

В. О. РОМАНЕЦЬ

аспірант кафедри автоматизації
та інтелектуальних інформаційних технологій
Вінницький національний технічний університет
ORCID: 0009-0009-6075-0210

О. В. БІСІКАЛО

доктор технічних наук, професор, завідувач кафедри автоматизації
та інтелектуальних інформаційних технологій
Вінницький національний технічний університет
ORCID: 0000-0002-7607-1943

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ ВІДЕОПОТОКУ ДОРОЖНЬОГО РУХУ НА ОСНОВІ EDGE-CLOUD АРХІТЕКТУРИ

У статті представлено розробку інформаційної технології інтелектуального аналізу відеопотоку дорожнього руху на основі гібридної Edge-Cloud архітектури. Технологія спрямована на забезпечення обробки даних у реальному часі, суттєве зменшення мережевого навантаження та підтримку автономної роботи периферійних пристроїв в умовах обмеженої пропускну здатності мереж. Запропоновано модульну структуру системи, що складається з чотирьох функціональних блоків: інтерфейс користувача, API-рівень, Edge-рівень та Cloud-рівень. Блоки взаємодіють через стандартизовані протоколи (REST, gRPC, MQTT, WebSocket), що забезпечує масштабованість, гнучкість та можливість незалежного розвитку компонентів.

Edge-рівень виконує повний пайплайн обробки відеопотоку безпосередньо на пристрої NVIDIA Jetson Orin Nano: детекцію об'єктів модифікованою моделлю YOLOv8-nano з модулями C3Ghost-CVAM, багатооб'єктний трекінг удосконаленим методом ByteTrack+ та виявлення аномалій дорожнього руху. До хмарного рівня передаються лише компактні метадані (координати, ідентифікатори треків, типи аномалій, статистика), що усуває потребу в передачі великих обсягів відео. Розроблено UML-діаграми послідовності для етапів ініціалізації системи та безперервної обробки відеопотоку, а також діаграму діяльності процесу навчання моделі детекції.

Програмна реалізація базується на сучасному стеку: Python 3.10+, PyTorch 2.1+, Ultralytics YOLOv8 з експортом у TensorRT (FP16/INT8), PostgreSQL з TimescaleDB, Redis для кешування та черг, FastAPI для REST/gRPC інтерфейсів, Docker для контейнеризації. Експериментальна оцінка підтвердила ефективність технології: економію мережевого трафіку на 98–99.8 % (з типових 4–6 Мбіт/с або $\approx 0,5$ –0,75 МБ/с на камеру при повному відеопотоці H.264 1080p@30fps до 8–15 КБ/с метаданих), швидкість обробки 55–75 FPS на Jetson Orin Nano з TensorRT та середнє значення MCC = 0,742 для виявлення аномалій.

Архітектура забезпечує високу модульність, масштабованість та стійкість до збоїв: Edge-пристрій автономно функціонує при втраті з'єднання з Cloud, зберігаючи події в локальному буфері з подальшою синхронізацією. Розроблена технологія має перспективне застосування в інтелектуальних транспортних системах, моніторингу дорожньої безпеки розумних міст та сценаріях з обмеженим доступом до ширококомунікаційних мереж.

Ключові слова: інформаційна технологія, Edge-Cloud архітектура, комп'ютерний зір, детекція об'єктів, багатооб'єктний трекінг, виявлення аномалій, інтелектуальні транспортні системи, YOLOv8, ByteTrack, Jetson Orin Nano, TensorRT.

V. O. ROMANETS

Postgraduate Student at the Department of Automation
and Intelligent Information Technologies
Vinnytsia National Technical University
ORCID: 0009-0009-6075-0210

O. V. BISIKALO

Doctor of Technical Sciences, Professor,
Head of the Department of Automation
and Intelligent Information Technologies
Vinnytsia National Technical University
ORCID: 0000-0002-7607-1943



INFORMATION TECHNOLOGY FOR INTELLIGENT TRAFFIC VIDEO STREAM ANALYSIS BASED ON EDGE-CLOUD ARCHITECTURE

The paper presents the development of an information technology for intelligent analysis of road traffic video streams based on a hybrid Edge-Cloud architecture. The proposed solution is designed to enable real-time data processing, significantly reduce network load, and ensure autonomous operation of edge nodes under conditions of limited bandwidth.

The system is implemented as a modular structure comprising four main functional blocks: user interface, API layer, Edge layer, and Cloud layer. Each block performs clearly defined tasks and interacts via standardized protocols (REST, gRPC, MQTT, WebSocket), providing high scalability, flexibility, and the possibility of independent component evolution.

The Edge layer executes the complete video processing pipeline directly on the NVIDIA Jetson Orin Nano device. The pipeline includes object detection using a modified YOLOv8-nano model enhanced with C3Ghost-CBAM modules, multi-object tracking with an improved ByteTrack+ method, and anomaly detection in road traffic. Only compact metadata (bounding boxes, track IDs, anomaly types, statistics) are transmitted to the Cloud layer, eliminating the need to send large volumes of video data.

UML sequence diagrams were developed for the system initialization phase and continuous video stream processing, as well as an activity diagram for the object detection model training process.

The software implementation is based on a modern technology stack: Python 3.10+, PyTorch 2.1+, Ultralytics YOLOv8 with export to TensorRT (FP16/INT8 quantization), PostgreSQL with TimescaleDB extension for time-series data, Redis for caching and message queuing, FastAPI for REST/gRPC interfaces, and Docker for containerization.

Experimental evaluation confirmed the high efficiency of the technology: network traffic reduction of 98–99.8 % (from typical 4–6 Mbps or ≈ 0.5 – 0.75 MB/s per camera for full H.264 1080p@30fps video stream to 8–15 KB/s of JSON metadata), processing speed of 55–75 FPS on NVIDIA Jetson Orin Nano using TensorRT, and an average Matthews Correlation Coefficient (MCC) of 0.742 for anomaly detection.

The proposed architecture demonstrates strong modularity, scalability, and fault tolerance: the edge device can operate autonomously during temporary loss of Cloud connectivity, storing detected events in a local buffer and automatically synchronizing them upon connection restoration.

The developed technology shows strong potential for application in intelligent transportation systems (ITS), road safety monitoring in smart cities, and scenarios with constrained network bandwidth (4G/5G, remote areas).

Key words: information technology, Edge-Cloud architecture, computer vision, object detection, multi-object tracking, anomaly detection, intelligent transportation systems, YOLOv8, ByteTrack, Jetson Orin Nano, TensorRT.

Постановка проблеми

Системи відеоспостереження становлять основу сучасної інфраструктури забезпечення дорожньої безпеки, ефективного управління транспортними потоками та комплексного моніторингу транспортної мережі. За даними Всесвітньої організації охорони здоров'я, щорічно внаслідок дорожньо-транспортних пригод гине близько 1,19 мільйона осіб, що робить питання автоматизації моніторингу дорожнього руху надзвичайно актуальним [1].

Однак значне зростання обсягів відеоданих, що надходять із розгалуженої мережі камер спостереження, створює значні труднощі для традиційних методів аналізу. Аномальні події є досить рідкісними – за різними оцінками, вони відбуваються менше ніж 1 % часу спостереження, що робить ручний моніторинг неефективним і ресурсозатратним процесом [2]. Традиційні підходи до аналізу відеопотоку передбачають передачу повного відеосигналу до централізованого сервера (від 3 до 6 Mbps на камеру для потоку 1080p@30fps з компресією H.264), що створює значне навантаження на мережеву інфраструктуру та підвищує латентність обробки [3].

Найважливішими проблемами, з якими стикається галузь, є: високі обчислювальні вимоги точних моделей детекції, що унеможливує їх використання на пристроях з обмеженими ресурсами [4]; недостатня стійкість методів багатооб'єктного трекінгу до оклюзій та перемикань ідентичності [5]; обмежена кількість спеціалізованих датасетів для виявлення аномалій у дорожньому русі [6, 7].

Аналіз останніх досліджень і публікацій

За останнє десятиліття інтерес до використання глибокого навчання у системах відеоспостереження значно зріс. Сімейство детекторів YOLO (You Only Look Once), розроблене Redmon та ін. [8], стало де-факто стандартом для детекції об'єктів у реальному часі. YOLOv8, представлений компанією Ultralytics у 2023 році [9], характеризується дуже добрим балансом між точністю та швидкістю інференсу, що сприяло його широкому застосуванню в задачах реального часу. Авторами у роботі [10] запропоновано модифіковану архітектуру YOLOv8, що дозволило зменшити кількість параметрів моделі (на 26 %) без суттєвого зниження якості детекції (менше 1 % mAP50), що дозволяє використовувати метод детекції у edge обробці.

У сфері багатооб'єктного трекінгу значним досягненням став метод ByteTrack [11], який використовує як високо-впевнені, так і низько-впевнені детекції для асоціації об'єктів, що суттєво підвищує стійкість до оклюзій. OS-SORT [12] доповнив цей підхід механізмами компенсації обсервації та віртуальних траєкторій. У роботі [13] автори запропонували модифікацію ByteTrack, що дозволила покращити якість трекінгу, за показниками MOTA та IDF, з мінімальною втратою швидкодії.

Концепція Edge-Cloud архітектури для систем комп'ютерного зору активно досліджується у роботах [14, 15]. Згідно з оглядом [14], edge computing дозволяє розміщувати обчислювальні вузли близько до джерел даних, що забезпечує суттєве зниження латентності завдяки уникненню передачі даних через мережу до віддаленої хмари – наприклад, час передачі кадру камери до хмари може перевищувати 200 мс, тоді як локальна обробка на edge значно скорочує цей показник і робить можливим реальний час обробки. Крім того, локальна обробка даних зменшує обсяг необхідного для передавання трафіку, що особливо важливо для bandwidth-інтенсивних застосувань, таких як аналіз відеопотоків з камер спостереження. Автори підкреслюють, що edge computing забезпечує автономну роботу систем у разі втрати з'єднання з хмарою, оскільки дані обробляються локально, що підвищує стійкість до мережевих збоїв і дозволяє системі продовжувати функціонування без постійного доступу до центрального сервера. Додатковими перевагами є покращення приватності даних (обробка чутливої інформації не виходить за межі локального trusted edge-сервера) та масштабування системи без створення центральних вузьких місць у мережі. Однак питання оптимального розподілу функцій між Edge та Cloud рівнями для задач аналізу дорожнього руху залишається недостатньо дослідженим.

У роботі [16] запропонований фреймворк SAE-MCVT для трекінгу у реальному часі транспортних засобів у багатокамерних системах на базі edge computing. Автори пропонують розподілену архітектуру, де edge-пристрої обробляють відеопотоки локально через модулі детекції (YOLOv11n), трекінгу (ByteTrack) та екстракції ознак (ResNet-50), передаючи лише метадані до центрального сервера. Це забезпечує масштабованість для міського рівня, з IDF1-оцінкою 61.96 на датасеті RoundaboutHD, демонструючи ефективність у подоланні оклюзій та нелінійних траєкторій.

Робота [17] фокусується на семантичній edge-cloud комунікації для реального часу моніторингу трафіку в міських умовах з використанням ViT та LLM. Система детектує регіони інтересу (RoI) за допомогою YOLOv11 на edge пристрої, кодує їх у компактні вектори та передає до cloud для реконструкції зображень і генерації описів трафіку через LLaVA 1.5. Це зменшує обсяг даних на 99.9 %, забезпечуючи точність 89 % у мобільних мережах (з квантизацією для стійкості до помилок у каналах типу AWGN).

Формулювання мети дослідження

Метою роботи є розробка інформаційної технології інтелектуального аналізу відеопотоку дорожнього руху на основі Edge-Cloud архітектури, що забезпечує виявлення аномалій у реальному часі з мінімальним використанням мережевих ресурсів та можливістю автономної роботи Edge-пристроїв.

Основними завданнями дослідження є: проектування загальної структури інформаційної технології з чітким розподілом функцій між Edge та Cloud рівнями; розробка UML-діаграм для формалізації процесів ініціалізації, обробки та навчання; вибір та обґрунтування технологічного стеку для програмної реалізації; експериментальна оцінка ефективності розробленої технології.

Викладення основного матеріалу дослідження

Проектування загальної структури інформаційної технології. Інформаційна технологія аналізу відеопотоку дорожнього руху реалізується як сукупність взаємопов'язаних модулів, згрупованих у чотири функціональні блоки: інтерфейс користувача, API-рівень, Edge-рівень та Cloud-рівень. Розрізнення блоків забезпечує масштабованість і гнучкість системи: обчислювальна частина на Edge-рівні може незалежно еволюціонувати від інтерфейсу користувача та хмарних шарів зберігання й аналітики. Загальна структурна схема розробленої інформаційної технології наведена на рис. 1.

Блок «Інтерфейс користувача» забезпечує взаємодію оператора з системою та включає: модуль налаштування ROI та калібрації камери (вхідні дані: координати зон інтересу, параметри калібрації, URL камер; вихідні дані: конфігураційний файл для Edge-рівня); модуль візуалізації (вхідні дані: агреговані треки, теплові карти, статистика; вихідні дані: візуальні елементи, експортовані звіти PDF/CSV).

Блок «API-рівень» забезпечує комунікацію між компонентами системи та включає: модуль аутентифікації та авторизації (JWT-токени з ролями: оператор, аналітик, адміністратор); модуль обробки запитів від Edge (валідація, фільтрація, нормалізація через gRPC/REST); модуль керування чергою завдань (Kafka/RabbitMQ); модуль моніторингу стану Edge-вузлів.

Блок «Edge-рівень» є базовим компонентом системи, на якому виконується повний пайплайн обробки відеопотоку. Структуру модулів Edge-рівня наведено в таблиці 1.

Блок «Cloud-рівень» забезпечує централізоване зберігання, аналітику та сповіщення: модуль агрегації даних з множини Edge-пристроїв; база даних PostgreSQL з розширеннями TimescaleDB (часові ряди) та PostGIS (геопросторові запити); аналітичний модуль (Web UI Dashboard); система сповіщення (email/SMS/Slack).

Розробка UML-діаграм інформаційної технології. Функціонування інформаційної технології складається з двох послідовних етапів: етапу ініціалізації та налаштування (виконується одноразово) та етапу безперервної обробки відеопотоку в режимі реального часу (виконується циклічно).

Етап ініціалізації виконується адміністратором системи один раз при встановленні камери або при зміні умов сцени. Мета цього етапу – визначити геометричні параметри сцени, провести калібрування камери та встановити

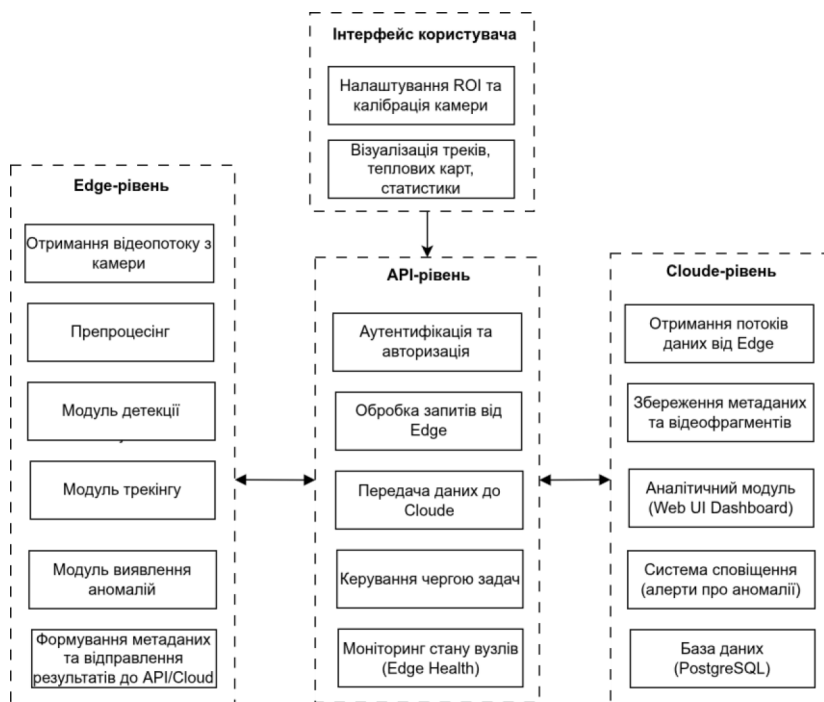


Рис. 1. Загальна структурна схема розробленої інформаційної технології аналізу відеопотоку дорожнього руху

Таблиця 1

Модулі Edge-рівня інформаційної технології

Модуль	Вхідні дані	Вихідні дані
Захоплення відеопотоку	RTSP/RTMP-потоки, локальні файли	Кадри з метаданими (timestamp, camera_id)
Препроцесінг	Сирі кадри	Нормалізовані кадри 640 × 640
Детекція (YOLOv8 + C3Ghost-CBAM) [10]	Препроцесовані кадри, ROI	Bounding boxes, класи, confidence
Трекінг (ByteTrack+) [13]	Детекції послідовних кадрів	Треки з track_id, швидкістю
Виявлення аномалій [6]	Треки, калібрація	Події за типом, timestamp
Формування метаданих	Треки, аномалії, статистика	JSON/Protobuf для Cloud

пороги для виявлення аномалій. UML-діаграма послідовності етапу ініціалізації включає наступні кроки: завантаження репрезентативного кадру; вибір опорних точок для гомографічного перетворення; визначення стоп-лінії, ROI перехрестя та зони світлофора; встановлення порогів (v_max, a_emergency, t_roi_max); генерація та збереження конфігураційного файлу.

Етап обробки відеопотоку виконується циклічно для кожного кадру. UML-діаграма послідовності включає взаємодію модулів: VideoCapture → Preprocessor → YOLOv8Detector → ByteTrackPlus → AnomalyDetector → MQTTPublisher.

UML-діаграма діяльності процесу навчання моделі детекції формалізує послідовність: підготовка датасету (11400 зображень) → конфігурація моделі YOLOv8-nano з C3Ghost-CBAM → навчання (100 epoch, batch_size = 16) → валідація → експорт в ONNX → INT8-квантизація → тестування на Edge-пристрої.

Програмна реалізація інформаційної технології. Вибір програмних засобів здійснювався на основі критеріїв продуктивності, масштабованості, наявності документації та вартості. Порівняння фреймворків інференсу для Edge-рівня наведено в таблиці 2.

Таблиця 2

Порівняння фреймворків інференсу на Edge-пристрої

Фреймворк	FPS	RAM, ГБ	Розмір, МБ	Вибір
PyTorch Mobile	18–28	1.1–1.5	6	–
ONNX Runtime	15–22	0.8–1.2	12	–
TensorFlow Lite	20–30	0.6–1.0	8–10	–
TensorRT (float16)	46–65	0.8–1.2	10–15	✓
TensorRT (int8)	55–80	0.6–1.0	8–12	✓✓

За результатами аналізу та тестування на Jetson Orin Nano найкращим вибором для інференсу модифікованої моделі YOLOv8n визнано TensorRT (з квантизацією FP16 або INT8). Цей фреймворк забезпечує суттєво вищу швидкість обробки (55–75 FPS залежно від режиму квантизації), помірне споживання оперативної пам'яті (≈ 0.8 – 1.2 ГБ) та компактний розмір експортованої моделі (≈ 10 – 15 МБ). ONNX Runtime, хоча й демонструє стабільні 18–25 FPS у Python-реалізації, поступається TensorRT за продуктивністю на апаратній платформі NVIDIA Jetson. TensorRT також краще інтегрується з екосистемою NVIDIA (CUDA, cuDNN, TensorRT Execution Provider), що дозволяє досягти максимальної ефективності на Orin Nano. Технологічний стек Cloud-рівня включає: Python 3.10+, PyTorch 2.1+, Ultralytics YOLOv8 (з експортом у TensorRT), PostgreSQL з розширенням TimescaleDB для часових рядів, Redis для кешування та управління чергами, FastAPI (замість Flask) для REST/gRPC API, Docker + Docker Compose для контейнеризації та оркестрації.

Основною перевагою Edge-Cloud архітектури є економія мережевого трафіку. Порівняння обсягу даних наведено в таблиці 3.

Таблиця 3

Порівняння мережевого трафіку

Варіант передачі	На камеру	16 камер
Повний відеопотік H.264 1080p@30fps	3–6 Mbps (0.375–0.75 MB/s)	48–96 Mbps (6–12 MB/s)
Метадані JSON (запропоновано)	8–15 KB/s	128–240 KB/s
Економія трафіку	98–99.8 %	

Запропонований підхід забезпечує зменшення мережевого навантаження на 98–99.8 % (залежно від інтенсивності трафіку та кількості об'єктів на кадрі), що робить систему придатною для розгортання в умовах обмеженої пропускної здатності мереж (наприклад, у віддалених районах або при використанні 4G/5G). Додатковою перевагою є автономність Edge-пристрою при втраті з'єднання з Cloud: система продовжує виявляти та локально зберігати аномалії у буфері MQTT (до 10000 подій) з подальшою синхронізацією протягом 15–30 секунд після відновлення з'єднання.

Висновки

У роботі розроблено інформаційну технологію інтелектуального аналізу відеопотоку дорожнього руху на основі Edge-Cloud архітектури. Технологія реалізована як модульна система, що складається з чотирьох функціональних блоків: інтерфейс користувача, API-рівень, Edge-рівень та Cloud-рівень. Кожен блок виконує чітко визначені функції та забезпечує взаємодію через стандартизовані протоколи (REST/gRPC, MQTT, WebSocket), що гарантує масштабованість, гнучкість та легкість інтеграції.

Edge-рівень виконує повний пайплайн обробки відеопотоку в реальному часі безпосередньо на периферійному пристрої NVIDIA Jetson Orin Nano, включаючи:

- детекцію об'єктів модифікованою моделлю YOLOv8-nano з модулями C3Ghost-CBAM;
- багатооб'єктний трекінг удосконаленим методом ByteTrack+;
- виявлення чотирьох типів аномалій дорожнього руху.

До Cloud-рівня передаються лише метадані (координати, ідентифікатори треків, типи аномалій, статистика), що суттєво зменшує мережеве навантаження.

Експериментально підтверджено високу ефективність розробленої технології:

- досягнуто економії мережевого трафіку на 98–99.8 % (з типових 4–6 Mbps або ≈ 0.5 – 0.75 MB/s на камеру при повному відеопотоці H.264 1080p@30fps до 8–15 KB/s метаданих JSON);
- забезпечено швидкість обробки 55–75 FPS (залежно від режиму квантизації FP16/INT8) на NVIDIA Jetson Orin Nano при використанні TensorRT;
- отримано середнє значення коефіцієнта кореляції Метьюза (MCC) = 0.742 для виявлення аномалій, зокрема: проїзд на червоний сигнал – 0.861, перевищення швидкості – 0.758, аварійне гальмування – 0.515, перебування в зоні ROI – 0.834.

Запропонована архітектура характеризується високим рівнем модульності, масштабованості та стійкості до збоїв: Edge-пристрій здатний автономно працювати при тимчасовій втраті з'єднання з Cloud, зберігаючи виявлені аномалії в локальному буфері (до 10 000 подій) з подальшою автоматичною синхронізацією після відновлення зв'язку.

Отримані результати свідчать про перспективність використання гібридної Edge-Cloud архітектури для побудови сучасних інтелектуальних транспортних систем, особливо в умовах обмеженої пропускної здатності мереж, високих вимог до реального часу обробки та необхідності забезпечення автономності периферійних вузлів.

Розроблена технологія може бути основою для подальшого масштабування на багатокамерні системи, інтеграції з V2X-протоколами, а також розширення функціоналу на виявлення додаткових типів аномалій та прогнозування дорожніх ситуацій.

Список використаної літератури

1. World Health Organization. Global status report on road safety 2023. Geneva : WHO, 2023. 81 p. ISBN 978-92-4-008651-7. URL: <https://www.who.int/publications/i/item/9789240086517> (дата звернення: 09.02.2026).
2. Duong H.-T., Le V.-T., Hoang V. T. Deep learning-based anomaly detection in video surveillance: A survey // *Sensors*. 2023. Vol. 23, No. 11. Art. 5024. DOI: 10.3390/s23115024.
3. Human-Centric Anomaly Detection in Surveillance Videos Using YOLO-World and Spatio-Temporal Deep Learning [Електронний ресурс] / arXiv. 2025. URL: <https://arxiv.org/abs/2510.22056> (дата звернення: 09.02.2026).
4. What are the challenges of AI image processing on edge devices? [Електронний ресурс] / Tencent Cloud. 2025. URL: <https://www.tencentcloud.com/techpedia/125406> (дата звернення: 09.02.2026).
5. Advancing multi-object tracking through occlusion-awareness and trajectory optimization // *Knowledge-Based Systems*. 2025. Vol. 310. Art. 112930. DOI: 10.1016/j.knosys.2024.112930.
6. Романець В., Маслій Р. В. Виявлення аномалій у відеопотоці трафіку транспорту засобами комп'ютерного зору та глибинного навчання // *Вісник Вінницького політехнічного інституту*. 2025. Вип. 5. С. 146–155.
7. Pradeep Kumar P., Kant K. TU-DAT: A computer vision dataset on road traffic anomalies // *Sensors*. 2025. Vol. 25, No. 11. Art. 3259. DOI: 10.3390/s25113259. URL: <https://www.mdpi.com/1424-8220/25/11/3259> (дата звернення: 09.02.2026).
8. Redmon J., Divvala S., Girshick R., Farhadi A. You only look once: Unified, real-time object detection // *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016. P. 779–788.
9. Jocher G., Chaurasia A., Qiu J. Ultralytics YOLOv8 [Електронний ресурс]. 2023. URL: <https://github.com/ultralytics/ultralytics> (дата звернення: 09.02.2026).
10. Романець В., Бісікало О. Виявлення об'єктів дорожнього руху з камер відеоспостереження // *Вісник Хмельницького національного університету. Технічні науки*. 2025. Т. 355, № 4. С. 491–497. DOI: 10.31891/2307-5732-2025-355-70
11. Zhang Y., Sun P., Jiang Y. et al. ByteTrack: Multi-object tracking by associating every detection box // *Proceedings of the European Conference on Computer Vision (ECCV)*. 2022. P. 1–21. DOI: 10.1007/978-3-031-20047-2_1
12. Cao J., Weng X., Khirodkar R. et al. Observation-centric SORT: Rethinking SORT for robust multi-object tracking // *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023. P. 9686–9696.
13. Романець В., Бісікало О. Метод багатооб'єктного трекінгу з обробкою оклюзій для систем відеоспостереження // *Вимірювальна та обчислювальна техніка в технологічних процесах*. 2025. Т. 84, № 4. С. 440–445. DOI: 10.31891/2219-9365-2025-84-53
14. Chen J., Ran X. Deep learning with edge computing: A review // *Proceedings of the IEEE*. 2019. Vol. 107, No. 8. P. 1655–1674. DOI: 10.1109/JPROC.2019.2921977
15. Wang X., Han Y., Leung V. C. M., Niyato D., Yan X., Chen X. Convergence of edge computing and deep learning: A comprehensive survey // *IEEE Communications Surveys & Tutorials*. 2020. Vol. 22, No. 2. P. 869–904. DOI: 10.1109/COMST.2020.2970550
16. Lin Y., Lockyer S., Stanek F., Zarbock M., Evans A., Li W., Zhang N. SAE-MCVT: A Real-Time and Scalable Multi-Camera Vehicle Tracking Framework Powered by Edge Computing [Електронний ресурс] / arXiv. 2025. arXiv:2511.13904v1. URL: <https://arxiv.org/pdf/2511.13904> (дата звернення: 09.02.2026).
17. Onsu M. A., Lohan P., Kantarci B., Syed A., Andrews M., Kennedy S. Semantic Edge-Cloud Communication for Real-Time Urban Traffic Surveillance with ViT and LLMs over Mobile Networks [Електронний ресурс] / arXiv. 2025. arXiv:2509.21259v1. URL: <https://arxiv.org/pdf/2509.21259> (дата звернення: 09.02.2026).

References

1. World Health Organization. (2023). Global status report on road safety 2023. <https://www.who.int/publications/i/item/9789240086517>.
2. Duong, H.-T., Le, V.-T., & Hoang, V. T. (2023). Deep learning-based anomaly detection in video surveillance: A survey. *Sensors*, 23(11), Article 5024. <https://doi.org/10.3390/s23115024>
3. Human-centric anomaly detection in surveillance videos using YOLO-World and spatio-temporal deep learning. (2025). *arXiv*. <https://arxiv.org/abs/2510.22056>
4. What are the challenges of AI image processing on edge devices? (2025). Tencent Cloud. <https://www.tencentcloud.com/techpedia/125406>
5. Advancing multi-object tracking through occlusion-awareness and trajectory optimization. (2025). *Knowledge-Based Systems*, 310, Article 112930. <https://doi.org/10.1016/j.knosys.2024.112930>
6. Romanets, V., & Masliy, R. V. (2025). Vyyavlennya anomalii u videopototsi trafiku transportu zasobamy komp'yuternoho zoru ta hlybynnoho navchannya [Anomaly detection in traffic video streams using computer vision and deep learning]. *Visnyk Vinnys'koho Politekhnichnoho Instytutu*, (5), 146–155.

7. Pradeep Kumar, P., & Kant, K. (2025). TU-DAT: A computer vision dataset on road traffic anomalies. *Sensors*, 25(11), Article 3259. <https://doi.org/10.3390/s25113259>
8. Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 779–788).
9. Jocher, G., Chaurasia, A., & Qiu, J. (2023). Ultralytics YOLOv8 [Computer software]. GitHub. <https://github.com/ultralytics/ultralytics>.
10. Romanets, V., & Bisikalo, O. (2025). Vvyavlennya ob'ektiv dorozhn'oho rukhu z kamer videosposterezhennya [Detection of road traffic objects from surveillance cameras]. *Visnyk Khmel'nyts'koho Natsional'noho Universytetu. Tekhnichni Nauky*, 355(4), 491–497. <https://doi.org/10.31891/2307-5732-2025-355-70>
11. Zhang, Y., Sun, P., Jiang, Y., et al. (2022). ByteTrack: Multi-object tracking by associating every detection box. In *Proceedings of the European Conference on Computer Vision (ECCV)* (pp. 1–21). https://doi.org/10.1007/978-3-031-20047-2_1
12. Cao, J., Weng, X., Khirodkar, R., et al. (2023). Observation-centric SORT: Rethinking SORT for robust multi-object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 9686–9696).
13. Romanets, V., & Bisikalo, O. (2025). Metod bahatoob'yektnoho trekingu z obrobkoyu oklyuziy dlya system videosposterezhennya [Method of multi-object tracking with occlusion handling for surveillance systems]. *Їммірувал'на та Обчислювал'на Техніка в Технологічних Протсесах*, 84(4), 440–445. <https://doi.org/10.31891/2219-9365-2025-84-53>
14. Chen, J., & Ran, X. (2019). Deep learning with edge computing: A review. *Proceedings of the IEEE*, 107(8), 1655–1674. <https://doi.org/10.1109/JPROC.2019.2921977>
15. Wang, X., Han, Y., Leung, V. C. M., Niyato, D., Yan, X., & Chen, X. (2020). Convergence of edge computing and deep learning: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 22(2), 869–904. <https://doi.org/10.1109/COMST.2020.2970550>
16. Lin, Y., Lockyer, S., Stanek, F., Zarbock, M., Evans, A., Li, W., & Zhang, N. (2025). SAE-MCVT: A real-time and scalable multi-camera vehicle tracking framework powered by edge computing. *arXiv*. <https://arxiv.org/abs/2511.13904>
17. Onsu, M. A., Lohan, P., Kantarci, B., Syed, A., Andrews, M., & Kennedy, S. (2025). Semantic edge–cloud communication for real-time urban traffic surveillance with ViT and LLMs over mobile networks. *arXiv*. <https://arxiv.org/abs/2509.21259>

Дата першого надходження статті до видання: 15.01.2026

Дата прийняття статті до друку після рецензування: 19.02.2026

Дата публікації (оприлюднення) статті: 30.04.2026