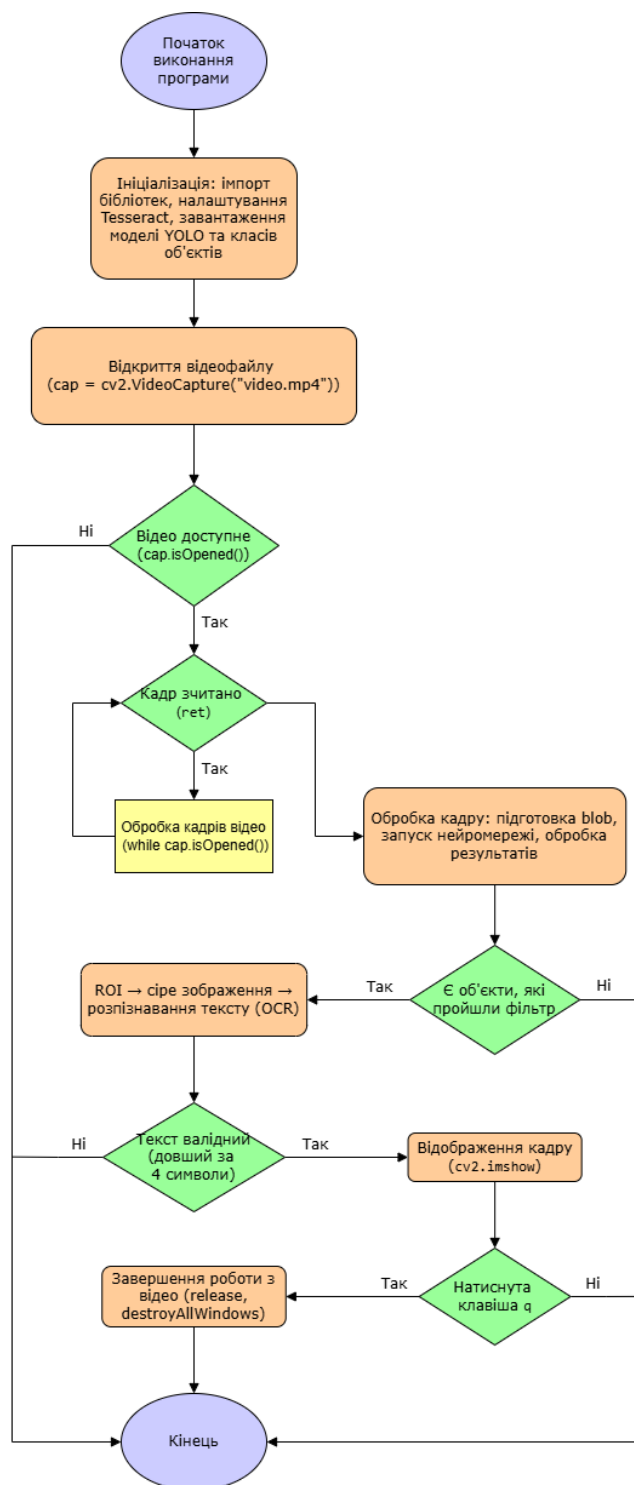


Рис. 3.1 Архітектура програмного рішення

Блок-схема (рис. 3.1) ілюструє послідовність виконання основних етапів. На початку відбувається імпорт необхідних бібліотек, зокрема cv2, pytesseract і numpy, та ініціалізація Tesseract через вказання шляху до його виконуваного файлу. Далі завантажуються конфігурація моделі YOLOv4-tiny, включно з вагами та класами

об'єктів (наприклад, «car», «bus»), після чого відкривається відеопотік з IP-камери або з відеофайлу через протокол RTSP.

Після завершення ініціалізації система переходить у режим циклічного зчитування кадрів. Кожен кадр перетворюється у формат blob і передається до згорткової нейромережі YOLO для виявлення об'єктів. Результати обробки проходять фільтрацію: залишаються лише транспортні засоби, а перекриті об'єкти обробляються за допомогою методу Non-Maximum Suppression (NMS). На основі координат об'єкта локалізується номерний знак, і над виділеною областю виконується розпізнавання тексту за допомогою pytesseract.

Після цього розпізнаний номер зберігається у вигляді тексту, а разом із ним – знімок кадру, мітка часу та інші метадані. У разі потреби ці результати можуть бути виведені на екран або передані в зовнішні системи через API. Таким чином, реалізована модель повністю покриває вимоги до продуктивності, точності та масштабованості, що були визначені на етапі постановки задачі.

Розробка програмного модуля для автоматичного розпізнавання номерних знаків транспортних засобів реалізована за допомогою мови Python з використанням бібліотек OpenCV, NumPy, pytesseract, а також попередньо навченої нейронної мережі YOLOv4-tiny. Програма забезпечує обробку відеопотоку в режимі квазіреального часу, виконуючи виявлення транспортних засобів у кадрі, локалізацію області номерного знака та оптичне розпізнавання тексту з подальшим виведенням і збереженням результатів.

На початковому етапі здійснюється імпорт необхідних бібліотек, які забезпечують доступ до функціоналу комп'ютерного зору (cv2), чисельних обчислень (numpy) та оптичного розпізнавання символів (pytesseract). Для коректної роботи Tesseract OCR на операційних системах Windows виконується явне вказання шляху до його виконуваного файлу(рис.3.2).

```

1 import cv2
2 import pytesseract
3 import numpy as np
4
5
6 pytesseract.pytesseract.tesseract_cmd = r'C:\Program Files\Tesseract-OCR\tesseract.exe'

```

Рис. 3.2 Імпорт бібліотек і налаштування OCR

Далі здійснюється завантаження згорткової нейронної мережі YOLOv4-tiny, яка призначена для виявлення об'єктів у зображенні. Завдяки своїй спрощеній архітектурі ця модель демонструє високу швидкість роботи при збереженні задовільного рівня точності, що є доцільним у задачах обробки відео. Мережа ініціалізується шляхом підключення конфігураційного файлу (.cfg) та зважених параметрів (.weights).

Після цього зчитується перелік класів об'єктів, що розпізнаються моделлю. Для цього використовується текстовий файл coco.names.txt, у якому кожен рядок відповідає певному класу (наприклад, «car», «bus», «truck»). Список класів буде використано пізніше для фільтрації лише транспортних засобів(рис.3.3).

```

9 net = cv2.dnn.readNet("yolov4-tiny.weights", "yolov4-tiny.cfg")
10
11 with open("coco.names.txt", "r") as f:
12     classes = [line.strip() for line in f.readlines()]

```

Рис. 3.3 Завантаження моделі YOLOv4-tiny

Після завантаження моделі відбувається підключення відеопотоку, джерелом якого може бути відеофайл або IP-камера. В даному випадку для тестування використовується локальне відео(рис.3.4).

```

16 cap = cv2.VideoCapture("video.mp4")

```

Рис. 3.4 Відкриття відеопотоку

Основна логіка системи реалізована у вигляді циклу, що зчитує кадри з відео. Для кожного кадру визначаються його розміри та здійснюється підготовка зображення для подачі у нейронну мережу. Для цього зображення конвертується у формат blob (бінарне велике зображення), масштабується до розміру 416×416 пікселів, нормалізується та змінюється порядок кольорових каналів(рис.3.5).

```

18 while cap.isOpened():
19     ret, frame = cap.read()
20     if not ret:
21         break
22
23     height, width = frame.shape[:2]
24
25
26     blob = cv2.dnn.blobFromImage(frame, 1/255.0, (416, 416), swapRB=True, crop=False)
27     net.setInput(blob)

```

Рис. 3.5 Обробка відео в циклі

Мережа повертає список виявлених об'єктів, після чого здійснюється постобробка результатів. Для кожної детекції обчислюється клас із найвищим значенням впевненості (confidence). З метою оптимізації відбору враховуються лише ті об'єкти, які належать до класів "car", "bus" або "truck", та мають імовірність вище порогу 0.5 (рис.3.6).

```

29 outputs = net.forward(layer_names)
30 boxes = []
31 confidences = []
32 class_ids = []
33
34 for out in outputs:
35     for detection in out:
36         scores = detection[5:]
37         if len(scores) == 0:
38             continue
39
40         class_id = np.argmax(scores)
41         if class_id >= len(classes):
42             continue
43
44         confidence = scores[class_id]
45         if confidence > 0.5 and classes[class_id] in ['car', 'bus', 'truck']:
46             cx, cy, w, h = (detection[0:4] * np.array([width, height, width, height])).astype("int")
47             x = int(cx - w / 2)
48             y = int(cy - h / 2)
49             boxes.append([x, y, w, h])
50             confidences.append(float(confidence))
51             class_ids.append(class_id)

```

Рис. 3.6 Фільтрація транспортних засобів

Для усунення надлишкових детекцій застосовується алгоритм приглушення перекриттів — Non-Maximum Suppression (NMS). Цей етап дозволяє залишити лише ті вікна, що мають максимальну впевненість серед перекриваних детекцій (рис.3.7).

```
53 indexes = cv2.dnn.NMSBoxes(boxes, confidences, 0.5, 0.4)
```

Рис.3.7 Обчислення координат і усунення перекриттів (NMS)

У фінальній частині програми кожне вікно, яке залишилося після фільтрації, вважається областю з потенційним номерним знаком. Виділена область вирізається з кадру, конвертується у відтінки сірого, після чого передається на обробку OCR-системі Tesseract(рис.3.8).

```
55 for idx in indexes:
56     i = idx[0] if isinstance(idx, (list, tuple, np.ndarray)) else idx
57     x, y, w, h = boxes[i]
58     roi = frame[y:y+h, x:x+w]
59
60
61     if roi.size == 0:
62         continue
63
64     gray = cv2.cvtColor(roi, cv2.COLOR_BGR2GRAY)
65     text = pytesseract.image_to_string(gray, config='--psm 7')
66     text = text.strip()
```

Рис.3.8 Локалізація номера та розпізнавання OCR

Застосування параметра `--psm 7` дозволяє налаштувати Tesseract на обробку одного рядка тексту, що підвищує точність при розпізнаванні номерних знаків. Якщо текст містить достатню кількість символів (наприклад, більше 4), він виводиться на кадрі разом із прямокутником навколо виявленої області(рис.3.9).

```

68         if len(text) > 4:
69             cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)
70             cv2.putText(frame, text, (x, y - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.9, (36, 255, 12), 2)
71             print("Знайдено номер:", text)

```

Рис.3.9 Інтеграція результату OCR у відеопотік

На завершальному етапі кадр виводиться у вікно з результатами, і якщо користувач натискає клавішу q, цикл переривається, а ресурси вивільнюються (рис.3.10).

```

74         cv2.imshow("Number Plate Detection", frame)
75         if cv2.waitKey(200) == ord('q'):
76             break
77
78     cap.release()
79     cv2.destroyAllWindows()

```

Рис.3.10 Завершення циклу обробки та очищення ресурсів

Таким чином, розроблена система демонструє повний цикл виявлення, локалізації та розпізнавання номерних знаків у реальному часі, поєднуючи ефективність згорткових нейромереж із гнучкістю OCR. Запропоноване рішення є модульним, масштабованим і може бути адаптоване до різних джерел відео або типів камер.

3.3 Тестування роботи нейромережі та програмного функціоналу

Метою тестування розробленої системи є перевірка коректності функціонування програмного забезпечення, що здійснює виявлення транспортних засобів на відеопотоці та розпізнавання номерних знаків. Основна увага приділяється оцінці ефективності застосування нейронної мережі YOLOv4-tiny для детекції об'єктів категорій «car», «bus» та «truck», а також точності оптичного розпізнавання символів (OCR) за допомогою інструменту Tesseract. Таке

тестування дозволяє визначити, наскільки успішно система виконує поставлені завдання в реальних умовах роботи.

У процесі перевірки аналізуються ключові параметри: точність виявлення об'єктів, коректність зчитування текстової інформації з номерних знаків, стабільність роботи системи при обробці відеопотоку, а також її продуктивність за часовими характеристиками. Зібрані результати дозволяють зробити обґрунтовані висновки щодо доцільності обраної архітектури, ефективності використаних алгоритмів і технологій, а також визначити можливі напрями подальшої оптимізації та вдосконалення розробленого програмного функціоналу.

Тестування програмної системи проводилося з метою перевірки її працездатності в умовах, наближених до реальних сценаріїв застосування інтелектуального моніторингу дорожнього трафіку. Передбачалося, що джерелом відео для обробки в рамках проєкту може бути як IP-камера, що забезпечує безперервну трансляцію потоку в режимі реального часу, так і локальний відеофайл. У межах тестування було використано відеозапис, завантажений з відкритого джерела[30], який моделює типову ситуацію міського транспортного руху. Таким чином, для даного етапу розробки джерелом даних обрано локальний відеофайл video.mp4 (рис.3.11), що дозволяє провести стабільне та відтворюване тестування.

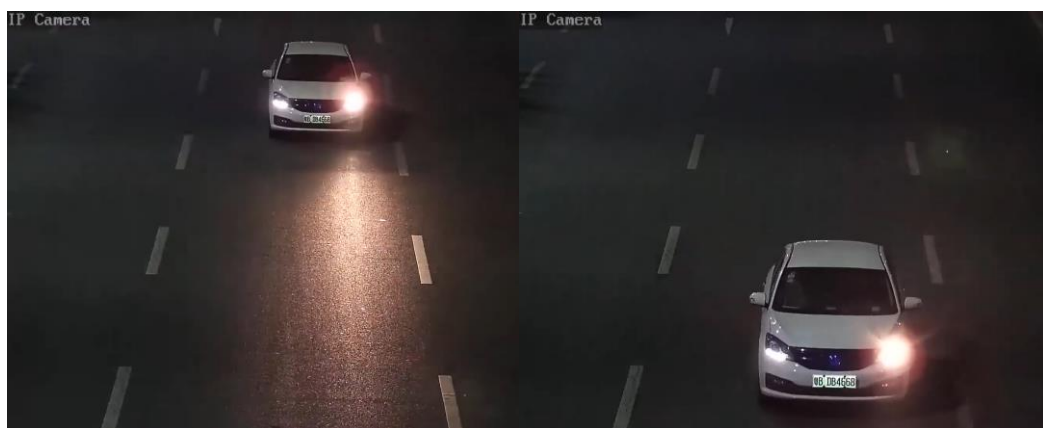


Рис.3.11 кадри з відеофайлу для тестування моделі

Характеристики відео включають стандартну роздільну здатність Full HD (1920×1080 пікселів), частоту кадрів 30 кадрів на секунду та достатній рівень освітлення для візуального виявлення транспортних засобів і їх номерних знаків. Завдяки високій якості зображення та чіткості деталей, дане відео дозволяє адекватно оцінити ефективність роботи нейронної мережі з виявлення об'єктів, а також точність оптичного розпізнавання текстової інформації. Застосування локального відеофайлу дає змогу забезпечити контрольованість тестового середовища, що є важливою умовою для проведення об'єктивного аналізу продуктивності системи.

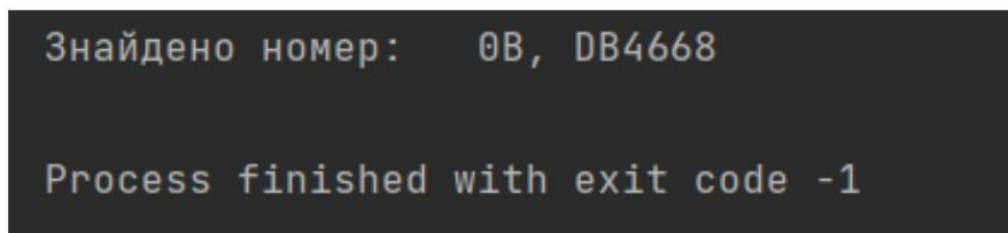
У процесі тестування було здійснено запуск розробленого програмного забезпечення із використанням локального відеофайлу, що відображає дорожню сцену в умовах нічного часу доби. Вхідне відео містить чітке зображення транспортного засобу, що рухається дорогою під кутом, з характерним освітленням від фар. Такий тип контенту дозволив перевірити здатність системи до виявлення об'єктів у складних умовах зйомки та до точного зчитування текстової інформації з номерного знака.



Рис.3.12 вхідний кадр і результат програми

У вихідному кадрі(рис.3.12), отриманому з відеопотоку, зображено транспортний засіб без жодної графічної або текстової обробки. Після обробки цього самого кадру за допомогою створеної програми було здійснено успішне

виявлення транспортного засобу за допомогою нейронної мережі YOLOv4-tiny, візуальне окреслення прямокутною рамкою області з номерним знаком, а також розпізнавання символів за допомогою OCR-алгоритму Tesseract. Поверх зображення програма автоматично накладає результат розпізнавання — "0B, DB4668", що підтверджує її функціональну спроможність.



```
Знайдено номер: 0B, DB4668  
Process finished with exit code -1
```

Рис.3.13 вивід програми

Крім візуального представлення результату, програма додатково формує текстовий вивід(рис.3.13), який містить повідомлення "Знайдено номер: 0B, DB4668". Такий підхід дозволяє не лише виводити інформацію для користувача в реальному часі, а й забезпечує можливість подальшої обробки даних, їх фіксації в логах або збереження в базі даних. Завершення роботи додатку супроводжується повідомленням Process finished with exit code -1, що пояснюється примусовим завершенням процесу за ініціативою користувача і не є ознакою критичної помилки.

Загалом результати тестування підтверджують коректність функціонування програмного забезпечення та придатність створеного рішення для вирішення задачі автоматичного виявлення транспортних засобів і розпізнавання номерних знаків у відеопотоці.

Попри задовільні результати тестування, подальше вдосконалення системи є доцільним з огляду на розширення її функціональності та підвищення точності виявлення й розпізнавання об'єктів. Одним із перспективних напрямків розвитку є впровадження модулів, що дозволяють автоматично ідентифікувати не лише номерні знаки, а й марку, модель та колір транспортного засобу. Таке розширення можливостей може бути реалізоване за допомогою глибших архітектур нейронних

мереж (наприклад, YOLOv8, EfficientNet або ResNet), натренованих на відповідних датасетах, таких як Stanford Cars, CompCars або VeRi-776 [31].

Також доцільним є вдосконалення модуля оптичного розпізнавання символів. Заміна Tesseract на сучасніші бібліотеки, зокрема EasyOCR, дозволяє досягти вищої точності зчитування номерних знаків, особливо у випадках складних фонів, нестабільного освітлення або викривлених зображень [32]. EasyOCR підтримує декілька мов, працює на основі глибоких згорткових мереж та демонструє високу ефективність при обробці номерів у реальному часі.

Ще одним важливим напрямом удосконалення є оптимізація обробки відео з IP-камер у режимі реального часу. Це забезпечить інтеграцію системи у практичні задачі міського моніторингу, зокрема для контролю руху на перехрестях або в автоматизованих системах доступу. Для досягнення такої цілі доцільно реалізувати апаратне прискорення (наприклад, через GPU) та використання легких моделей або фреймворків, які підтримують багатопотокову обробку.

З урахуванням зазначених напрямків удосконалення, розроблена система може бути масштабована до рівня повноцінного модуля інтелектуального відеоспостереження, здатного автоматично ідентифікувати об'єкти, накопичувати статистику та реагувати на виявлені події в режимі реального часу.

У результаті тестування підтверджено коректність роботи розробленої системи в задачах виявлення транспортних засобів та розпізнавання номерних знаків у відеопотоці. Нейромережа YOLOv4-tiny успішно ідентифікує об'єкти, а модуль Tesseract забезпечує зчитування номерів із достатньою точністю. Система продемонструвала стабільну роботу навіть в умовах слабкого освітлення, а текстові результати розпізнавання збігалися з графічним виводом. Отримані результати свідчать про практичну придатність програмного забезпечення та потенціал для подальшого вдосконалення й інтеграції у системи інтелектуального моніторингу.

ВИСНОВКИ

У межах виконання кваліфікаційної роботи було здійснено комплексне дослідження щодо використання технологій машинного навчання та комп'ютерного зору в інтелектуальних системах моніторингу безпеки. Проаналізовано сучасний стан транспортного менеджменту в умовах урбанізації, визначено вимоги до систем автоматичного розпізнавання номерних знаків та обґрунтовано необхідність впровадження інтелектуальних підходів для підвищення ефективності систем відеоспостереження. Вивчено функціональні можливості існуючих інтелектуальних транспортних систем (ITS), а також їх обмеження в контексті задач дорожнього моніторингу. Здійснено порівняльний аналіз сучасних технічних засобів і програмного забезпечення, придатного для реалізації подібних систем, що дозволило обґрунтовано обрати стек технологій, до якого увійшли Python, OpenCV, нейронна мережа YOLO та OCR-бібліотеки Tesseract і EasyOCR.

У рамках практичної частини розроблено архітектуру та реалізовано програмний модуль, який виконує виявлення транспортних засобів та розпізнавання номерних знаків у відеопотоці. Проведено тестування розробленої системи на реальних відеоданих, наближених до умов експлуатації. Отримані результати засвідчили доцільність застосування обраного підходу та підтвердили ефективність інтеграції методів комп'ютерного зору та глибокого навчання в системи безпеки. Зокрема, система продемонструвала здатність до обробки відео в реальному часі, виявлення транспортних засобів з високою точністю та задовільні результати оптичного розпізнавання символів на номерних знаках.

Практичні результати роботи можуть бути використані як основа для впровадження інтелектуальних підсистем дорожнього моніторингу у межах розумної інфраструктури міста. Запропоноване рішення має потенціал масштабування та подальшого розвитку, зокрема шляхом підвищення стійкості до складних погодних умов, покращення якості зображень у нічний час, а також інтеграції з базами даних органів безпеки для автоматизованого аналізу дорожніх

правопорушень. Отже, результати дослідження підтверджують актуальність теми, обґрунтованість обраного підходу та доцільність подальшого впровадження розробленої системи в практику інтелектуального моніторингу безпеки.

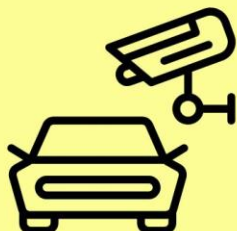
ПЕРЕЛІК ПОСИЛАНЬ

1. Mohanty S. P., Choppali U., Kougiianos E. Everything you wanted to know about smart cities: The Internet of things is the backbone. *IEEE Consumer Electronics Magazine*. 2016. Т. 5, № 3. С. 60–70. URL: <https://doi.org/10.1109/mce.2016.2556879>.
2. Complex Solution of Ecological and Economic Problems of Traffic Jams / O. M. Matsenko та ін. *Mechanism of an Economic Regulation*. 2019. № 4. С. 6–15. URL: <https://doi.org/10.21272/mer.2019.86.01>.
3. Yang H.-C., Chen M.-Q., Lin I.-L. Application of Big Data Analysis of Traffic Accidents and Violation Reports for Improving Traffic Safety. *Sensors and Materials*. 2024. Т. 36, № 3. С. 1243. URL: <https://doi.org/10.18494/sam4868>.
4. Bischoff J., Nagel K. Integrating explicit parking search into a transport simulation. *Procedia Computer Science*. 2017. Т. 109. С. 881–886. URL: <https://doi.org/10.1016/j.procs.2017.05.414>.
5. An Intelligent IoT Based Traffic Light Management System: Deep Reinforcement Learning / S. Damadam та ін. *Smart Cities*. 2022. Т. 5, № 4. С. 1293–1311. URL: <https://doi.org/10.3390/smartcities5040066> (дата звернення: 08.05.2025).
6. Zavadko V. Intelligent Traffic Management Systems: A Lowdown of Software & Hardware Components. *intellias*. 2022. URL: <https://intellias.com/intelligent-traffic-management/>.
7. Seo S. B., Singh D. Smart Town Traffic Management System using LoRa and Machine Learning Mechanism. *IEEE Technology Policy and Ethics*. 2018. Т. 3, № 6. С. 1–4. URL: <https://doi.org/10.1109/ntpe.2018.9778109>.
8. Haykin S. S. Neural networks: A comprehensive foundation. Englewood Cliffs, N.J : Prentice Hall, 1994. 700 с.
9. MCCULLOCH W., PITTS W. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biology*. 1990. Т. 52, № 1-2. С. 99–115. URL: [https://doi.org/10.1016/s0092-8240\(05\)80006-0](https://doi.org/10.1016/s0092-8240(05)80006-0) (дата звернення: 14.05.2025).
10. Nielsen M. Neural Networks and Deep Learning. 2015. URL: <http://neuralnetworksanddeeplearning.com/acknowledgements.html>.

11. Johnson J. M., Khoshgoftaar T. M. Survey on deep learning with class imbalance. *Journal of Big Data*. 2019. Т. 6, № 1. URL: <https://doi.org/10.1186/s40537-019-0192-5> (дата звернення: 14.05.2025).
12. Vázquez F. Deep Learning made easy with Deep Cognition. 2017. <https://becominghuman.ai/deep-learning-made-easy-with-deep-cognition-403fbe445351>.
13. Ajitesh Kumar A. Different Types of CNN Architectures Explained: Examples. 2023. URL: <https://vitalflux.com/different-types-of-cnn-architectures-explained-examples/>.
14. Zhang Z., Wang Y. A CNN-based license plate recognition system using real-time video // *Journal of Visual Communication and Image Representation*. – 2021. – Vol. 78. – Article ID: 103121. – DOI: [10.1016/j.jvcir.2021.103121](https://doi.org/10.1016/j.jvcir.2021.103121)
15. License-Plate-Recognition-YOLOv7-and-CNN [Електронний ресурс]. – Режим доступу: <https://github.com/mrzaizai2k/License-Plate-Recognition-YOLOv7-and-CNN>
16. Rao A. OpenCV Python Tutorial: Computer Vision With OpenCV In Python. *edureka*. 2025. URL: <https://www.edureka.co/blog/python-opencv-tutorial/>.
17. TensorFlow: What is it and How Does it Work?. *botpenguin*. URL: <https://botpenguin.com/glossary/tensorflow>.
18. Порівняння найкращих інструментів машинного навчання: TensorFlow, Keras, Scikit-learn і PyTorch. *zfort*. 2023. URL: <https://www.zfort.com.ua/blog/porivnyannya-naikrashikh-instrumentiv-mashinnogo-navchannya-tensorflow-keras>.
19. Jagtap G. A Comprehensive Guide to Mastering Image Processing using OpenCV. *eumentis*. 2024. URL: <https://eumentis.com/blog/what-is-opencv-and-its-role-in-image-processing>.
20. What is OpenCV?. *youngwonks*. URL: <https://www.youngwonks.com/blog/What-is-OpenCV>.
21. Rashid T. Learn Python in One Day and Learn It Well. LCF Publishing, 2016.

22. Rosebrock A. Deep Learning for Computer Vision with Python. PyImageSearch, 2020. URL: <https://pyimagesearch.com/>.
23. Why Python is the Most Popular Language for Machine Learning?. *Analytics Vidhya*. 2020. URL: <https://www.analyticsvidhya.com/>.
24. Bradski G., Kaehler A. Learning OpenCV 3: Computer Vision in C++ with the OpenCV Library. O'Reilly Media.
25. Wang C.-Y., Bochkovski A., Liao H.-Y. M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv preprint*. 2023. URL: <https://arxiv.org/abs/2207.02696>.
26. DS-2CD7A26G0/P-IZHS Data Sheet. *Hikvision*. 2023. URL: <https://www.hikvision.com/>.
27. CUDA GPUs. *Nvidia Developer*. 2022. URL: <https://developer.nvidia.com/cuda-gpus>.
28. IEEE 802.3ab Standard for Gigabit Ethernet. *IEEE*. 2017. URL: <https://standards.ieee.org/>.
29. DiskStation DS220+ Technical Specifications. *Synology*. 2023. URL: <https://www.synology.com/en-global/products/DS220+>.
30. Unik CCTV. (2020). *Vehicle number plate recognition camera test (LPR / ANPR)* [Video]. YouTube. <https://www.youtube.com/watch?v=dP1UQ59d0Vw>
31. Ultralytics. YOLOv8: State-of-the-art object detection. Ultralytics Documentation. 2023. URL: <https://docs.ultralytics.com>
32. JaidevAI. EasyOCR: Ready-to-use OCR with 80+ languages. GitHub. 2020. URL: <https://github.com/JaidevAI/EasyOCR>

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація)



ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-
КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

НАВЧАЛЬНО-НАУКОВИЙ
ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ

КАФЕДРА ІНФОРМАЦІЙНИХ
СИСТЕМ ТА ТЕХНОЛОГІЙ

ІНТЕЛЕКТУАЛЬНА СИСТЕМА МОНІТОРИНГУ ДОРОЖНЬОГО ТРАФІКУ У SMART CITY

КИЇВ 2025

ПРЕЗЕНТАЦІЯ ДО ЗАХИСТУ БАКАЛАВРСЬКОЇ РОБОТИ
СТУДЕНТКИ ГРУПИ ІСД-41
СПЕЦІАЛЬНОСТІ 126 – ІНФОРМАЦІЙНІ СИСТЕМИ ТА
ТЕХНОЛОГІЇ ФЕДОРЧЕНКО МАРІЇ ОЛЕКСАНДРІВНИ

НАУКОВИЙ КЕРІВНИК – СРІБНА ІРИНА МИКОЛАЇВНА
Д. Т. Н., ДОЦЕНТ

МЕТА, ЗАВДАННЯ РОБОТИ



Мета дослідження

Розробка автоматизованої системи розпізнавання номерних знаків на основі технологій комп'ютерного зору та нейромереж для моніторингу дорожнього трафіку в умовах Smart City.



Виконані завдання

- Проведено аналіз концепції Smart City та її ключових компонентів.
- Вивчено актуальні проблеми сучасного транспортного менеджменту.
- Опрацьовано теоретичні основи та принципи функціонування нейронних мереж.
- Обґрунтовано вибір оптимальних інструментів для реалізації системи моніторингу.
- Розроблено архітектуру прототипу інтелектуальної системи дорожнього спостереження.
- Створено програмну модель з використанням комп'ютерного зору та ШІ-алгоритмів.
- Проведено тестування працездатності системи

АКТУАЛЬНІСТЬ ПРОБЛЕМИ ТРАНСПОРТНОГО МЕНЕДЖМЕНТУ



Транспортні затори



Висока аварійність (рис.1.1)



Недостатня кількість паркувальних місць

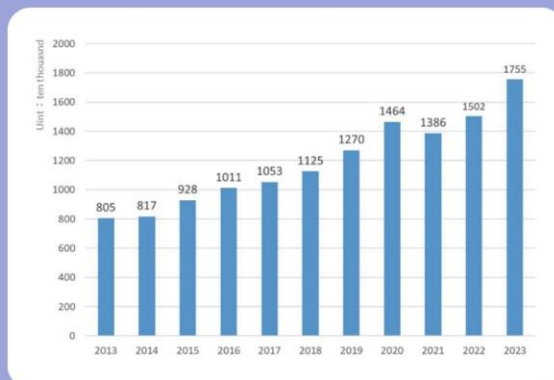


Рисунок 1.1 Зростання кількості порушень ПДР

03

Існуючі системи моніторингу та їх обмеження

04

Існуючі системи:

- Використовують датчики IoT, відеокамери з комп'ютерним зором та AI.
- Збирають і аналізують дані в реальному часі для контролю руху.
- Інтегрують хмарні платформи та **технології LoRa** (рис.1.2).

Функціональні обмеження:

- Високі витрати: розгортання та підтримка дорожньої інфраструктури.
- Проблеми інтеграції: складність модернізації старих транспортних систем.
- Кібербезпека: ризики витоку та несанкціонованого доступу до даних.
- Людський фактор: недовіра до автоматизованих систем.
- Технічні проблеми: залежність від стабільності мережі зв'язку та обладнання.

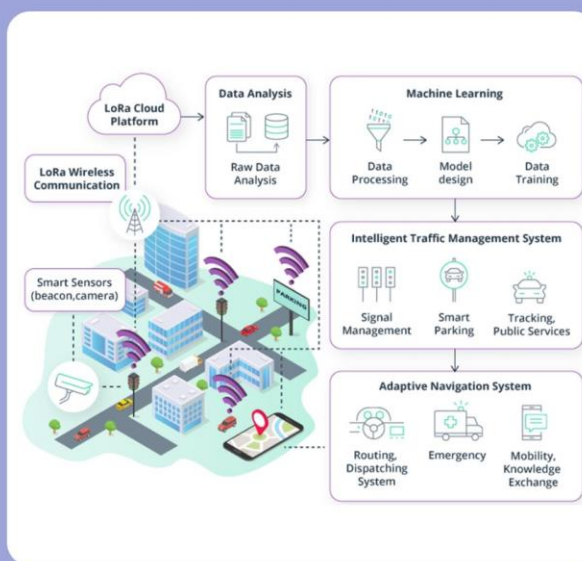


Рисунок 1.2 Типова архітектура систем

04

Переваги інтеграції інтелектуальних систем у міську інфраструктуру:

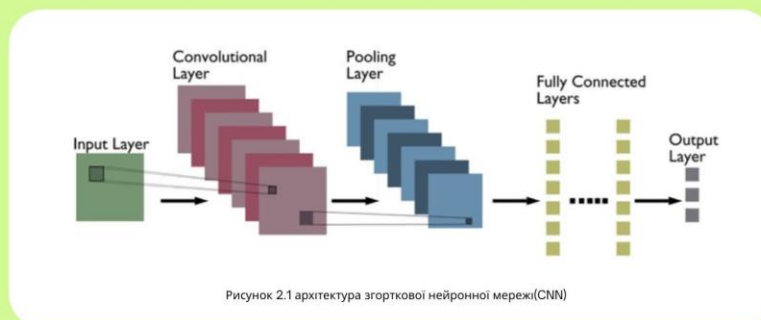
- **Менше заторів** – динамічне регулювання руху, адаптивні світлофори
- **Підвищення безпеки** – виявлення аварій, контроль порушень ПДР
- **Економія часу і пального** – оптимізовані маршрути, менше простоїв
- **Краще паркування** – датчики та камери показують вільні місця
- **Екологічний ефект** – зменшення викидів та шумового забруднення
- **Аналітика для міст** – планування транспортної мережі на основі даних

Місто	Зміни у транспортному потоці (%)	Скорочення часу подорожі (%)	Збільшення використання громадського транспорту (%)
Лондон	-16; -30	-21; -30	+25; +15; +49; +18
Сінгапур	-44; -10; -15	20–30 км/год; 45–65 км/год	N/A
Стокгольм	-20	-33	+5
Мілан	-34; -49	-17; +7; +4.7	N/A
Гетеборг	-10; -2.5	-10; -20	+6
Рим	-20	+4; +5	+5

Таблиця 1.1 Вплив ITS на трафік і транспортні показники мегаполісів

Переваги інтеграції інтелектуальних систем у міську інфраструктуру

05



Штучна нейронна мережа (ШНМ) — це модель, яка імітує роботу людського мозку: вона навчається розпізнавати шаблони, аналізувати дані та приймати рішення.

Основний елемент — **нейрон**, який отримує сигнали, підсумовує їх, і видає результат через активаційну функцію.

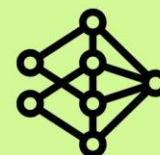
Нейрони об'єднані в шари:

- **вхідний** (отримує дані),
- **прихований** (аналізує),
- **вихідний** (видає результат).

CNN (згорткова нейронна мережа)(Рис.2.1) — це тип глибокої штучної нейронної мережі, спеціально розроблений для розпізнавання образів та обробки зображень.

Ключові поняття та принцип роботи нейромереж

06



Програмне забезпечення:

- Python – проста у використанні мова, ідеальна для задач комп'ютерного зору та ML.
- OpenCV – обробка відео та зображень у реальному часі.
- YOLOv4-tiny – швидке виявлення транспорту.
- Tesseract OCR – розпізнавання номерних знаків.

**Апаратне забезпечення:**

- **IP-камера Hikvision DS-2CD7A26G0/P-IZHS** (рис.2.2) - Забезпечує високу якість зображення (2MP) та нічне бачення - це важливо для точного розпізнавання номерних знаків у будь-який час доби.
- **NAS-сервер Synology DiskStation DS220+** (рис.2.3) - Дозволяє надійно зберігати відеодані та результати розпізнавання, підтримує резервне копіювання й інтеграцію з камерами.
- **Обчислювальний модуль** (ПК або вбудована система) - Процесор Intel Core i5 / AMD Ryzen 5, 8–16 ГБ ОЗП, SSD-диск — забезпечують реальну швидкість обробки відео та стабільну роботу системи.



Рисунок 2.2 Вигляд камери Hikvision



Рисунок 2.3 Вигляд NAS-сервер Synology DiskStation

Вибір технічних інструментів для створення системи

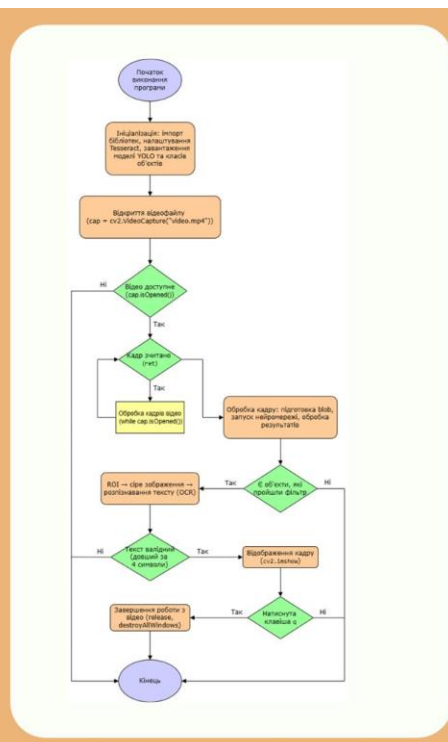
07

АРХІТЕКТУРА ПРОТОТИПУ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ ДОРОЖНЬОГО СПОСТЕРЕЖЕННЯ

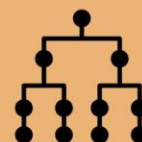
08



Рисунок 2.4 функціональна схема взаємодії компонентів



1.1 Реалізація моделі



Основна логіка обробки

- **Імпорт бібліотек:** cv2, numpy, pytesseract.
- **Завантаження YOLOv4-tiny:** конфіг, ваги, класи (car, bus, truck).
- **Зчитування відео:** з файлу або IP-камери через RTSP.
- **Детекція об'єктів:** конвертація кадру в blob, передача у модель YOLO.
- **Фільтрація об'єктів:** застосування Non-Maximum Suppression для видалення перекриттів.
- **OCR-розпізнавання:** виділення номерного знака, переведення у gray, розпізнавання через Tesseract (--psm 7).
- **Збереження результатів:** текст, вивід на екран або передача в інші системи.

09

```

1 import cv2
2 import pytesseract
3 import numpy as np
4
5
6 pytesseract.pytesseract.tesseract_cmd = r'C:\Program Files\Tesseract-OCR\tesseract.exe'
7
8
9 net = cv2.dnn.readNet("yolov4-tiny.weights", "yolov4-tiny.cfg")
10
11 with open("coco.names.txt", "r") as f:
12     classes = [line.strip() for line in f.readlines()]
13
14 layer_names = net.getUnconnectedOutLayersNames()
15
16 cap = cv2.VideoCapture("video.mp4")
  
```

```

24 blob = cv2.dnn.blobFromImage(frame, 1/255.0, (416, 416), swapRB=True, crop=False)
25 net.setInput(blob)
26
27 outputs = net.forward(layer_names)
28 boxes = []
29 confidences = []
30 class_ids = []
31
32 for out in outputs:
33     for detection in out:
34         scores = detection[5:]
35         if len(scores) == 0:
36             continue
37
38         class_id = np.argmax(scores)
39         if class_id >= len(classes):
40             continue
41
42         confidence = scores[class_id]
43         if confidence > 0.5 and classes[class_id] in ['car', 'bus', 'truck']:
44             cx, cy, w, h = (detection[0:4] * np.array([width, height, width, height])).astype("int")
45             x = int(cx - w / 2)
46             y = int(cy - h / 2)
47             boxes.append([x, y, w, h])
48             confidences.append(float(confidence))
49             class_ids.append(class_id)
  
```

10

1.2 Реалізація моделі

Імпорт бібліотек та ініціалізація OCR ->
завантаження неймережі YOLOv4-tiny ->
Підключення відео -> обробка кожного кадру та
виявлення транспорту -> фільтрація об'єктів +
координати

```

63 indexes = cv2.dnn.NMSBoxes(bboxes, confidences, 0.5, 0.4)
64
65 for idx in indexes:
66     i = idx[0] if isinstance(idx, (list, tuple, np.ndarray)) else idx
67     x, y, w, h = bboxes[i]
68     roi = frame[y:y+h, x:x+w]
69
70     if roi.size == 0:
71         continue
72
73     gray = cv2.cvtColor(roi, cv2.COLOR_BGR2GRAY)
74     text = pytesseract.image_to_string(gray, config='--psm 7')
75     text = text.strip()

```

```

67
68 if len(text) > 4:
69     cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)
70     cv2.putText(frame, text, (x, y - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.9, (36, 255, 12), 2)
71     print("Знайдено номер:", text)
72
73     cv2.imshow("Number Plate Detection", frame)
74     if cv2.waitKey(200) == ord('q'):
75         break
76
77 cap.release()
78 cv2.destroyAllWindows()
79

```

11

1.3 Реалізація моделі

Фільтрація об'єктів через NMS → виділення номерного знака → розпізнавання тексту зображення за допомогою OCR → Результат → Завершення

Тестування роботи нейромережі та програмного функціоналу

12

Вхідний кадр (Input): система отримує зображення транспортного засобу з камери. У цьому випадку — кадр з автомобілем у нічний час з увімкненими фарами.

Обробка: за допомогою моделі YOLOv4-tiny виявляється транспортний засіб. Далі виділяється область номерного знака, і до неї застосовується Tesseract OCR для розпізнавання символів.

Вихідний кадр (Output): система успішно виділила автомобіль і наклала на зображення розпізнаний номер — 0В, DB4668, що також було виведено в консолі.

```
Знайдено номер: 0В, DB4668
```

```
Process finished with exit code -1
```

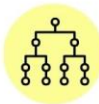


Висновки

13



Проведено аналіз проблем транспортного менеджменту в умовах урбанізації та обґрунтовано актуальність впровадження інтелектуальних систем відеомоніторингу. Обрано оптимальний стек технологій: Python, OpenCV, YOLO, Tesseract/EasyOCR для реалізації задач розпізнавання номерних знаків.



Розроблено архітектуру та створено робочий програмний модуль, який виконує детекцію транспортних засобів і розпізнавання номерів у відеопотоці в реальному часі. Система протестована на практичних прикладах і продемонструвала високу точність та стабільну роботу.



Рішення є масштабованим і придатним до впровадження у міську інфраструктуру. Можливе подальше вдосконалення: покращення нічного бачення, підвищення стійкості до погодних умов, інтеграція з базами даних та автоматичний аналіз правопорушень.

АПРОБАЦІЯ РЕЗУЛЬТАТІВ РОБОТИ



14



Апробація дослідження здійснювалась у формі участі в фахових конференціях:

II всеукраїнська науково-технічна конференція «**Технологічні горизонти: дослідження та застосування інформаційних технологій для технологічного прогресу України і світу**» з поданням тез на тему «**Застосування штучного інтелекту для вивчення Адаптаційних властивостей сільськогосподарських Культур**».

VII міжнародна науково-технічна конференція «**Сучасний стан та перспективи розвитку IoT**» з поданням тез на тему «**Використання технологій машинного навчання та комп'ютерного зору в інтелектуальних системах моніторингу безпеки**».



Рисунок 1.1 компоненти розумного міста



Розумне місто (Smart City) — це міський простір, який використовує інформаційно-комунікаційні технології (ІКТ) для оптимізації ресурсів, підвищення якості життя мешканців, покращення міського управління та забезпечення сталого розвитку.

Основні **компоненти**(рис.1.1): цифрова інфраструктура, інтелектуальний транспорт, енергоефективність, розумне управління, активні громадяни

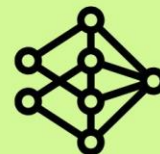
Мета: забезпечити сталий розвиток, комфорт та безпеку міського середовища.

Концептуальні засади та ключові характеристики розумного міста

03

1.1 Ключові поняття та принцип роботи нейромереж

07



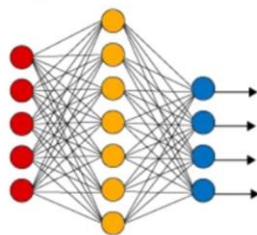
Штучна нейронна мережа (ШНМ) — це модель, яка імітує роботу людського мозку: вона навчається розпізнавати шаблони, аналізувати дані та приймати рішення.

Основний елемент — **нейрон**, який отримує сигнали, підсумовує їх, і видає результат через активаційну функцію.

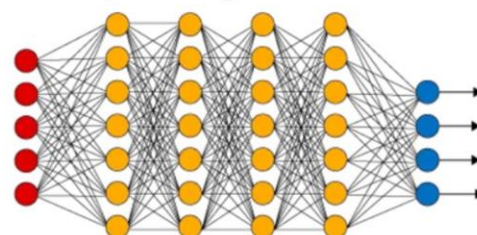
Нейрони об'єднані в шари:

- **вхідний** (отримує дані),
- **прихований** (аналізує),
- **вихідний** (видає результат).

Simple Neural Network



Deep Learning Neural Network



● Input Layer

● Hidden Layer

● Output Layer

Рисунок 2.1 Схема мілких та глибоких нейромереж

Популярні інструменти для розпізнавання зображень



09



TensorFlow

Гнучка бібліотека від Google для глибокого навчання. Підтримує великі проекти, використовується для аналізу відео та зображень у реальному часі.



Keras

Простий інтерфейс до TensorFlow. Підходить для швидкого створення нейронних мереж, зокрема згорткових (CNN).



PyTorch

Бібліотека від Facebook з динамічним графом. Популярна у дослідженнях та складних експериментах.



OpenCV

Основна бібліотека для комп'ютерного зору. Дає змогу працювати із зображеннями та відео, підтримує інтеграцію з CNN, OCR, YOLO тощо.

Постановка задачі та технічні вимоги до системи

12

Постановка задачі:

➤ Розробити систему автоматичного розпізнавання номерних знаків у реальному часі для інтелектуального моніторингу дорожнього трафіку, з можливістю детекції транспортних засобів;

➤ Локалізації номерних знаків;

➤ OCR-розпізнавання символів;

➤ Збереження результатів (зображення + метадані);

Категорія	Вимога
Мова розробки	Python 3.x
Бібліотеки	OpenCV, pytesseract, YOLOv7, NumPy
Камера	Hikvision DS-2CD7A26G0/P-IZHS
Обчислення	Комп'ютер з GPU (підтримка CUDA)
Зберігання даних	NAS Synology DS220+
Формат відео	RTSP-потік
Пропускна здатність	Gigabit Ethernet
Макс. час обробки	≤ 200 мс на кадр
Точність розпізнавання	≥ 85%
Надійність	Автовідновлення після збоїв, стабільна робота 24/7
Масштабованість	Можливість підключення нових камер без зміни архітектури
Безпека	Шифрування даних, обмеження доступу, ведення журналів подій
Сценарії використання	Паркінги, контроль трафіку, безпека об'єктів, аналітика

ДЯКУЮ ЗА УВАГУ!