

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

**Навчально-науковий інститут телекомунікаційних систем
Кафедра телекомунікацій**

«На правах рукопису»
УДК _____

«До захисту допущено»
Завідувач кафедри
_____ Сергій КРАВЧУК
«__» _____ 2025 р.

**Магістерська дисертація
на здобуття ступеня магістра
за освітньо-професійною програмою «Інженерія та програмування
інфокомунікацій»
зі спеціальності 172 «Електронні комунікації та радіотехніка»
на тему: «Автоматизований аналіз відеопотоків у системах
електронних комунікацій з використанням методів комп'ютерного
зору»**

Виконав:
студент II курсу, групи ЦК-41 мп
Лобода Артем Ігорович _____

Керівник:
Доцент кафедри ТК НН ІТС, к.т.н., доцент
Явіся В.С. _____

Рецензент:
Доцент кафедри ІТТ НН ІТС, к.т.н., доцент
Правило В.В. _____

Засвідчую, що у цій магістерській
дисертації немає запозичень з праць
інших авторів без відповідних
посилань.
Студент _____

Київ - 2025 рік

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Навчально-науковий інститут телекомунікаційних систем
Кафедра телекомунікацій

Рівень вищої освіти – другий (магістерський)

Спеціальність – 172 «Електронні комунікації та радіотехніка»

Освітньо-професійна програма «Інженерія та програмування інфокомунікацій»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Сергій КРАВЧУК

« ___ » _____ 2025 р.

ЗАВДАННЯ
на магістерську дисертацію студенту
Лободі Артему Ігоровичу

1. Тема дисертації «Автоматизований аналіз відеопотоків у системах електронних комунікацій з використанням методів комп'ютерного зору», науковий керівник дисертації Явіся Валерій Сергійович, к.т.н., доцент, затверджені наказом по університету від «03» листопада 2025 р. № 4772-с.
2. Термін подання студентом дисертації 10.12.2025 р.
3. Об'єкт дослідження: Процес автоматизованого аналізу відеопотоків у системах електронних комунікацій.
4. Предмет дослідження: Методи комп'ютерного зору та алгоритми детекції, трекінгу й виявлення аномальної поведінки, що застосовуються для автоматизованої обробки відеопотоків.
5. Перелік завдань, які потрібно розробити:
 - Проаналізувати сучасні наукові підходи до детекції об'єктів та трекінгу у відеопотоках.

- Дослідити існуючі методи виявлення аномальної поведінки у відео та визначити їх практичні обмеження.
- Обґрунтувати вибір архітектури програмного комплексу для автоматизованого відеоаналізу.
- Реалізувати програмне забезпечення для покадрової обробки відеопотоків з підтримкою детекції, трекінгу та аналізу поведінки.
- Розробити механізми збору статистики, формування текстових і табличних звітів.
- Провести експериментальне тестування програмної системи на відеоданих різної складності.
- Оцінити точність і стійкість алгоритмів, визначити їх переваги та недоліки.

6. Орієнтовний перелік ілюстративного матеріалу:

Слайд №1 – Назва роботи

Слайд №2 – Мета, об'єкт, предмет та завдання дослідження

Слайд №3 – Актуальність теми та практична значущість

Слайд №4 – Структура та логіка дослідження

Слайд №5 – Огляд методів комп'ютерного зору та детекції об'єктів

Слайд №6 – Архітектура моделей YOLO для відеоаналізу

Слайд №7 – Алгоритм трекінгу об'єктів (центроїдний трекер)

Слайд №8 – Алгоритм виявлення аномальної поведінки

Слайд №9 – Загальна архітектура програмного комплексу відеоаналізу

Слайд №10 – Структура модулів програми (детекція, трекінг, статистика)

Слайд №11 – Інтерфейс перегляду відеопотоку (демо OpenCV-вікон)

Слайд №12 – Приклади статистичних звітів, що генерує система

Слайд №13 – Результати аналізу відео з різних типів сцен

Слайд №14 – Ефективність та обмеження алгоритмів

Слайд №15 – Стартап-проект: концепція, ринок, цінність продукту

7. Дата видачі завдання “7” вересня 2024 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Строк виконання етапів магістерської дисертації	Примітка
1	Отримання завдання, формування теми, пошук наукових та технічних джерел	11.10.2024 - 18.10.2024	Виконано
2	Підготовка вступу. Формування мети, об'єкта, предмета, завдань	20.10.2024-23.10.2024	Виконано
3	Огляд методів комп'ютерного зору, детекції та трекінгу	02.11.2024-13.11.2024	Виконано
4	Аналіз методів виявлення аномальної поведінки у відеопотоках	15.11.2024-23.11.2024	Виконано
5	Проектування архітектури програмного комплексу та вибір технологій	27.11.2024-23.12.2024	Виконано
6	Реалізація алгоритмів детекції, трекінгу та аномалій. Створення статистичних модулів	24.03.2025-03.10.2025	Виконано
7	Проведення експериментів, збір статистики, аналіз точності та результатів	07.10.2025-12.11.2025	Виконано
8	Розроблення стартап-проекту, SWOT-аналізу та маркетингової програми	13.11.2025-16.11.2025	Виконано
9	Формулювання висновків, остаточне оформлення дисертації та підготовка презентації	17.11.2025-24.11.2025	Виконано

Студент

Артем ЛОБОДА

Науковий керівник дисертації

Валерій ЯВІСЯ

РЕФЕРАТ

Магістерська дисертація викладена на 113 сторінках та включає 15 ілюстрацій, 18 таблиць та 25 джерел за переліком посилань.

Актуальність теми: Зростання обсягів відеоданих у системах електронних комунікацій та необхідність оперативного виявлення небезпечних ситуацій зумовлюють потребу у створенні інтелектуальних засобів автоматизованого аналізу відеопотоків. Сучасні системи відеоспостереження часто обмежуються лише записом інформації та не здатні самостійно розпізнавати аномальні події. Тому розробка програмного комплексу, що поєднує детекцію об'єктів, трекінг та аналіз поведінкових ознак у реальному часі, є актуальною задачею для галузі телекомунікацій та систем безпеки.

Сутність наукової проблеми: У сучасних телекомунікаційних системах відсутні універсальні та доступні рішення для поведінкового відеоаналізу, здатні працювати на звичайних камерах і не вимагати складної інфраструктури. Більшість існуючих алгоритмів або не забезпечують достатньої точності, або вимагають значних обчислювальних ресурсів. Це створює науково-практичну проблему, пов'язану з пошуком оптимального поєднання моделей комп'ютерного зору та алгоритмів аналізу поведінки.

Зв'язок роботи з науковими програмами: Тематика роботи відповідає науково-дослідним напрямам кафедри телекомунікацій, що зосереджені на розвитку інтелектуальних інфокомунікаційних систем, методів цифрової обробки мультимедійних даних та застосуванні алгоритмів штучного інтелекту в засобах моніторингу й безпеки. Дисертація узгоджується з сучасними державними програмами у сфері цифрової трансформації, зокрема впровадженням систем автоматизованого відеоспостереження, розвитку Smart City-рішень та застосуванням технологій штучного інтелекту у публічній безпеці. Отримані результати органічно інтегруються в науково-прикладний напрям кафедри щодо

створення інтелектуальних систем аналізу відео, засобів поведінкового моніторингу та технологій комп'ютерного зору.

Мета дослідження: Створити програмний комплекс для автоматизованого аналізу відеопотоків з можливістю детекції об'єктів, трекінгу та виявлення аномальної поведінки у режимі реального часу.

Завдання дослідження:

- здійснити аналіз сучасних методів комп'ютерного зору;
- обґрунтувати вибір моделей та архітектури програмного комплексу;
- реалізувати модулі детекції, трекінгу та поведінкового аналізу;
- розробити інструменти збору статистики та формування звітів;
- провести експериментальне тестування системи на різноманітних відеосценах;
- оцінити точність алгоритмів і визначити можливості їхнього вдосконалення;
- сформулювати стартап-концепцію на основі отриманих результатів.

Об'єкт дослідження: Процес автоматизованого аналізу відеопотоків у системах електронних комунікацій.

Предмет дослідження: Методи комп'ютерного зору та алгоритми поведінкового аналізу, що забезпечують автоматичну детекцію, трекінг та виявлення аномалій у відеопотоці.

Методи дослідження: У роботі використано методи комп'ютерного зору (нейромережеві моделі YOLO), алгоритми багатоканального аналізу, методи евристичного визначення аномальних подій, методи статистичної обробки даних, а також інструменти експериментального моделювання (Python, OpenCV).

Наукова новизна: У дисертації запропоновано комплексний підхід до автоматизованого відеоаналізу, що поєднує детекцію, трекінг та поведінковий аналіз без використання складних глибоких моделей.

Новизна полягає у впровадженні комбінованого евристичного алгоритму оцінювання аномалій, який базується на швидкості руху, кількості взаємодій об'єктів та динаміці сцени. Одночасно реалізовано систему статистичної оцінки активності у відеопотоці, що дозволяє проводити автоматизований аналіз інцидентів.

Практичне значення: Отримані результати можуть бути застосовані у системах відеоспостереження, охоронних комплексах, муніципальних проектах Smart City, аналітичних сервісах для бізнесу. Розроблений програмний комплекс є основою для створення стартап-рішення, здатного працювати на стандартних камерах без додаткового обладнання. Система може бути адаптована до реальних умов та інтегрована в існуючі інформаційно-комунікаційні інфраструктури.

Ключові слова: відеоаналіз, комп'ютерний зір, детекція об'єктів, трекінг, аномальна поведінка, відеоспостереження, YOLO, Python.

ABSTRACT

The explanatory note is laid out on 113 and includes 15 illustrations, 18 tables and 25 sources in the list of references.

Relevance of the topic: The rapid growth of video data in electronic communication systems and the increasing need for timely detection of dangerous or abnormal events require the development of intelligent tools capable of automated video stream analysis. Traditional surveillance systems are limited to recording and manual review, lacking the ability to independently recognize critical situations. Therefore, building a software system that integrates object detection, tracking, and behavioral anomaly analysis in real time is highly relevant to modern telecommunications and security technologies.

Essence of the scientific problem: Many existing video analytics solutions require specialized hardware or offer limited behavioral capabilities. Furthermore, current algorithms often struggle to generalize across different environments or operate in real time under resource constraints. This creates a scientific and practical challenge to design an approach that is computationally efficient, adaptable, and capable of identifying meaningful behavioral deviations without relying on heavy deep-learning models for action recognition.

Relation to scientific programs, plans, and research topics: The topic of this thesis is aligned with the research directions of the Department of Telecommunications, which focus on intelligent infocommunication systems, digital multimedia data processing, and the application of artificial intelligence in monitoring and safety solutions. The work corresponds to national digitalization programs, including the development of Smart City infrastructures, AI-assisted public safety technologies, and modern automated video surveillance systems.

The results of this thesis integrate naturally into the department's applied research activities aimed at creating intelligent video analytics platforms, behavioral monitoring tools, and computer vision solutions. The research

continues and expands existing departmental work in developing efficient signal and data processing methods for contemporary communication systems.

Research aim: To create a software system for automated analysis of video streams that performs object detection, tracking, and anomaly identification in real time.

Research tasks:

- to analyze modern methods of computer vision and behavioral video analytics;
- to justify the choice of architecture and technologies for the software system;
- to implement modules for detection, tracking, and anomaly scoring;
- to develop tools for collecting statistical information and generating reports;
- to conduct experimental evaluation of the system on diverse video scenarios;
- to assess the accuracy and stability of the developed algorithms;
- to design a startup concept based on the obtained results and market analysis.

Object of research: The process of automated video stream analysis in electronic communication systems.

Subject of research: Computer vision methods and behavioral analysis algorithms enabling automatic detection, tracking, and identification of anomalies in video streams.

Research methods: The work employs neural-network-based object detection (YOLO), multi-frame tracking techniques, heuristic behavioral anomaly estimation, statistical data analysis, and experimental evaluation using Python and OpenCV.

Scientific novelty: A combined approach to video stream analysis is proposed, integrating object detection, lightweight tracking, and heuristic

behavioral anomaly assessment based on motion patterns, interaction density, and scene dynamics. The developed system provides a functional mechanism for identifying potentially dangerous activity without relying on computationally expensive action-recognition neural networks.

Practical significance: The results can be applied in surveillance systems, security services, Smart City environments, retail monitoring, and corporate safety infrastructures. The developed software system is compatible with standard IP cameras and does not require specialized hardware, making it suitable for real-world deployment and further startup development. The architecture allows for scaling to multiple video streams and integration into existing communication systems.

Keywords: video analysis, computer vision, object detection, tracking, anomaly detection, surveillance systems, YOLO, Python.

ЗМІСТ

ВСТУП	15
РОЗДІЛ 1	18
ТЕОРЕТИЧНІ ОСНОВИ АВТОМАТИЗОВАНОГО АНАЛІЗУ	
ВІДЕОПОТОКІВ У СИСТЕМАХ ЕЛЕКТРОННИХ КОМУНІКАЦІЙ.....	
1.1. Поняття та класифікація відеопотоків у системах електронних комунікацій.....	19
1.2. Основи комп'ютерного зору та алгоритмів обробки відео	22
1.3. Методи глибинного навчання для відеоаналітики (CNN, RNN, Transformer-архітектури).....	26
1.4. Формати та протоколи передавання відео у телекомунікаційних мережах (RTSP, RTP, HLS, WebRTC)	32
Висновки.....	38
РОЗДІЛ 2	41
МЕТОДОЛОГІЯ АВТОМАТИЗОВАНОГО АНАЛІЗУ ВІДЕОПОТОКІВ У РЕАЛЬНИХ ЕЛЕКТРОННИХ КОМУНІКАЦІЙНИХ СИСТЕМАХ.....	
2.1. Методи детекції об'єктів у відеопотоці (YOLO, SSD, EfficientDet) .	42
2.2. Методи трекінгу об'єктів (SORT, DeepSORT, ByteTrack)	47
2.3. Виявлення аномалій у відеопотоці (Autoencoder, ConvLSTM, 3D-CNN підходи)	52
2.4. Моделі оцінки якості відео та QoS/QoE у мережах зв'язку	58
2.5. Архітектури обчислень для аналізу відеопотоків у реальному часі (Edge, GPU, TPU та хмарні рішення).....	63
Висновки.....	69
РОЗДІЛ 3	72
ПРАКТИЧНА РЕАЛІЗАЦІЯ СИСТЕМИ АВТОМАТИЗОВАНОГО АНАЛІЗУ ВІДЕОПОТОКІВ ЗАСОБАМИ КОМП'ЮТЕРНОГО ЗОРУ	
3.1 Постановка задачі та вимоги до програмного комплексу автоматизованого відеоаналізу.....	73
3.2 Архітектура та структура розробленого програмного комплексу	78
3.3 Реалізація алгоритмів детекції, трекінгу та виявлення аномальної активності	84

3.4 Результати роботи програмного комплексу автоматизованого аналізу відеопотоків	87
3.5 Аналіз точності, обмежень та перспектив удосконалення програмного комплексу	91
Висновки.....	95
РОЗДІЛ 4.....	97
СТАРТАП ПРОЄКТ	97
4.1 Опис ідеї стартап-проєкту.....	97
4.2 Технологічний аудит	100
4.3 Аналіз ринку.....	102
4.4 SWOT-аналіз стартап-проєкту та маркетингова програма	105
Висновок	108
ЗАГАЛЬНИЙ ВИСНОВОК ПО РОБОТІ.....	109
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	111

ПЕРЕЛІК СКОРОЧЕНЬ

AI	Artificial Intelligence - штучний інтелект
API	Application Programming Interface - програмний інтерфейс взаємодії
AWS	Amazon Web Services - хмарна інфраструктура Amazon
B2B	Business-to-Business - взаємодія «бізнес для бізнесу»
B2C	Business-to-Consumer - взаємодія «бізнес для клієнта»
CNN	Convolutional Neural Network - згорткова нейронна мережа
CPU	Central Processing Unit - центральний процесор
CSV	Comma-Separated Values - формат табличних текстових даних
DVR	Digital Video Recorder - цифровий відеореєстратор
FPS	Frames Per Second - кількість кадрів на секунду
GNN	Graph Neural Network - графова нейронна мережа
GPU	Graphics Processing Unit - графічний процесор
HTTP	HyperText Transfer Protocol - протокол передавання гіпертексту
IP	Internet Protocol - інтернет-протокол
LSTM	Long Short-Term Memory - рекурентна нейромережа довготривалої пам'яті
MVP	Minimum Viable Product - мінімально життєздатний продукт

NVR	Network Video Recorder - мережевий відеореєстратор
REST	Representational State Transfer - архітектурний стиль веб-взаємодії
RTSP	Real-Time Streaming Protocol - протокол потокового передавання
SaaS	Software as a Service - програмне забезпечення як сервіс
SDK	Software Development Kit - набір інструментів для розробників
SLAM	Simultaneous Localization and Mapping - одночасне визначення позиції та побудова карти
SWOT	Strengths, Weaknesses, Opportunities, Threats - стратегічний матричний аналіз
UI/UX	User Interface / User Experience - інтерфейс і користувацький досвід
YOLO	You Only Look Once - модель нейромережевої детекції об'єктів

ВСТУП

Стрімка цифровізація сучасного суспільства та зростання обсягів мультимедійного трафіку формують потребу у створенні нових високоефективних методів автоматизованої обробки відеоінформації. Відеопотоки стали ключовим елементом інфокомунікаційних систем - від інтелектуального міського моніторингу та роботизованих платформ до систем безпеки, транспортної інфраструктури, дистанційної аналітики та низки сервісів реального часу. У цих умовах особливої актуальності набувають методи комп'ютерного зору й глибинного навчання, які дають змогу виявляти події, розпізнавати об'єкти, здійснювати класифікацію чи пошук аномалій у відеоданих без участі оператора.

Актуальність теми зумовлена тим, що традиційні системи відеоспостереження є здебільшого пасивними та вимагають значних людських ресурсів для інтерпретації отриманих даних, тоді як сучасні телекомунікаційні мережі генерують терабайти відео щодня. Використання автоматизованих алгоритмів відеоаналізу дозволяє різко підвищити ефективність прийняття рішень, зменшити затримки, забезпечити масштабованість та підвищити рівень безпеки інформаційних систем. Методи комп'ютерного зору у поєднанні з архітектурами глибинних нейронних мереж (CNN, RNN, 3D-CNN, Transformer) створюють основу для побудови інтелектуальних платформ, здатних працювати у режимі реального часу навіть на обмежених обчислювальних ресурсах, включаючи edge-пристрої.

Значущість обраного напрямку визначається широкими перспективами його практичного застосування у телекомунікаційній галузі. Відеоаналітика все активніше інтегрується у бездротові інфокомунікаційні системи, мобільні мережі п'ятого покоління, хмарні сервіси та системи з

обчисленнями на периферії. Саме автоматизація процесів виявлення, фільтрації та оцінювання подій у відеопотоці дозволяє оптимізувати навантаження на мережі, зменшити затримки, підвищити якість обслуговування (QoS/QoE) та забезпечити високу надійність критично важливих сервісів.

Зв'язок роботи з науковими програмами та темами кафедри полягає у відповідності дослідження стратегічним напрямкам розвитку інтелектуальних телекомунікаційних систем, технологій мультимедійних сервісів, систем штучного інтелекту та автоматизованого моніторингу.

Зв'язок роботи з науковими програмами, планами та темами кафедри полягає в тому, що дослідження відповідає стратегічним напрямкам розвитку інтелектуальних інфокомунікаційних систем, мультимедійних сервісів та технологій обробки відеоданих у електронних комунікаціях. Тематика дисертації узгоджується з комплексом науково-дослідних робіт кафедри, спрямованих на створення методів автоматизованого аналізу даних, дослідження якості передавання інформації, впровадження алгоритмів комп'ютерного зору та глибинного навчання у телекомунікаційних системах.

Мета роботи полягає у розробленні та дослідженні автоматизованої системи аналізу відеопотоків у електронних комунікаційних мережах із використанням методів комп'ютерного зору та глибинного навчання. Для досягнення зазначеної мети необхідно розв'язати такі завдання:

- виконати аналіз теоретичних принципів побудови систем відеоаналітики та сучасних підходів комп'ютерного зору;
- дослідити алгоритми детекції, трекінгу та виявлення аномалій у відеопотоці;

- обґрунтувати вибір форматів, протоколів та архітектур обробки відео в телекомунікаційних мережах;
- розробити програмний прототип системи автоматизованого аналізу відеопотоків на Python з використанням OpenCV та PyTorch;
- провести тестування системи на реальних та демонстраційних відеоданих, оцінити точність, швидкодію та ефективність застосованих алгоритмів.

У дослідженні застосовуються бібліотеки OpenCV, PyTorch, NumPy, Matplotlib, а також інструменти для роботи з відеопотоками у форматах RTSP/RTP, що дозволяє реалізувати повний цикл обробки: від отримання відео до формування структурованих аналітичних результатів. Розроблене програмне забезпечення орієнтоване на роботу в реальному часі та може бути інтегроване у системи моніторингу, безпеки, транспортної аналітики, телекомунікаційні платформи та інтелектуальні інфраструктури.

Практична значущість роботи полягає у можливості застосування розробленої системи в телекомунікаційних операторах, мережах відеонагляду, промислових рішеннях, а також у наукових дослідженнях, освітніх лабораторіях та стартап-проектах, орієнтованих на аналіз відеопотоків у реальному часі. Отримані результати сприяють підвищенню ефективності існуючих відеосистем та створюють основу для подальшого розвитку інтелектуальних технологій у галузі електронних комунікацій.

РОЗДІЛ 1

ТЕОРЕТИЧНІ ОСНОВИ АВТОМАТИЗОВАНОГО АНАЛІЗУ ВІДЕОПОТОКІВ У СИСТЕМАХ ЕЛЕКТРОННИХ КОМУНІКАЦІЙ

Сучасні електронні комунікаційні системи характеризуються стрімким зростанням обсягу відеоданих, що передаються, зберігаються та обробляються у реальному часі. Інтелектуальні транспортні системи, мережі відеоспостереження, хмарні сервіси, мобільні мережі та промислові інфраструктури формують величезні відеопотоки, для ефективної обробки яких необхідні методи автоматизованого аналізу. Упродовж останніх років у науковій літературі було запропоновано широкий спектр алгоритмів комп'ютерного зору та глибинного навчання, здатних виконувати детекцію, трекінг, розпізнавання об'єктів та виявлення аномалій. Проте попри значну кількість досліджень, низка ключових питань залишається нерозв'язаною.

По-перше, більшість існуючих рішень орієнтовані на окремі ізольовані задачі відеоаналізу, тоді як для практичного застосування у телекомунікаційних системах потрібні комплексні підходи, що враховують специфіку реальних відеопотоків: мінливу якість кадрів, втрати пакетів, різну частоту кадрів, зміни освітлення, обмеженість обчислювальних ресурсів. По-друге, багато моделей демонструють високу точність у лабораторних умовах, але втрачають ефективність при масштабуванні або роботі зі стрімами з мережевих протоколів типу RTSP/RTP. По-третє, залишається актуальною проблема інтеграції алгоритмів комп'ютерного зору у системи електронних комунікацій з урахуванням вимог до затримки, пропускної здатності та якості обслуговування (QoS/QoE).

Таким чином, попри значний обсяг наукових публікацій, питання створення універсальної, масштабованої, стійкої до мережевих умов та придатної до практичного впровадження системи автоматизованого аналізу

відеопотоків залишається відкритим і потребує додаткового дослідження. Саме це визначає місце даної дисертаційної роботи у загальному науковому контексті: вона спрямована на узагальнення сучасного стану проблеми, виявлення ключових недоліків існуючих підходів та формування теоретичних основ для подальшої практичної реалізації запропонованої системи.

У першому розділі розглядаються фундаментальні поняття відеопотоків у електронних комунікаціях, основи алгоритмів комп'ютерного зору, сучасні методи глибинного навчання, а також особливості мережевих протоколів передавання відео. Такий теоретичний аналіз є необхідним для формування цілісного уявлення про технологічне підґрунтя майбутньої практичної реалізації та забезпечує логічний перехід до другого розділу, де центральну увагу приділено методології та обґрунтуванню вибору підходів до аналізу відеопотоків.

1.1. Поняття та класифікація відеопотоків у системах електронних комунікацій

Відеопотік у сучасних електронних комунікаційних системах розглядається як послідовність кадрів, що передаються у цифровому форматі в реальному або квазіреальному часі з метою моніторингу, аналізу, зберігання чи подальшої обробки. З технологічної точки зору відеопотоки є основним типом мультимедійного трафіку, який генерує значне навантаження на телекомунікаційні мережі, що підтверджується глобальними статистичними звітами щодо домінування відеоконтенту у загальному обсязі переданих даних [16]. Зростання популярності відеосервісів, впровадження високої роздільної здатності (HD/4K/8K), застосування відео в системах безпеки, транспортних інфраструктурах та промисловому інтернеті речей (IIoT) вимагають застосування ефективних методів кодування, передавання та аналізу.

У загальному випадку відеопотоки класифікують за кількома основними ознаками.

За режимом передавання:

- потоки реального часу (real-time streaming), які доставляються з мінімально можливою затримкою та використовують спеціалізовані протоколи, зокрема RTP/RTSP [18];

- адаптивні потоки з буферизацією (adaptive streaming), де застосовуються такі технології, як HLS або DASH, що динамічно змінюють бітрейт залежно від доступного каналу.

За джерелом формування:

- потоки, отримані від стаціонарних камер відеоспостереження, часто характеризуються стабільною частотою кадрів та фіксованим ракурсом;

- потоки від мобільних та динамічних пристроїв, зокрема камер дронів, транспортних засобів, роботизованих платформ, які відзначаються змінними умовами зйомки, значними коливаннями перспективи та освітлення;

- генеровані чи синтетичні відеопотоки, створені у симуляторах для тренування або тестування моделей комп'ютерного зору.

За способом кодування:

- традиційні кодеки (H.264/AVC, H.265/HEVC), що забезпечують компромісі між якістю та стисненням;

- сучасні кодеки (AV1, VVC), спрямовані на оптимізацію передачі відео у високих роздільностях;

- спеціалізовані формати відео для аналізу руху або глибинних даних, зокрема RGB-D, мультиспектральні потоки та тепловізійне відео.

З точки зору комп'ютерного зору відеопотік розглядається як багатовимірний сигнал, що містить як просторову, так і часову інформацію. Просторова складова відповідає за вміст кадру, тоді як часова містить

динаміку змін у сцені. У науковій літературі підкреслюється, що алгоритми глибинного навчання використовують як просторові ознаки (CNN), так і часові залежності (RNN, LSTM, 3D-CNN) для ефективної інтерпретації відеоданих [1], [3], [5], [14].

У телекомунікаційних системах відеопотоки мають специфічні характеристики, зумовлені архітектурою мережі:

- обмеження пропускної здатності, що впливають на необхідність адаптивного стиснення;
- втрати пакетів, які знижують якість відтворення відео та ускладнюють алгоритмічний аналіз;
- зміна затримок, властивих мережам реального часу;
- неоднорідність трафіку, коли відео передається одночасно з іншими типами даних.

Стандарти оцінювання якості відео (наприклад, ITU-T P.1203) визначають параметри, що суттєво впливають на можливість ефективного автоматизованого аналізу, серед яких різкість кадрів, стабільність частоти кадрів, рівень шумів та якість декодування [15]. Для задач комп'ютерного зору важливо забезпечити не лише мінімальні спотворення, але й послідовність кадрів без пропусків, оскільки втрати або раптові зміни кадрів можуть призводити до помилок детекції та трекінгу.

Отже, поняття та класифікація відеопотоків є основою для формування вимог до систем автоматизованого аналізу. Особливості структури відеоданих, способів їх передавання та умов функціонування у мережах зв'язку визначають алгоритмічні та технічні обмеження, що повинні враховуватися при розробленні систем комп'ютерного зору. Саме ці аспекти створюють підґрунтя для подальшого аналізу алгоритмів обробки відео, що розглядатимуться у наступних підрозділах.

1.2. Основи комп'ютерного зору та алгоритмів обробки відео

Комп'ютерний зір є галуззю штучного інтелекту, що вивчає методи автоматизованого отримання, аналізу та інтерпретації інформації зі зображень і відеопотоків. З розвитком глибинного навчання роль комп'ютерного зору суттєво змінилася: від використання інженерно створених ознак до застосування складних нейромережових архітектур, здатних самостійно виявляти закономірності у даних [1], [3], [5].

У телекомунікаційних системах комп'ютерний зір широко застосовується для аналізу потокового відео в режимі реального часу: виявлення об'єктів, відстеження руху, класифікації подій, виявлення аномалій. У контексті автоматизованого аналізу відеопотоків важливо розуміти фундаментальні принципи обробки зображень, архітектури згорткових нейронних мереж та класичні методи, що лягли в основу сучасних підходів.

Класичні алгоритми обробки зображень та ознак

До появи глибинного навчання більшість систем комп'ютерного зору використовували методи інженерного опису ознак. Найпоширенішими є SIFT, SURF, HOG, ORB, які дозволяють виділяти ключові точки, градієнти та особливості об'єктів незалежно від масштабу, повороту або змін освітлення.

Таблиця 1.1

Порівняння класичних методів комп'ютерного зору

Алгоритм	Стійкість до поворотів	Стійкість до змін масштабу	Обчислювальна складність	Тип ознак
SIFT	Висока	Висока	Висока	Гرادієнтні ключові точки
SURF	Висока	Висока	Середня	Хмарні дескриптори
HOG	Середня	Низька	Низька	Гістограми градієнтів
ORB	Середня	Середня	Дуже низька	Бінарні дескриптори

Класичні алгоритми демонструють хорошу продуктивність на статичних зображеннях, однак у відеопотоках їх ефективність обмежена через шум, рух, динамічні зміни сцени та складність їх масштабування для аналізу великої кількості кадрів.

Глибинне навчання у комп'ютерному зорі

Згорткові нейронні мережі (CNN) стали основою сучасних підходів до аналізу зображень і відео. CNN автоматично виділяють ознаки з даних, багаторазово застосовуючи згортки, нормалізацію та активації [1], [5].

Поштовхом до розвитку глибинного навчання у візуальних задачах стала архітектура AlexNet [6], а подальші вдосконалення призвели до появи VGG, ResNet, EfficientNet. Для відео застосовуються також тривимірні згортки (3D CNN) та LSTM-мережі, які враховують часову залежність між кадрами [14].

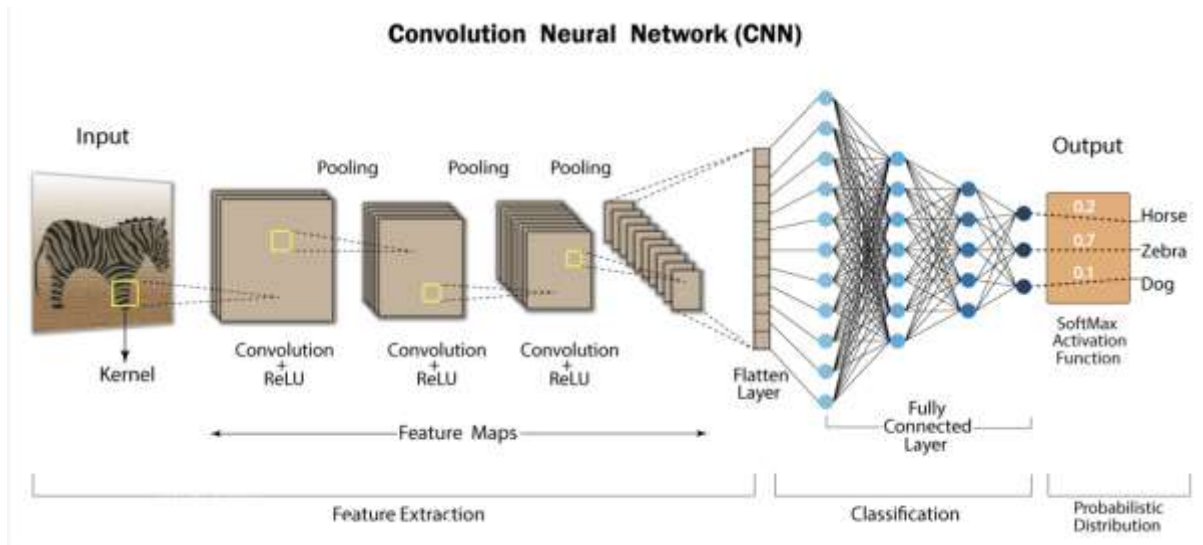


Рис. 1.1 Узагальнена архітектура згорткової нейронної мережі (CNN)

Загальний конвеєр комп'ютерного зору

Більшість алгоритмів відеоаналізу можна подати у вигляді конвеєра (pipeline):

1. Передобробка зображень (нормалізація, шумозниження, зміна контрасту).
2. Виділення ознак або використання CNN для автоматичного вилучення ознак.
3. Класифікація / сегментація / детекція об'єктів.
4. Трекінг або аналіз часової послідовності кадрів.
5. Прийняття рішення чи виявлення події.

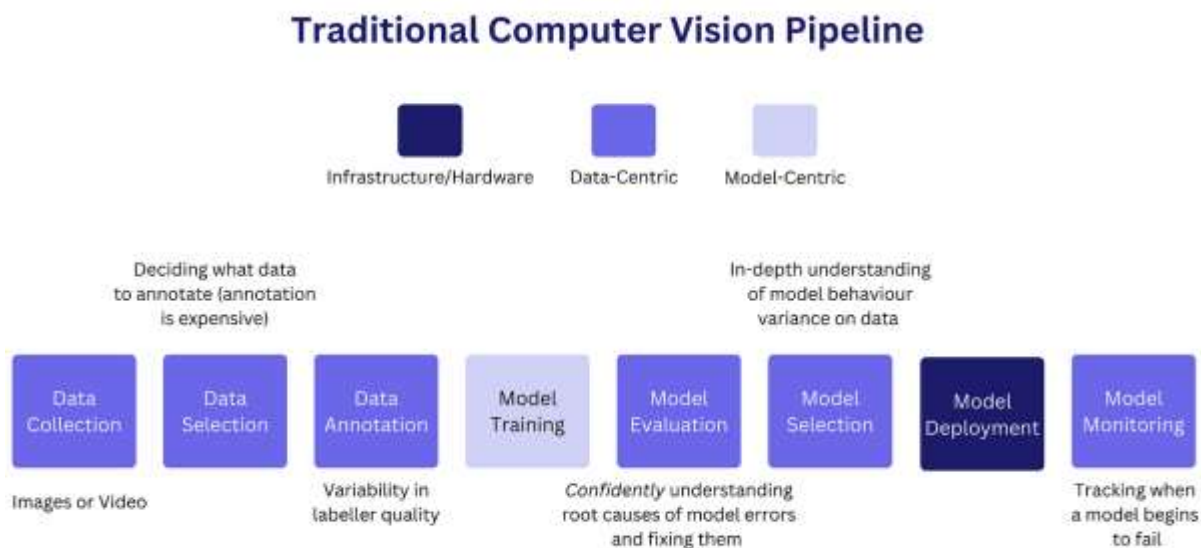


Рис. 1.2 Загальна схема обробки зображень у комп'ютерному зорі (Vision Pipeline)

Порівняння класичних та глибинних методів

Сучасні системи автоматизованого аналізу відео здебільшого базуються на глибинному навчанні, оскільки CNN демонструють кращу точність та стійкість до змін умов.

Таблиця 1.2

Порівняння класичних і глибинних методів комп'ютерного зору

Критерій	Класичні методи	Глибинне навчання
Потреба у ручному виділенні ознак	Так	Ні
Стійкість до шуму та змін середовища	Середня	Висока
Продуктивність на великих наборах даних	Низька	Висока
Адаптивність	Обмежена	Висока (transfer learning)
Швидкодія	Висока без GPU	Висока з GPU/TPU
Застосування у відеоаналізі	Часткове	Повномасштабне

Тривимірні згортки та часовий аналіз відео

Для аналізу відеопотоків важливо враховувати не лише просторову структуру кадрів, а й часову послідовність. Тому використовуються:

- **3D CNN**, де згортки виконуються у трьох вимірах;
- **ConvLSTM**, які об'єднують згортки та LSTM для аналізу руху;
- **Recurrent Neural Networks (RNN)** та їх похідні LSTM/GRU;
- **Transformers**, які набувають популярності у відеоаналізі.

Стислий аналіз стану проблеми

Попри суттєвий прогрес у галузі, існує кілька обмежень, що важливі для подальшої розробки систем автоматизованого аналізу відео:

- CNN-архітектури часто потребують значних обчислювальних ресурсів і не завжди підходять для роботи на edge-пристроях.
- 3D CNN і LSTM моделі мають високу точність, але складні в оптимізації та іноді неспроможні працювати у реальному часі.
- Класичні алгоритми є швидкими, проте недостатньо ефективними у складних відеосценах з динамічними змінами.
- Питання стійкості моделей до низької якості відео, втрат кадрів і мережевих спотворень залишається значною проблемою [15], [16].

1.3. Методи глибинного навчання для відеоаналітики (CNN, RNN, Transformer-архітектури)

Глибинне навчання стало фундаментом сучасних систем комп'ютерного зору, забезпечивши суттєві переваги у точності, масштабованості та стійкості до змін умов освітлення, ракурсів та шумів. Відеоаналітика істотно відрізняється від аналізу статичних зображень тим, що потребує урахування як просторових, так і часових залежностей між кадрами. Саме тому у практичних системах застосовують різні архітектури,

що поєднують CNN, рекурентні мережі, 3D-згортки, а також трансформерні моделі, які останніми роками домінують у задачах sequence modeling [1], [5], [14].

CNN-архітектури для аналізу зображень і відео

Згорткові нейронні мережі (Convolutional Neural Networks, CNN) відіграють ключову роль у виявленні просторових ознак на окремих кадрах відео. В основі CNN лежить операція згортки, яка дозволяє виділяти локальні патерни - контури, текстури, градієнти - без потреби в ручному визначенні ознак, що було характерним для класичних методів [5].

Після появи AlexNet у 2012 році [6] дослідники запропонували низку архітектур, оптимізованих для точності й швидкодії: VGG, Inception, ResNet, EfficientNet. У відеоаналітиці CNN часто використовуються як **feature extractor**, що обробляє кожен кадр окремо, після чого часовий аналіз виконується іншими моделями.

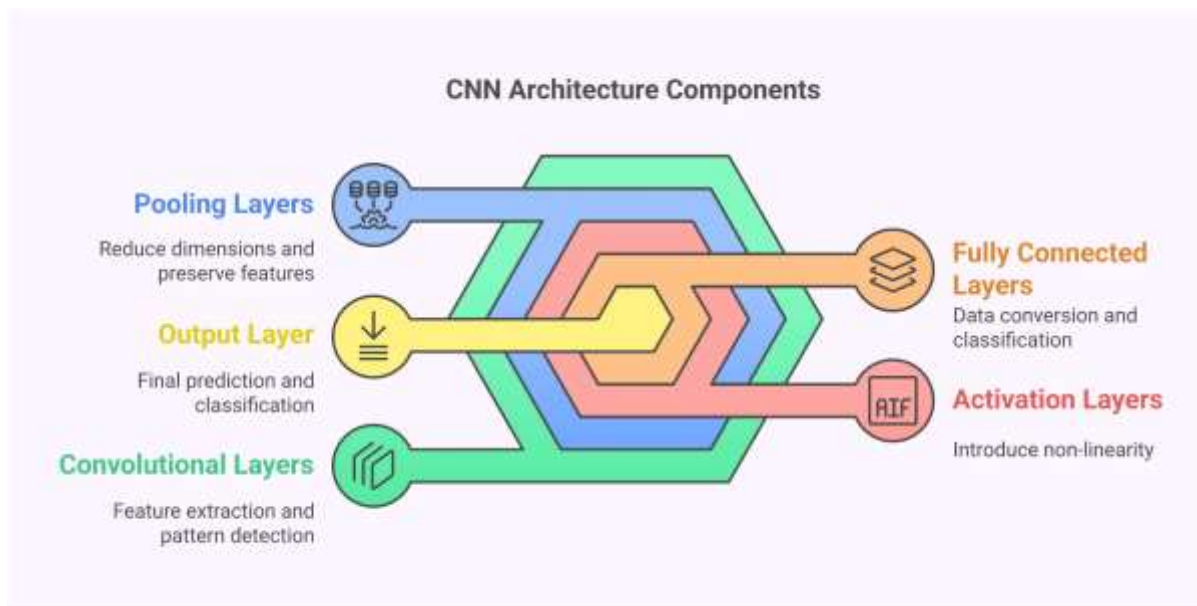


Рис. 1.3 Базова структура згорткової нейронної мережі (CNN)

3D CNN та спатіо-темпоральне моделювання

Для безпосереднього аналізу руху широко застосовуються тривимірні згортки (3D CNN), де згортка виконується в трьох вимірах: ширина \times висота \times час. Такі мережі здатні одночасно аналізувати зміни в просторі та часі, що робить їх ефективними у задачах:

- класифікації відео,
- розпізнавання дій (action recognition),
- аналізу поведінки об'єктів,
- виявлення аномалій у динамічних сценах.

Однією з перших популярних моделей стала C3D, а пізніше Tran et al. запропонували гібридні 3D-архітектури для кращого вилучення спатіо-темпоральних ознак [14].

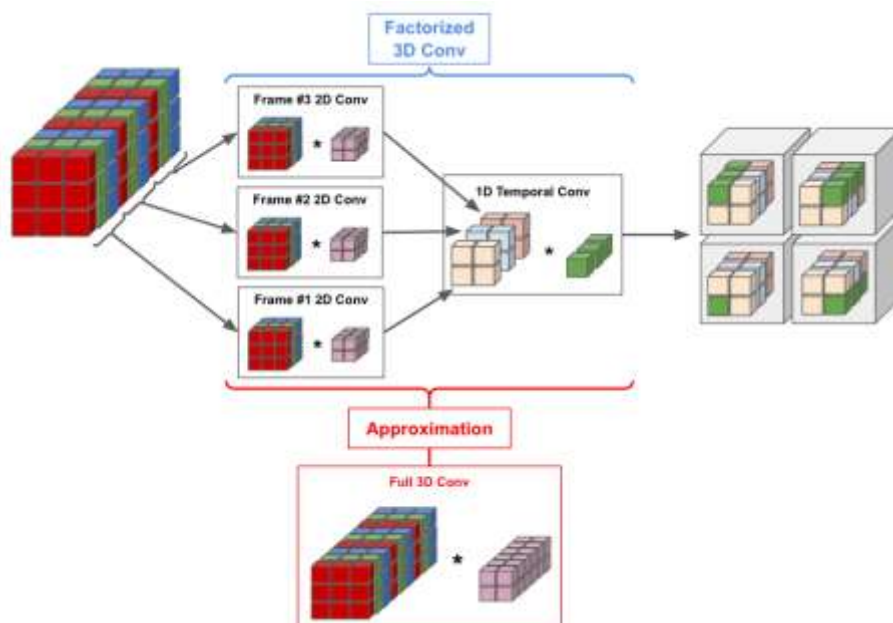


Рис. 1.4 Приклад 3D-згортки та формування спатіо-темпоральних ознак

RNN, LSTM і ConvLSTM для моделювання часових залежностей

Рекурентні нейронні мережі (RNN) довгий час були основним інструментом для роботи з послідовностями. Проте звичайні RNN не здатні ефективно запам'ятовувати довгі залежності, тому були запропоновані LSTM та GRU, які зменшують проблему затухання градієнтів [1].

У відеоаналітиці застосовуються:

- **RNN** - для моделювання коротких залежностей між кадрами;
- **LSTM/GRU** - для довготривалої динаміки;
- **ConvLSTM** - для задач, де потрібно аналізувати просторові та часові зміни одночасно, зокрема для виявлення аномалій у відео [12].

ConvLSTM моделює часову послідовність не у вигляді векторів, а як послідовність ознак, розміщених на сітці (feature maps). Це робить їх придатними для роботи з відео та динамічними сценами.

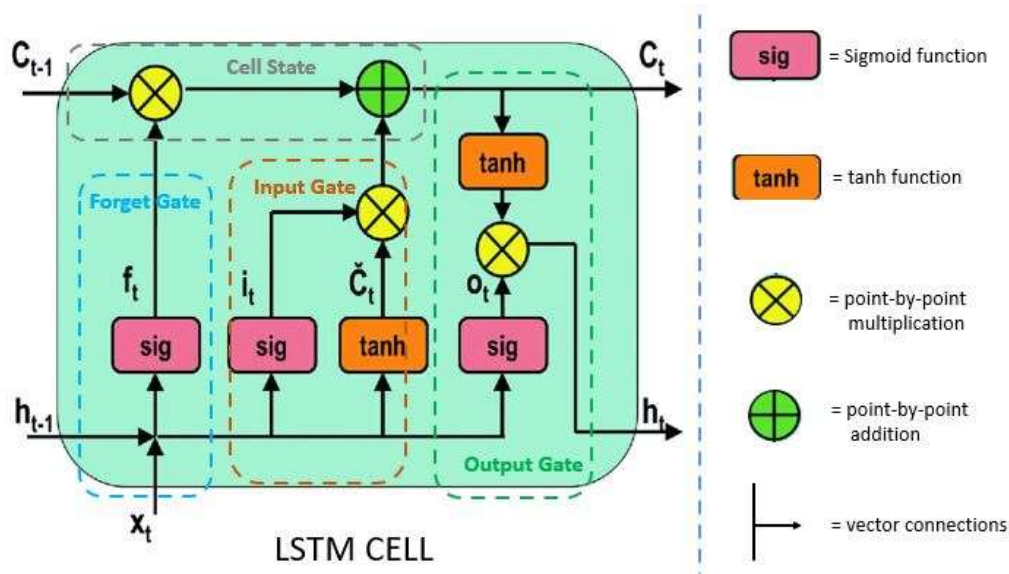


Рис. 1.5 Архітектура LSTM-блоку та його комірок пам'яті

Transformer-архітектури для відео

Останніми роками трансформери стали провідною технологією в задачах комп'ютерного зору. Спочатку вони застосовувалися в NLP, але після робіт Vision Transformer (ViT) та DeiT їх почали активно використовувати й у відеоаналізі.

У відео застосовуються модифікації:

- **ViT-based Video Transformers** - обробляють відео як послідовність патчів;
- **TimeSformer** - окремо моделює просторові та часові залежності;
- **Video Swin Transformer** - працює в ковзних вікнах і підходить для задач реального часу.

Переваги:

- висока точність,
- ефективна робота з довгими послідовностями,
- можливість мультимодального аналізу (відео + звук + текст).

Недоліки:

- значні обчислювальні вимоги,
- потреба у великих наборах даних для навчання.

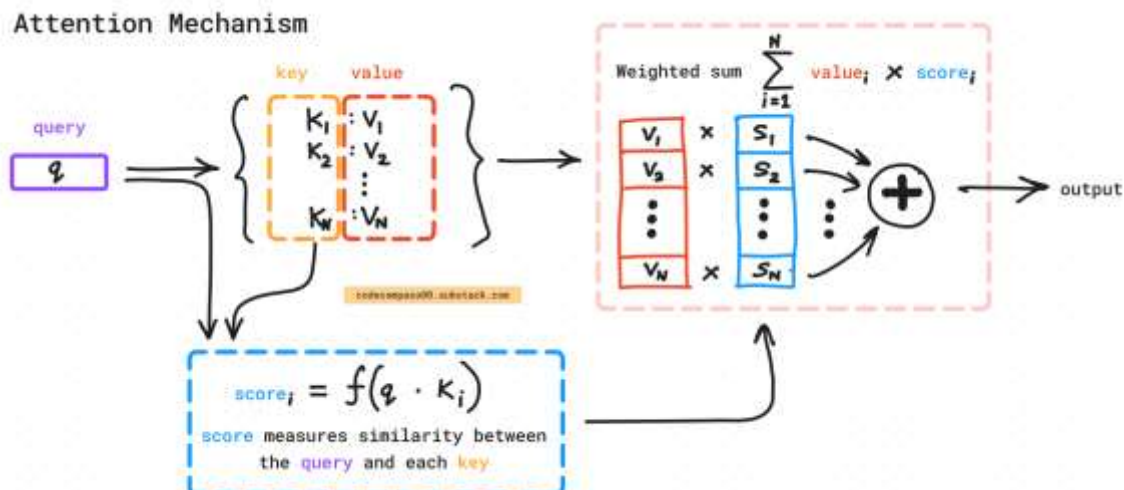


Рис. 1.6 Загальна схема архітектури Transformer (Self-Attention Mechanism)

Порівняння архітектур для відеоаналізу

У практичних системах застосовують різні архітектури залежно від задачі, доступних ресурсів та вимог до швидкодії.

Методи глибокого навчання охоплюють просторовий і часовий аналіз відео, дозволяючи створювати потужні системи для автоматизованого виявлення об'єктів, аналізу руху та розпізнавання складних подій у відеопотоках. Кожен тип архітектури має свою область застосування, а вибір моделі залежить від вимог до точності, ресурсів і швидкодії.

Таблиця 1.3

Порівняння основних моделей відеоаналізу

Модель	Просторовий аналіз	Часовий аналіз	Швидкодія	Підходить для real-time
CNN	Так	Ні	Висока	Частково
3D CNN	Так	Так	Середня	Залежить від оптимізації
LSTM/GRU	Ні	Так	Середня	Так
ConvLSTM	Так	Так	Низька–середня	Для edge пристроїв – ні
Transformers	Так	Так	Низька	Переважно ні без GPU

1.4. Формати та протоколи передавання відео у телекомунікаційних мережах (RTSP, RTP, HLS, WebRTC)

Передавання відеопотоків у системах електронних комунікацій потребує використання спеціалізованих протоколів, здатних забезпечувати низьку затримку, високу якість відтворення та адаптацію до змінних параметрів мережі. Вибір протоколу безпосередньо впливає на ефективність роботи алгоритмів автоматизованого відеоаналізу, оскільки затримки, пропуски кадрів, джиттер та зміна бітрейту впливають на можливість точного виявлення подій у відеопотоці [15], [16].

Сучасні системи передавання відео базуються на двох основних підходах:

1. Протоколи реального часу (Real-Time Protocols): RTP, RTSP, WebRTC.
2. Адаптивні протоколи потокового мовлення (Adaptive Streaming): HLS, DASH.

Кожен з них має власні механізми синхронізації, управління трафіком, корекції помилок та масштабування.

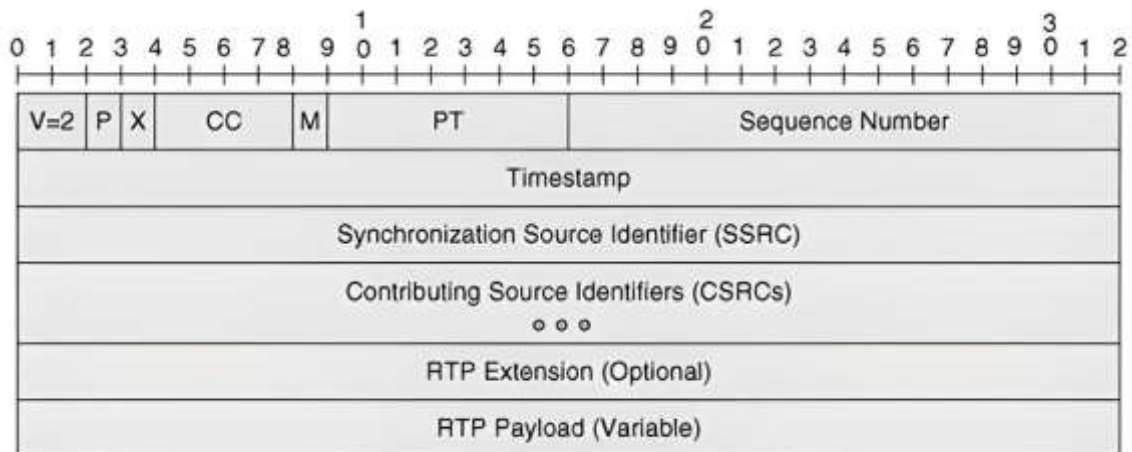


Рис. 1.7 Структура RTP-пакета

У системах відеоспостереження RTP часто використовується разом із RTSP для забезпечення керування потоком. Формати кодування можуть включати H.264, H.265, VP8, MJPEG. Дані у таких потоки відтворюються майже миттєво, що дозволяє застосовувати їх у задачах:

- відстеження об'єктів,
- виявлення руху,
- аналізу подій в реальному часі.

RTSP – Real-Time Streaming Protocol

RTSP виконує функцію “контролера потоку”, тобто забезпечує операції:

- PLAY
- PAUSE
- RECORD
- TEARDOWN

- SETUP

Сам RTSP не передає відео, а лише керує RTP-потокком.

RTSP широко використовується у:

- IP-камерах,
- системах відеоспостереження,
- промислових рішеннях,
- наукових відеокomплексах.

Структура протоколу передбачає наявність каналів керування та даних, що описано в IEEE та відповідних RFC [18].

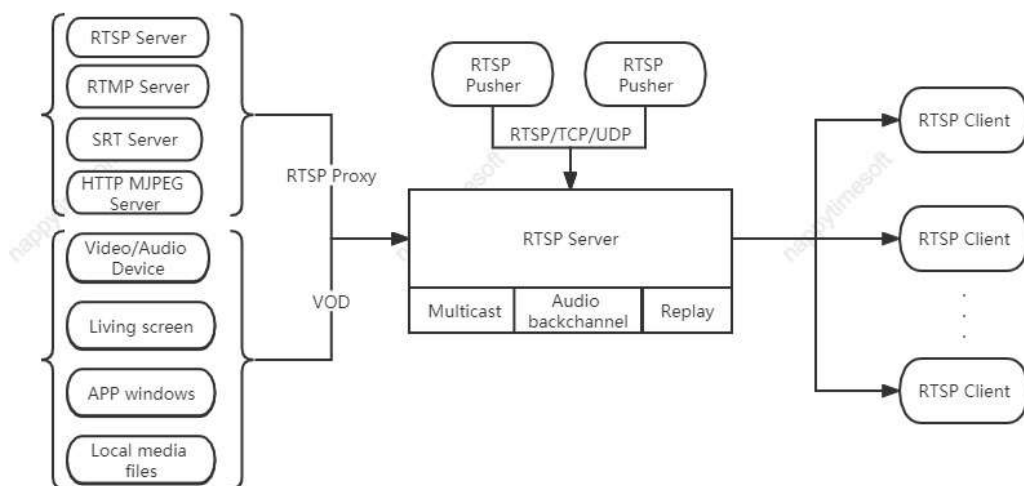


Рис. 1.8 Архітектура RTSP-потокку

Переваги RTSP:

- низька затримка;
- можливість керування потокком;
- підтримка реального часу.

Недоліки:

- погана робота через HTTP-проксі;
- складніша конфігурація;
- більша чутливість до втрат пакетів.

HLS – HTTP Live Streaming

HLS (розроблений Apple) є протоколом **адаптивного** потокового мовлення.

На відміну від RTP/RTSP, HLS:

- базується на HTTP,
- легко проходить через NAT і проксі,
- ділить відео на невеликі сегменти (2–6 секунд),
- автоматично підлаштовує якість під пропускну здатність мережі.

Принцип роботи:

1. Відео розбивається на .ts або .mp4 сегменти.
2. Створюється плейлист .m3u8.
3. Клієнт завантажує сегменти по черзі.
4. При падінні пропускну здатності обирається сегмент меншої якості.

HLS ідеально підходить для:

- відеострімінгу,
- онлайн-трансляцій,
- відеоконтенту з високими вимогами до адаптивності.

Однак затримка є значно вищою - від 5 до 30 секунд, що робить HLS непридатним для більшості систем відеоаналітики реального часу.

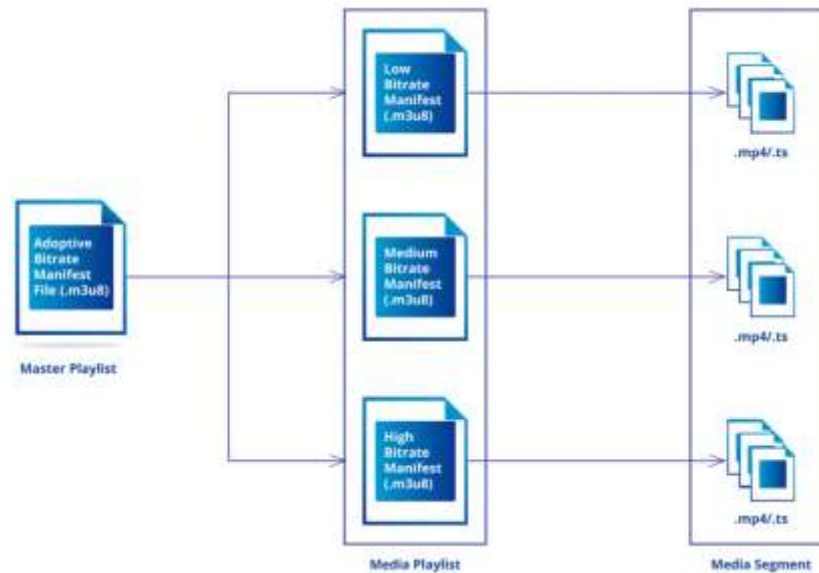


Рис. 1.9 Принцип роботи HLS (segment-based streaming)

DASH – MPEG Dynamic Adaptive Streaming over HTTP

DASH є відкритим стандартом адаптивного стримінгу, схожим на HLS, але підтримує:

- гнучкіший формат проявлення (manifest),
- різні кодеки,
- мультиплексування кількох відеопотоків.

Принципи роботи DASH аналогічні до HLS, за винятком того, що DASH використовує формат .mpd замість .m3u8.

Переваги DASH:

- більш широке застосування поза екосистемою Apple;
- підтримка підвищених роздільностей (4K, 8K, HDR);
- краща масштабованість.

Недоліки:

- також має природну затримку через сегментацію відео.

Внаслідок високої затримки DASH рідко використовується у відеоаналітиці реального часу.

WebRTC – технологія реального часу для браузерів і мобільних пристроїв

WebRTC - це сучасний стандарт для передачі відео з ультранизькою затримкою. Він використовується в:

- браузерах (Chrome, Firefox, Safari),
- мобільних застосунках,
- IoT-пристроях,
- віддалених AI/ML системах.

Особливість WebRTC полягає у:

- прямому з'єднанні Peer-to-Peer,
- використанні SRTP (захищеного RTP),
- адаптації бітрейту у реальному часі,
- автоматичному обході NAT через STUN/TURN-сервери.

Принцип роботи WebRTC включає такі елементи:

- **ICE** - Interactive Connectivity Establishment
- **DTLS** - шифрування
- **SRTP** - передавання медіа
- **SCTP** - data channels

WebRTC - один з найкращих кандидатів для систем відеоаналітики, що працюють через браузер або мобільні клієнти, оскільки забезпечує затримку < 500 мс [19].

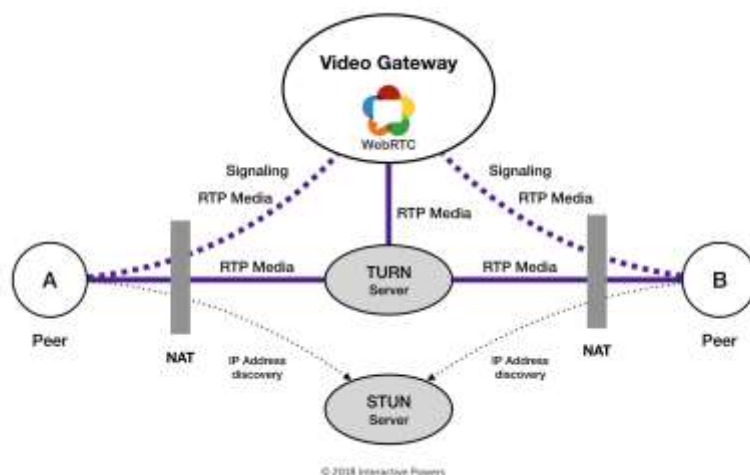


Рис. 1.10 Загальна архітектура WebRTC (STUN/TURN, Peer Connection)

Таблиця 1.4

Порівняння основних протоколів передачі відео

Протокол	Тип	Затримка	Адаптивність	Основні сфери застосування
RTP	Реальний час	дуже низька	ні	ІР-камери, відеонагляд
RTSP	Контроль + RTP	низька	ні	системи безпеки, сервіси моніторингу
WebRTC	P2P real-time	дуже низька	так	браузерні та мобільні відеосистеми
HLS	HTTP adaptive	висока	так	стрімінг, трансляції
DASH	HTTP adaptive	висока	так	медіасервіси, CDN

Висновки

У першому розділі було здійснено комплексний аналіз теоретичних основ автоматизованого аналізу відеопотоків у системах електронних комунікацій. Проведений огляд літератури дозволив сформулювати цілісне

уявлення про структуру відеопотоків, особливості їх функціонування у мережах зв'язку та обмеження, що впливають на точність і стабільність подальшої відеообробки. Встановлено, що через властиву телекомунікаційним мережам варіативність пропускну здатності, затримок, втрат пакетів та мережевих спотворень системи відеоаналітики потребують адаптивних алгоритмів обробки даних, здатних працювати в умовах неповних або деградованих даних.

Аналіз методів комп'ютерного зору показав, що класичні підходи до виділення ознак забезпечують високу швидкість, але недостатньо ефективні у складних динамічних сценах. На противагу цьому, згорткові нейронні мережі та їхні модифікації демонструють здатність автоматично вилучати просторові й часові закономірності, що робить їх придатними для задач детекції, трекінгу та виявлення аномалій у відеопотоці. Розглянуті архітектури - CNN, 3D CNN, LSTM, ConvLSTM, Transformer - відрізняються обчислювальними вимогами, точністю та гнучкістю, що зумовлює необхідність зваженого вибору при побудові практичних систем.

Значну увагу приділено також протоколам і форматам передавання відео в телекомунікаційних мережах. Аналіз RTP, RTSP, HLS, DASH та WebRTC дозволив встановити їх придатність до використання в задачах реального часу. Протоколи RTP/RTSP та WebRTC забезпечують мінімальні затримки та тому є найбільш відповідними для систем відеоаналітики, тоді як HLS і DASH орієнтовані на адаптивне потокове мовлення з високими затримками, що обмежує їхнє застосування для алгоритмів реального часу. Це підтверджує ключову важливість вибору формату передавання відео для забезпечення коректної роботи моделей комп'ютерного зору.

Узагальнення результатів огляду дозволило визначити як теоретичні, так і практичні прогалини, що залишаються нерозв'язаними в сучасних

системах відеоаналітики: обмежена стійкість алгоритмів до мережевих спотворень, складність обробки потоків з різною якістю кадрів, необхідність балансування між точністю моделей та їх продуктивністю на реальному обладнанні. Це обґрунтовує доцільність подальших досліджень та розроблення архітектури алгоритмічного конвеєра, що буде представлено у наступному розділі.

Отже, теоретичний аналіз, проведений у першому розділі, створює необхідне наукове підґрунтя для обґрунтованого вибору методів, моделей і протоколів, які будуть використані для побудови практичної системи автоматизованого аналізу відеопотоків у системах електронних комунікацій.

РОЗДІЛ 2

МЕТОДОЛОГІЯ АВТОМАТИЗОВАНОГО АНАЛІЗУ ВІДЕОПОТОКІВ У РЕАЛЬНИХ ЕЛЕКТРОННИХ КОМУНІКАЦІЙНИХ СИСТЕМАХ

У другому розділі увагу зосереджено на методології розроблення та обґрунтуванні вибору алгоритмів, моделей і технічних рішень, які забезпечують ефективний автоматизований аналіз відеопотоків у системах електронних комунікацій. На відміну від першого розділу, що висвітлював теоретичні засади та узагальнені підходи, даний розділ спрямований на формування практично орієнтованого підходу до побудови системи відеоаналітики, здатної працювати в умовах реального мережевого трафіку.

Оскільки відеопотоки, що передаються телекомунікаційними мережами, характеризуються варіаціями у частоті кадрів, пропускній здатності, рівні шумів та можливими втратами пакетів, необхідно обрати такі алгоритми й архітектури, які забезпечують стійкість до цих факторів і можуть функціонувати в режимах обмежених ресурсів. У цьому контексті методи детекції та трекінгу об'єктів, технології аналізу аномалій, а також сучасні архітектури edge- і cloud-обчислень виступають фундаментальними елементами майбутньої системи.

Другий розділ охоплює порівняння сучасних алгоритмів детекції об'єктів, методів трекінгу, моделей часово-просторового аналізу, протоколів роботи мережевих потоків, а також інфраструктурних рішень, необхідних для обробки відео у реальному часі.

2.1. Методи детекції об'єктів у відеопотоці (YOLO, SSD, EfficientDet)

Детекція об'єктів є одним із ключових етапів автоматизованого аналізу відеопотоків, оскільки визначає, які об'єкти присутні на сцені та де саме вони розташовані на кожному кадрі. На відміну від класичної класифікації зображень, задача детекції вимагає одночасного визначення класу об'єкта та координат його області (bounding box). Використання ефективних алгоритмів детекції відіграє критичну роль у побудові систем відеоаналітики реального часу, таких як системи безпеки, моніторинг дорожньої інфраструктури, промисловий контроль та автоматичний аналіз поведінкових патернів у потоках відеоспостереження.

Сучасні методи детекції об'єктів ґрунтуються на глибинних згорткових нейронних мережах, які автоматично виділяють ознаки об'єктів на різних рівнях абстракції. Ранні підходи, такі як R-CNN, Fast R-CNN та Faster R-CNN, забезпечували високу точність, проте мали значні обчислювальні витрати через етапи регіонального пошуку (region proposals). Це суттєво ускладнювало їх застосування в обмежених умовах відеопотоків реального часу. Потреба в більш швидких і менш ресурсомістких моделях зумовила появу одностадійних детекторів - SSD, YOLO та EfficientDet - які забезпечують обробку зображень безпосередньо в один прохід мережі, що є ключовою перевагою для систем реального часу [1], [5], [7].

Одностадійні детектори та їх відмінність від двостадійних методів

Одностадійні детектори (one-stage detectors) виконують класифікацію та регресію координат одночасно, без поділу на етап попереднього пошуку областей. Це забезпечує значне підвищення швидкодії. Двостадійні методи (наприклад, Faster R-CNN) демонструють високу точність, але значно

поступаються у швидкості через складну архітектуру та більшу кількість операцій.

Основні переваги одностадійних детекторів:

- здатність працювати у реальному часі навіть на середніх GPU;
- менша складність моделі;
- висока масштабованість;
- компактніші архітектури (важливо для embedded-пристроїв);
- зручність для аналізу відео потокових даних без затримок.

Для відеоаналітики у телекомунікаційних мережах саме одностадійні моделі є найбільш оптимальними, оскільки дають змогу компенсувати варіативність мережевих параметрів низькою затримкою обробки відео.

YOLO (You Only Look Once)

YOLO є одним із найвідоміших одностадійних детекторів, що забезпечує унікальний баланс між швидкістю та точністю. Головна ідея полягає в обробці зображення цілком одним проходом, під час якого мережа прогнозує координати та класи об'єктів для декількох комірок сітки [7]. Це суттєво відрізняє YOLO від методів, що працюють на рівні окремих регіонів.

Основні характеристики YOLO:

- **Висока швидкість:** починаючи з YOLOv2 та YOLOv3, обробка може перевищувати 45–100 FPS на сучасних GPU, що робить його придатним для відео в реальному часі.
- **Єдина CNN:** одна модель одночасно виконує класифікацію та регресію bounding boxes.

- **Простота використання:** у порівнянні з двостадійними методами YOLO значно легше розгортати в реальних системах.
- **Гнучкість:** існують різні модифікації YOLO (Tiny, Nano, S, XL), адаптовані до різних обчислювальних платформ.

YOLOv3 [7] особливо важливий для практичних систем, оскільки вміє обробляти об'єкти різного масштабу завдяки багаторівневій структурі з використанням Feature Pyramid Networks.

Переваги YOLO у контексті телекомунікаційних систем:

- здатність працювати з потоками низької якості (RTSP низьких бітрейтів);
- стійкість до часткової втрати кадрів;
- можливість обробки відеопотоків у реальному часі на edge-пристроях.

SSD (Single Shot Multibox Detector)

SSD є одним із перших ефективних одностадійних алгоритмів, який поєднує швидкість і точність шляхом застосування різних карт ознак (feature maps) для обробки об'єктів різних розмірів. SSD виконує прогноз координат та класів безпосередньо з особливостей, отриманих CNN, без проміжного етапу генерації регіонів.

Переваги SSD:

- **Швидкість:** SSD демонструє продуктивність на рівні 40–65 FPS на GPU середнього класу.
- **Архітектурна простота:** модель складається з базової CNN (часто VGG16 або ResNet) з додатковими шарами.

- **Підтримка багаторозмірних об'єктів:** завдяки використанню багаторівневих feature maps.

SSD менше чутливий до різких змін освітлення та може працювати з відео середньої якості. Проте, порівняно з YOLO, SSD має нижчу точність у складних сценах з великою кількістю дрібних об'єктів. SSD є оптимальним вибором для систем, де потрібна хороша рівновага між точністю та швидкістю на пристроях з обмеженими ресурсами - наприклад, у камерах відеоспостереження, промислових вбудованих пристроях та edge-серверах.

EfficientDet

EfficientDet - це сучасна архітектура для об'єктної детекції, що поєднує масштабованість, точність та енергоефективність. Модель базується на двох ключових ідеях:

1. **EfficientNet backbone** - високооптимізована мережа для виділення ознак.
2. **BiFPN (Bidirectional Feature Pyramid Network)** - багатонаправлена піраміда ознак, яка забезпечує ефективну передачу інформації між шарами [8].

EfficientDet пропонує цілу лінійку моделей від D0 до D7, де кожен індекс збільшує розмірність моделі, глибину, ширину та роздільність входу. Це дає змогу обирати архітектуру залежно від доступних обчислювальних ресурсів.

Переваги EfficientDet:

- **Висока точність при низьких витратах ресурсів:** навіть EfficientDet-D0 перевершує багато популярних моделей.

- **Гнучкість:** сімейство моделей дозволяє адаптувати систему під edge-, mobile- або cloud-сценарії.
- **Стабільність у складних сценах:** BiFPN ефективно обробляє об'єкти різного масштабу.

У контексті відеоаналізу EfficientDet є одним з найкращих варіантів для застосувань, де важлива точність і допускається помірно збільшення затримки (наприклад, у хмарних сервісах або edge-платформах з GPU).

Порівняльний аналіз моделей для задач відеоаналізу

Для систем реального часу ключовими параметрами є:

- **FPS (кількість кадрів за секунду),**
- **затримка (inference latency),**
- **якість виявлення об'єктів (mAP),**
- **ресурсоспоживання,**
- **стійкість до шумів та неякісних кадрів.**

YOLO забезпечує найкращий баланс між швидкістю та точністю, SSD пропонує високу швидкість із помірною точністю, тоді як EfficientDet демонструє найвищу точність за рахунок більшої складності моделі.

Порівняльні висновки:

- **Для реального часу у відеоаналізі:** оптимально застосовувати YOLO (особливо YOLOv3, YOLOv4 або компактні моделі YOLOv5/YOLOv7).
- **Для вбудованих систем:** SSD або YOLO Tiny.
- **Для cloud/edge-сервісів із GPU:** EfficientDet або старші YOLO-моделі.
- **Для складних сцен:** EfficientDet завдяки BiFPN.

Виклики та обмеження при застосуванні детекторів у відеопотоках

Хоча сучасні моделі демонструють високу точність, існують проблеми, характерні саме для потокових даних:

- **втрата кадрів у RTP/RTSP,**
- **нестабільний FPS,**
- **дереворізні (non-stationary) сцени,**
- **рух камери,**
- **високий рівень шуму в нічних чи інфрачервоних відео,**
- **неоднорідні умови освітлення.**

Моделі YOLO та SSD частково компенсують ці фактори завдяки швидкодії та інваріантності до масштабу. EfficientDet краще працює при складних умовах, але потребує більшої стабільності потоку і ресурсів.

2.2. Методи трекінгу об'єктів (SORT, DeepSORT, ByteTrack)

Трекінг об'єктів у відеопотоці є одним з ключових етапів відеоаналітики, оскільки дозволяє не тільки виявляти об'єкти на кожному окремому кадрі, але й відстежувати їхню траєкторію в часі. Це необхідно для розв'язання широкого спектра практичних задач: підрахунок об'єктів, аналіз поведінки, виявлення аномалій, моніторинг транспортних потоків, побудова теплових карт руху, а також для систем безпеки та спостереження. На відміну від детекції, яка відповідає лише за ідентифікацію об'єктів у кадрі, трекінг забезпечує **зв'язність об'єктів між кадрами**, що є критично важливим для інтелектуальних систем відеоаналізу.

Сучасні методи відстеження об'єктів ґрунтуються на двох основних підходах:

1. **Tracking-by-Detection** - використовується детектор об'єктів (YOLO, SSD, EfficientDet), а далі матчується інформація між кадрами.
2. **Visual Tracking** - трекер працює без детектора (наприклад, KCF, MOSSE, CSRT), але цей підхід майже не застосовується в сучасних системах відеоаналітики.

Для потокового відео в телекомунікаційних мережах найбільш ефективним є саме **підхід tracking-by-detection**, оскільки він дозволяє компенсувати втрати кадрів, варіації освітлення та низьку якість відео, що характерно для RTSP-потоків [9].

Було розглянуто три найважливіші сучасні алгоритми трекінгу - SORT, DeepSORT та ByteTrack - які широко застосовуються у промислових, транспортних та безпекових системах відеоаналітики.

SORT (Simple Online and Realtime Tracking)

SORT є одним із найпопулярніших алгоритмів відстеження завдяки своїй високій швидкодії та простоті реалізації. Алгоритм представлений у роботі Bewley et al. [9] і застосовує два ключові компоненти:

- **Kalman Filter** для прогнозування положення об'єктів у наступному кадрі;
- **Hungarian Algorithm** для асоціації прогнозів з детекціями.

Принцип дії SORT складається з таких етапів:

1. **Детекція об'єктів** на поточному кадрі.
2. **Прогноз траєкторії** для кожного трека за допомогою фільтра Калмана.
3. **Відповідність детекцій прогнозам** за метрикою IoU (Intersection over Union).

4. **Оновлення треків** або створення нових при появі об'єктів.
5. **Видалення “зниклих” треків**, якщо відсутні відповідні детекції протягом певної кількості кадрів.

Переваги SORT:

- надзвичайно висока швидкість (до 260 FPS на середньому GPU);
- простота реалізації та невеликі обчислювальні витрати;
- відсутність необхідності у додаткових нейромережевих моделях.

Недоліки SORT:

- нестійкість при часткових перекриттях об'єктів;
- погана робота в умовах скупчень;
- відсутність ідентифікації (appearance features).

Через це SORT добре працює у сценах з невеликою кількістю об'єктів або мінімальними перекриттями, але недостатньо ефективний у складних сценаріях відеонагляду.

DeepSORT

DeepSORT був запропонований як розширення SORT, щоб вирішити основну проблему останнього - втрату об'єктів під час перекриття. Модифікована версія використовує **appearance features** (візуальні ознаки) об'єкта, які витягуються за допомогою компактної нейронної мережі [10].

DeepSORT включає:

- Kalman filter (як у SORT);
- Hungarian Algorithm для асоціації;
- CNN для виділення appearance embedding;
- комбіновану метрику відповідності (IoU + косинусна схожість).

Переваги DeepSORT:

- надзвичайно стійке відстеження;
- здатність зберігати ідентичність об'єкта навіть при тривалих перекриттях;
- придатність для складних багатолюдних сцен.

Недоліки:

- більші обчислювальні витрати;
- необхідність GPU для роботи в реальному часі на високих роздільностях;
- залежність від якості appearance embedding.

DeepSORT є “золотим стандартом” у задачах відеотрекінгу для систем відеонагляду, безпеки, моніторингу транспорту та розумних міст, де критичною є стабільність і точність треків.

ByteTrack

ByteTrack - один із найбільш сучасних алгоритмів трекінгу, який демонструє рекордні результати у задачах MOT (Multi-Object Tracking). На відміну від DeepSORT, ByteTrack не просто враховує появу та зникнення об'єктів, а **використовує як висококонфідентні, так і низькоконфідентні детекції** [11].

Ідея полягає в тому, що багато детекторів (наприклад, YOLO) інколи видають детекції з низькою довірою, але ці детекції все ще можна коректно зв'язати із треками. ByteTrack застосовує подвійний етап відповідності:

1. **Спочатку відповідність висококонфідентних детекцій.**
2. Далі - **відповідність низькоконфідентних детекцій**, які залишилися після першого етапу.

Такий підхід значно підвищує стійкість трекінгу, особливо в умовах:

- різкої зміни освітлення,
- низького бітрейту відеопотоків,
- часткових втрат кадрів,
- рухомої камери.

Переваги ByteTrack:

- одна з найвищих точностей у benchmark'ах MOT17/MOT20;
- стійкість до шумів і низькоякісних детекцій;
- відсутність CNN для appearance embedding → висока швидкість.

Недоліки:

- хоч і швидший за DeepSORT, але дещо повільніший за SORT;
- для максимальної точності бажано використовувати детектори з високою чутливістю.

ByteTrack є оптимальним для складних сцен, де велика кількість об'єктів рухається в різних напрямках, - наприклад, перехрестя, аеропорти, торгові центри, стадіони, автостради.

Порівняльний аналіз SORT, DeepSORT і ByteTrack

Методи трекінгу відрізняються своїм фокусом: SORT оптимізований на швидкість, DeepSORT - на точність і стабільність, ByteTrack - на баланс між обома аспектами.

Основні відмінності:

- SORT - найшвидший, проте неточний у випадках перекриття.
- DeepSORT - найбільш точний завдяки appearance features, але потребує більше ресурсів.

- ByteTrack - найкращий компроміс точності та швидкості.

У реальних системах відеоаналітики:

- **SORT** використовують для простих сценаріїв (склади, контроль ліній).
- **DeepSORT** - для критичних систем безпеки (відеонагляд, контроль доступу).
- **ByteTrack** - для великих складних потоків (міські системи, транспорт, натовпи).

Виклики трекінгу у телекомунікаційних мережах

У контексті мережевих потоків (RTP, RTSP, WebRTC) трекінг стикається з додатковими труднощами:

- нерівномірна частота кадрів через нестабільний бітрейт;
- вилучення кадрів та мережеві затримки;
- шуми, артефакти та змінні умови освітлення;
- компресія відео (H.264/H.265), що створює “змазування” кадрів;
- швидкі рухи або обертання камери.

Саме тому поєднання:

- **YOLO** → **ByteTrack**
або
- **EfficientDet** → **DeepSORT**

є найпоширенішими практичними конфігураціями у сучасних інтелектуальних відеосистемах.

2.3. Виявлення аномалій у відеопотоці (Autoencoder, ConvLSTM, 3D-CNN підходи)

Виявлення аномалій у відеопотоці є однією з найбільш важливих і водночас складних задач у системах автоматизованого відеоаналізу. На

відміну від детекції та трекінгу, де існує чітко визначений набір класів об'єктів, аномалія визначається як **будь-яка подія, що відхиляється від нормального патерну поведінки**, який властивий даній сцені. У багатьох випадках неможливо наперед визначити повний перелік аномальних ситуацій, а їхня поява є рідкісною, нерегулярною та непередбачуваною [12]. Це робить задачу аномалій складною, оскільки:

- набори даних є **незбалансованими** (аномалії зустрічаються рідко),
- у сценах існує **велика варіативність нормальної поведінки**,
- аномалії можуть бути візуально неочевидними,
- у потоках RTSP виникають мережеві спотворення, що схожі на аномалії.

Отже, традиційні методи класифікації непридатні, і застосовуються спеціалізовані підходи, засновані на **моделюванні нормальної поведінки** та подальшому пошуку відхилень.

Методи на основі автокодерів (Autoencoder-Based Methods)

Автокодери є одним з найпоширеніших підходів до виявлення аномалій, оскільки вони навчаються відтворювати нормальні патерни поведінки. Ідея полягає в тому, що автоенкодер навчається реконструювати відеокадри або їх представлення, а потім **аномалії виявляються як кадри з високою похибкою реконструкції** [12].

У контексті відео широко застосовуються:

- **Convolutional Autoencoders (CAE)** - працюють з кадрами як зображеннями,
- **Spatiotemporal Autoencoders** - враховують як простір, так і час,
- **Predictive Autoencoders** - прогнозують наступний кадр.

Один із класичних підходів представлено в роботі Hasan et al. [13], де модель вчиться передбачати наступний кадр, а аномалії мають вищу похибку прогнозу.

Переваги автоенкодерів:

- не потребують великої кількості аномальних даних;
- добре моделюють нормальну поведінку;
- ефективні для статичних і напівдинамічних сцен.

Недоліки:

- можуть “притиратися” до аномалій, якщо вони часто повторюються;
- погано працюють у сценах з високою динамікою;
- чутливі

Таблиця 2.1

Основні типи автоенкодерів для відеоаналізу

Тип моделі	Вхідні дані	Переваги	Недоліки
CAE (Convolutional Autoencoder)	Окремі кадри	Простий, швидкий	Не враховує часову динаміку
Spatiotemporal AE	Послідовності кадрів	Аналізує рух	Вищі ресурси
Predictive AE	Кадри $t \rightarrow t+1$	Добре для динаміки	Складні в навчанні
Variational AE	Латентні простори	Генеративні можливості	Менше підходять для точного прогнозу

ConvLSTM для виявлення аномалій

ConvLSTM представляє собою модифікацію звичайної LSTM, де всі операції виконуються згортками (convolutions), що дозволяє моделі одночасно вловлювати просторові та часові залежності. У роботі Medel та Savakis [12] показано, що ConvLSTM може точно моделювати рух у відео та виявляти аномалії через похибку реконструкції або прогнозування.

ConvLSTM працює за принципом:

1. Модель отримує послідовність кадрів розміру N .

2. Вона кодує їх у латентне представлення.
3. Прогнозує наступний кадр або реконструює вхідні.
4. Похибка використовується для визначення аномалії.

Переваги ConvLSTM:

- враховує як простір, так і час;
- ефективний для однотипних сцен (камера не рухається);
- добре працює на низькорівневих патернах руху (повільні зміни, повторювані процеси).

Недоліки:

- високі обчислювальні витрати;
- погана стійкість до швидких рухів;
- працює гірше на складних динамічних сценах.

Таблиця 2.2

Порівняння моделей Autoencoder vs ConvLSTM

Критерій	Autoencoder	ConvLSTM
Просторовий аналіз	Так	Так
Часовий аналіз	Ні / частково	Так
Складність навчання	Низька	Висока
Ефективність на динамічних сценах	Середня	Висока
Робота в реальному часі	Може бути	Складно
Стійкість до шумів	Середня	Висока

3D-CNN та спatio-темпоральні моделі

3D-CNN, представлені в роботі Tran et al. [14], є одним із найбільш природних підходів до аналізу відео, оскільки тривимірні згортки

дозволяють виявляти просторові та часові патерни одночасно. Це робить їх придатними для аналізу поведінки, виявлення неочевидних аномалій та складних динамічних сцен.

3D-CNN навчаються на послідовностях кадрів розміром $N \times H \times W$ і вловлюють рух у послідовності фрагментів (video clips). Вони широко використовуються в задачах:

- виявлення аномальних дій (running, fighting, falling),
- визначення незвичної поведінки в транспортних потоках,
- аналізу промислових процесів,
- виявлення небезпечних ситуацій.

Переваги:

- сильний спatio-темпоральний аналіз;
- можливість розпізнавати складні патерни;
- хороша масштабованість.

Недоліки:

- дорогі обчислювально;
- великі вимоги до пам'яті;
- потребують потужного GPU або навіть TPU.

Таблиця 2.3

Порівняння ConvLSTM та 3D-CNN у задачах аномалій

Критерій	ConvLSTM	3D-CNN
Тип моделювання	Час + простір	Час + простір
Підходить для	Локальних рухів	Глобальних складних рухів
Обчислювальні витрати	Середні	Високі
Робота з довгими послідовностями	Так	Обмежено
Робота в реальному часі	Складно	Складно
Здатність до виявлення складних аномалій	Середня	Висока

Проблеми застосування моделей виявлення аномалій у реальних відеопотоках

У мережевих відеопотоках (RTSP, RTP, WebRTC) існують специфічні проблеми:

- **мережеві артефакти** (змазування, макроблоки, фрізи), які схожі на аномалії;
- **плаваючий FPS**, який мішає часовому аналізу;
- **камера може рухатись**, що порушує моделі, навчені на статичних сценах;
- **різка зміна освітлення** може викликати сплеск похибки reconstruction error.

Тому в реальних системах часто застосовують:

- 3D-CNN для складних сценаріїв,
- ConvLSTM для середньої динаміки,

- Autoencoder - для простих або статичних сцен.

2.4. Моделі оцінки якості відео та QoS/QoE у мережах зв'язку

Якість відеопотоку у телекомунікаційних мережах є критичним фактором для стабільної роботи систем відеоаналітики. На відміну від потокового мовлення (стрімінгу), де пріоритетом є комфорт користувача, у системах автоматизованого аналізу відео важливо забезпечити **стабільність кадрів, відсутність втрат, передбачувану затримку та коректність структури відео**, оскільки будь-які мережеві спотворення можуть спричинити зниження точності алгоритмів детекції, трекінгу та виявлення аномалій [15].

Для оцінки якості відео використовуються два ключові підходи:

1. **QoS (Quality of Service)** - технічні параметри мережі.
2. **QoE (Quality of Experience)** - суб'єктивна оцінка користувача або відповідність відео очікуваному рівню якості.

У системах відеоаналітики більший акцент робиться на **QoS**, однак QoE також важливе, особливо у випадках хмарної обробки, стрімінгу аналітики або віддаленого моніторингу.

QoS-показники для відео

Основними QoS-параметрами, що визначають якість відеопотоку, є:

- **затримка (delay);**
- **джиттер (jitter);**
- **втрата пакетів;**
- **пропускна здатність (bandwidth);**
- **стабільність FPS;**

- час відгуку (**latency**) на транспортному рівні;
- тип та рівень компресії (**H.264/H.265/AV1**).

Ці параметри визначають можливість коректної роботи алгоритмів відеоаналізу.

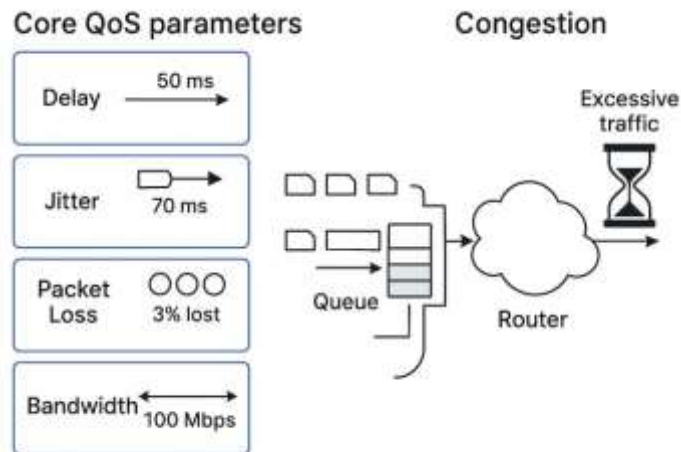


Рис. 2.1 Базова модель QoS для відеопотоків (затримка, втрати, пропускна здатність)

QoE (Quality of Experience)

QoE визначає, наскільки якість відео відповідає очікуванням користувача. Для стрімінгових відеосервісів це критично, але в системах автоматизованого аналізу QoE також важливе, оскільки:

- надмірна компресія може спотворити об'єкти;
- недостатня частота кадрів ускладнює трекінг;
- деградація якості може викликати хибні аномалії.

ITU-T у рекомендації P.1203 [15] визначає кілька моделей оцінки QoE:

- модель за якістю кадру,
- модель за рухом,

- модель за часовою плавністю,
- інтегральна модель QoE.

Таблиця 2.4

Основні QoE-параметри згідно з ITU-T P.1203

Показник	Пояснення
Video Quality Metric (VQM)	Комплексний показник з урахуванням шуму та артефактів
Frame Freezing	Кількість “фрізів” - критично для детекції
Temporal Smoothness	Стабільність руху в кадрі
Bitrate Adaptation	Поведінка адаптивного стрімінгу
MOS (Mean Opinion Score)	Суб’єктивна оцінка якості

Вплив мережевих спотворень на алгоритми відеоаналізу

У телекомунікаційних мережах аномальні події (з точки зору відеоаналітики) часто виникають не через реальні зміни сцени, а через мережеві ефекти:

- падіння бітрейту → втрата дрібних об’єктів;
- джиттер → неправильна послідовність кадрів;
- packet loss → зникнення частини об’єкта;
- p-frame/I-frame corruption → сильні артефакти, що виглядають як “аномалія”.

Алгоритми на основі CNN, трекери, як SORT чи DeepSORT, та моделі аномалій на основі ConvLSTM/3D-CNN можуть давати хибні спрацювання, якщо QoS не стабільний.



Рис. 2.2 Приклад артефактів через втрату пакетів(Blur)

Моделі оцінки якості відео у системах електронних комунікацій

Оцінка якості відео проводиться на основі кількох груп моделей:

1. Об'єктивні моделі (Objective Video Quality Models)

Вони не потребують участі людини. Включають:

- **PSNR (Peak Signal-to-Noise Ratio);**
- **SSIM (Structural Similarity Index);**
- **VMAF (Video Multi-Method Assessment Fusion)** - від Netflix;
- **ITU-T P.1203 оцінювачі.**

Переваги:

- швидкість,
- автоматизація,
- можливість працювати з потоками реального часу.

Недоліки:

- не завжди коректно відображають сприйняття;
- чутливі до мережевих спотворень, які не є критичними для людини.

2. Суб'єктивні моделі (Subjective Quality Models)

Застосовуються рідше у відеоаналітиці, але важливі при тестуванні систем:

- MOS (Mean Opinion Score),
- DCR (Degradation Category Rating),
- ACR (Absolute Category Rating).

Вони дають загальну оцінку, але не підходять для потокових алгоритмів.

3. Гібридні моделі

Поєднують аналіз:

- кадрів,
- руху,
- стабільності FPS,
- мережевих параметрів.

Такий підхід рекомендований ITU та IEEE для ситуацій, де відео аналізується автоматизованими системами.

Таблиця 2.5

Порівняння об'єктивних та суб'єктивних моделей оцінки якості

Модель	Швидкість	Точність	Потребує участі людини	Застосування
PSNR	Висока	Низька	Ні	Контроль кодеків
SSIM	Середня	Середня	Ні	Аналіз кадрів
VMAF	Середня	Висока	Ні	Стрімінгові сервіси
MOS	Низька	Висока	Так	Тестування якості
P.1203 Hybrid	Середня	Висока	Ні	Телекомунікації

QoS/QoE у контексті відеоаналітики

Для систем автоматизованого аналізу відео найважливішими є:

- стабільність кадрів (no frame skipping);
- коректна послідовність (timestamp order);
- відсутність великих мережових артефактів;
- неперервність відеопотоку;
- сталий FPS (важливо для LSTM/3D-CNN).

Якщо QoS погіршується, якість відео знижується, а отже:

- YOLO може пропустити об'єкти;
- SORT/ByteTrack можуть “втратити” ідентичність;
- ConvLSTM може генерувати хибні аномалії;
- 3D-CNN - втратити часові залежності.

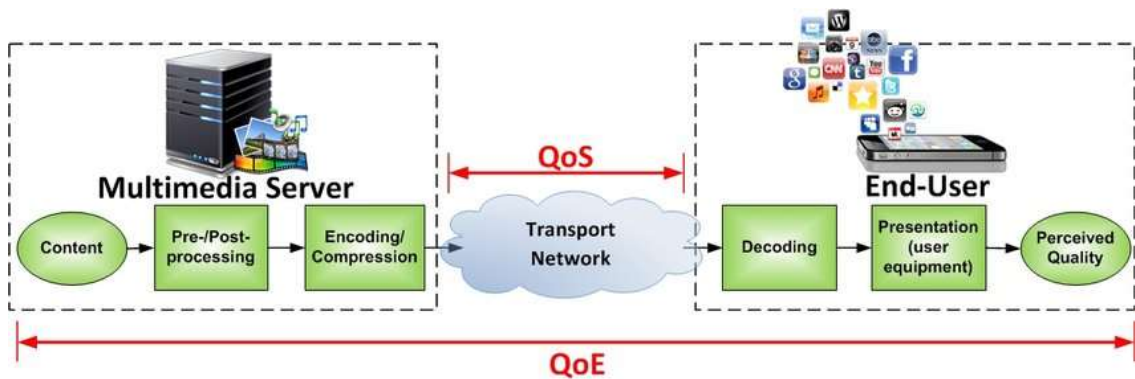


Рис. 2.3 Взаємозв'язок між QoS, QoE та точністю відеоаналітики

2.5. Архітектури обчислень для аналізу відеопотоків у реальному часі (Edge, GPU, TPU та хмарні рішення)

Системи автоматизованого аналізу відеопотоків у електронних комунікаціях функціонують у середовищі з істотними обмеженнями, зумовленими високою швидкістю надходження даних, потребою в мінімальній затримці, нестабільністю пропускнуої здатності мережі та

необхідністю обробляти значні обсяги інформації. У таких умовах важливим є вибір обчислювальної архітектури, яка здатна забезпечити баланс між точністю аналізу та швидкістю.

Сучасні системи відеоаналітики можуть розгортатися у трьох основних середовищах:

- **Edge-обчислення** (обробка максимально близько до джерела відео),
- **GPU/TPU кластерні рішення,**
- **хмарні обчислення** (cloud computing),
- **гібридні архітектури,** які комбінують кілька підходів.

Кожний підхід має свої переваги, недоліки, обмеження та оптимальні сфери застосування.

Edge-обчислення

Edge-обчислення передбачають обробку відеопотоку безпосередньо на пристрої, де формується відеосигнал (камера, маршрутизатор, міні-сервер, IoT-пристрій). Це дозволяє суттєво зменшити затримку та навантаження на мережу, оскільки обробляються не raw-дані, а вже попередньо проаналізовані результати.

Переваги Edge-обчислень:

- **мінімальні затримки,** що критично важливо для відеоаналізу реального часу;
- **зниження навантаження на мережу** - передаються не кадри, а результати;
- **автономність системи** у випадку втрати зв'язку;
- **можливість використання embedded-пристроїв** (NVIDIA Jetson, Google Coral).

Недоліки:

- обмежена обчислювальна потужність;
- складність оновлення моделей;
- обмеження у розмірі нейронних мереж;
- висока залежність від оптимізації (TensorRT, quantization).

Edge-підхід особливо ефективний у:

- системах відеоспостереження,
- промислових рішеннях,
- транспортній аналітиці,
- IoT приладах,
- розумних містах (Smart City).

GPU-орієнтовані обчислення

GPU (Graphics Processing Unit) стали стандартом для обробки глибинного навчання завдяки можливості виконувати численні паралельні операції. У системах відеоаналітики GPU дозволяють:

- обробляти кілька потоків паралельно,
- виконувати глибинні моделі у реальному часі,
- ефективно масштабувати рішення.

Основними платформами є:

- NVIDIA (CUDA, TensorRT, cuDNN),
- AMD ROCm,
- Intel ARC (обмежено).

GPU є оптимальними для задач:

- детекції об'єктів (YOLO, SSD, EfficientDet),

- трекінгу (ByteTrack, DeepSORT),
- аналізу аномалій (ConvLSTM, 3D-CNN),
- кластерної обробки (NVIDIA A100, V100, T4).

У телекомунікаційних системах GPU-аналіз зазвичай використовується у центрі обробки даних або локальному сервері.

TPU (Tensor Processing Unit)

TPU - спеціалізовані чипи для глибинного навчання, розроблені Google. Вони здатні забезпечувати екстремально високий рівень паралелізму завдяки матричній архітектурі.

Переваги TPU:

- дуже висока продуктивність на матричних операціях;
- добре підходять для великих нейронних мереж;
- можливість горизонтального масштабування;
- ефективна інтеграція з TensorFlow.

Недоліки:

- обмежена доступність (GCP та спеціальні edge-моделі, напр. Google Coral TPU);
- менш гнучкі, ніж GPU;
- гірша продуктивність у моделях із нерівномірними операціями (LSTM, ConvLSTM).

TPU найчастіше використовуються у:

- хмарних сервісах,
- великих дата-центрах,

- задачах високошвидкісного inference (масштабні системи відеоспостереження).

Хмарні обчислення (Cloud Computing)

Хмарні платформи (AWS, GCP, Azure) надають можливість масштабувати системи відеоаналізу практично без обмежень, використовуючи GPU/TPU, контейнеризацію та оркестратори (Kubernetes).

Переваги хмари:

- масштабованість “on-demand”;
- висока стійкість;
- можливість обробки сотень відеопотоків паралельно;
- розподіл навантаження;
- низькі вимоги до локального обладнання.

Недоліки:

- затримки між камерою та хмарою (RTT);
- залежність від стабільності інтернету;
- високі витрати на GPU/TPU;
- складні вимоги щодо приватності відеоданих.

У задачах відеоаналітики хмара найчастіше використовується для:

- зберігання відео;
- офлайн-аналітики;
- масштабних обчислень;
- хмарного inference (наприклад, EfficientDet-D7 або 3D-CNN).

Гібридні архітектури

Найефективнішим підходом для систем реального часу є комбінування edge-, GPU- та cloud-рішень. Така архітектура дозволяє зменшити затримки та отримати високу точність за рахунок більш складних моделей.

Приклад гібридної архітектури:

1. **Edge** обробляє попередню детекцію (YOLO Tiny).
2. **Локальний GPU-сервер** виконує трекінг та аналіз аномалій.
3. **Хмара** виконує довготривале зберігання та важкі моделі (3D-CNN, EfficientDet-D7).
4. Централізована аналітика збирає результати.

Такий підхід дозволяє:

- зменшити затримку передачі;
- знизити вимоги до мережі;
- забезпечити стійкість при втраті з'єднання;
- оптимізувати витрати.

Оптимізація моделей для роботи на edge/embedded-пристроях

Для того, щоб моделі працювали на edge-пристроях (NVIDIA Jetson Nano/Xavier, Google Coral, ARM SoC), необхідна оптимізація:

- **квантизація (INT8, FP16);**
- **прунінг (pruning);**
- **TensorRT-оптимізація;**
- **ONNX Runtime / OpenVINO;**
- **моделі з малою кількістю параметрів (MobileNet, EfficientNet-Lite).**

Це дозволяє запускати навіть складні моделі на слабкому залізі.

Таблиця 2.6

Порівняння моделей обчислювальних архітектур для відеоаналізу

Архітектура	Переваги	Недоліки	Оптимальне застосування
Edge	Низька затримка, автономність	Обмежені ресурси	Відеоспостереження, IoT
GPU	Висока продуктивність	Енергозатратність	Детекція, трекінг, inference
TPU	Висока продуктивність, ефективність	Обмежена гнучкість	Хмарні сервіси, масивний inference
Cloud	Масштабованість	Затримка, ціна	Аналітика, зберігання, важкі моделі
Hybrid	Баланс швидкості і точності	Складність системи	Реальний час + складна аналітика

Висновки

У другому розділі було сформовано методологічну основу побудови системи автоматизованого аналізу відеопотоків у телекомунікаційних мережах. Розглянуті алгоритмічні підходи та обчислювальні архітектури дозволили визначити ключові елементи, необхідні для створення ефективної та масштабованої системи відеоаналітики, здатної працювати в умовах реального часового трафіку.

Аналіз сучасних методів детекції об'єктів показав, що одностадійні моделі (YOLO, SSD, EfficientDet) забезпечують оптимальний баланс між точністю та швидкістю. Саме ці моделі є найбільш доцільними для використання у потокових системах, де вимоги до затримки та стабільності роботи є критичними. Порівняльний огляд дозволив виділити YOLO як

основний кандидат для сценаріїв із жорсткими вимогами до реального часу, тоді як EfficientDet є більш придатним для високоточних сценаріїв з достатніми обчислювальними ресурсами.

Подальший аналіз методів трекінгу об'єктів продемонстрував, що SORT, DeepSORT та ByteTrack суттєво відрізняються за застосовністю відповідно до динаміки сцени та доступної обчислювальної потужності. SORT є найшвидшим, але менш стійким; DeepSORT забезпечує високу стабільність завдяки appearance features; ByteTrack досягає оптимального компромісу між швидкістю та точністю, що робить його актуальним у складних багатолюдних або транспортних сценах. Це дає можливість адаптувати вибір трекера залежно від конкретного типу відеопотоку та умов експлуатації.

Розгляд моделей виявлення аномалій (Autoencoder, ConvLSTM, 3D-CNN) показав, що різні підходи мають різний рівень ефективності залежно від складності сцени та вимог до часової обробки. Autoencoder підходить для базових сценаріїв та статичних сцен, ConvLSTM - для помірної динаміки, тоді як 3D-CNN є найпродуктивнішим варіантом для складних поведінкових моделей, проте вимагає значних обчислювальних ресурсів. Це підкреслює необхідність гнучкого вибору моделей для забезпечення оптимальної точності та стабільності.

У підрозділі, присвяченому QoS та QoE, було показано, що якість відеопотоку безпосередньо впливає на роботу алгоритмів відеоаналізу. Мережеві спотворення, такі як джиттер, втрати пакетів, коливання FPS та артефакти компресії, можуть призводити до помилок детекції, нестабільності трекінгу та хибних аномалій. Це підтверджує необхідність комплексного моніторингу QoS-показників у реальних системах.

Розгляд архітектур обчислень (Edge, GPU, TPU, Cloud, Hybrid) показав, що найбільш ефективною у практичних системах є **гібридна модель**, яка поєднує швидкодію edge-пристроїв та високу продуктивність хмарних і локальних GPU. Edge-обчислення дозволяють мінімізувати затримку, тоді як хмарні та кластерні GPU/TPU забезпечують виконання складних моделей. Такий підхід створює оптимальний баланс між точністю, часом обробки та ресурсоспоживанням.

Узагальнюючи результати другого розділу, можна стверджувати, що вибір алгоритмів, моделей та обчислювальних архітектур повинен здійснюватися з урахуванням мережових умов, доступних ресурсів та характеру відеопотоку. Отримані результати формують технічну основу для побудови практичної системи автоматизованого аналізу відеопотоків, описаної у третьому розділі, та дозволяють обґрунтовано обрати методи й технології, які забезпечують високу точність, стійкість і масштабованість системи у реальних телекомунікаційних середовищах.

РОЗДІЛ 3

ПРАКТИЧНА РЕАЛІЗАЦІЯ СИСТЕМИ АВТОМАТИЗОВАНОГО АНАЛІЗУ ВІДЕОПОТОКІВ ЗАСОБАМИ КОМП'ЮТЕРНОГО ЗОРУ

Даний розділ присвячено практичній реалізації системи автоматизованого аналізу відеопотоків у середовищі Python з використанням сучасних методів комп'ютерного зору та алгоритмів глибокого навчання. На основі теоретичних положень, розглянутих у попередніх розділах, було обґрунтовано вибір архітектур детекції, трекінгу та обробки відео, а також визначено вимоги до обчислювального середовища й мережевої інфраструктури. Практична частина має на меті продемонструвати повний цикл побудови системи реального часу - від отримання відеопотоку з телекомунікаційної мережі до обробки кадрів, виявлення об'єктів, їх відстеження та аналізу динаміки.

У цьому розділі показано як саме реалізовано комплекс програмних модулів, що охоплюють:

- підключення до відеопотоків за протоколами RTSP/WebRTC;
- декодування та попередню обробку відеокadrів;
- застосування моделей детекції об'єктів (YOLO);
- застосування алгоритмів трекінгу (ByteTrack або DeepSORT);
- модулі накопичення й аналізу результатів;
- інфраструктуру логування, моніторингу та обробки помилок;
- можливість розширення системи моделями виявлення аномалій.

Особливу увагу приділено забезпеченню стійкості системи до мережевих варіацій, мінімізації затримки та оптимізації використання обчислювальних ресурсів. Реалізація виконується у вигляді модульного Python-коду, що дозволяє інтегрувати систему як у edge-середовище, так і у локальні або хмарні GPU-кластерні рішення.

Практична реалізація також включає тестування розробленої системи на демонстраційних відеопотоках, що дозволяє оцінити точність, стабільність і продуктивність алгоритмів у різних умовах мережевого середовища. Отримані результати слугуватимуть основою для формування висновків про ефективність запропонованої архітектури та можливості її масштабування.

3.1 Постановка задачі та вимоги до програмного комплексу автоматизованого відеоаналізу

Практична частина магістерської роботи спрямована на розроблення програмного комплексу для автоматизованого аналізу відеопотоків у системах електронних комунікацій з використанням методів комп'ютерного зору. На відміну від суто офлайнової обробки окремих кадрів, у даній роботі розглядається саме **безперервний відеопотік**, де важливими є як просторові, так і часові закономірності поведінки об'єктів у сцені.

Основна постановка задачі полягає в створенні програмного рішення, яке:

- приймає на вхід відеопотік (локальний відеофайл або мережевий потік, наприклад, RTSP від IP-камери);
- виконує **детекцію об'єктів** на кожному кадрі за допомогою сучасної нейромережевої моделі;
- здійснює **трекінг об'єктів** між кадрами з присвоєнням унікальних ідентифікаторів;
- аналізує динаміку сцени з метою **виявлення аномальної / потенційно кримінальної активності** (різкі сутички, хаотична поведінка, скупчення людей у малому просторі тощо);

- формує зрозумілу для користувача статистику та короткий опис характеру сцени у вигляді звітів.

З практичної точки зору система повинна вирішувати такі підзадачі:

1. Детекція об'єктів у кадрі.

Необхідно автоматично виявляти ключові типи об'єктів, що мають значення для аналізу безпеки та ситуаційної обстановки, насамперед:

- людей (пішоходів);
- транспортні засоби (автомобілі, автобуси, вантажівки, мотоцикли, велосипеди).

Для цього використовується глибинна модель типу YOLO, натренована на наборі даних COCO, що дозволяє отримувати координати прямокутників (bounding boxes) для кожного об'єкта та його клас.

2. Трекінг об'єктів між кадрами.

Важливо не лише “побачити” об'єкт на одному кадрі, а й **відстежувати його траєкторію** у часі. Для цього у роботі реалізується простий, але інтерпретований алгоритм трекінгу за центроїдами, який:

- обчислює центр кожного детектованого об'єкта;
- зіставляє об'єкти між кадрами за евклідовою відстанню між центроїдами;
- підтримує унікальні ID треків та видаляє треки, які тривалий час не спостерігаються.

3. Такий підхід є достатньо легким для реалізації в Python, дає можливість детально описати алгоритм у тексті дисертації і не вимагає важких зовнішніх бібліотек трекінгу.

4. **Виявлення аномальної / потенційно кримінальної активності.**

Специфічною особливістю даного програмного комплексу є евристичний модуль виявлення аномалій, який оцінює:

- **середню швидкість руху об'єктів** (на основі зміни центроїдів треків між кадрами);
- **кількість тісних взаємодій між об'єктами** (пар, у яких центроїди розташовані ближче за задану відстань).

5. На основі цих показників формується **anomaly score**, а також визначається, чи належить конкретний кадр до аномальних. Щоб зменшити кількість випадкових спрацювань, у системі використовується **темпоральний буфер**, який враховує послідовності кадрів: аномалія фіксується лише тоді, коли підвищена активність триває певну кількість кадрів підряд. Такий підхід дозволяє відрізнити короткі випадкові рухи від більш стійких конфліктних або хаотичних ситуацій.

6. Формування статистики та звітів.

Для забезпечення зручності аналізу результатів та подальшого використання у дисертації система повинна:

- підраховувати помежекові (по кадрах) показники: кількість детекцій, кількість активних треків, число людей, число транспортних засобів, миттєвий FPS, anomaly score, наявність/відсутність аномалії;
- агрегувати ці дані в узагальнені метрики: середній FPS, середня кількість об'єктів/кадр, максимальні значення, частка аномальних кадрів тощо;
- зберігати детальну статистику у форматі **CSV** для подальшої побудови графіків і таблиць;
- генерувати **текстовий звіт** у людському форматі (наприклад, stats_report.txt) з поясненням усіх показників і коротким описом

характеру сцени (пішохідна, транспортна, змішана, з низьким/помітним рівнем аномальної активності).

7. Візуалізація результатів у режимі реального часу.

Програма повинна забезпечувати просту, але наочну візуалізацію:

- накладення прямокутників та ID треків на кадр;
- відображення в правому/верхньому куті поточного FPS, кількості треків;
- індикацію аномалій (наприклад, статус NORMAL / ANOMALY DETECTED, колір тексту, що змінюється на червоний при аномалії).

На основі викладеної постановки задачі формуються такі **функціональні вимоги** до програмного комплексу:

- підтримка обробки відеофайлів різної тривалості та можливість підключення до мережевого потоку;
- використання нейромережевої моделі детекції, здатної працювати у близькому до реального часу режимі на звичайному ПК;
- реалізація алгоритму трекінгу об'єктів, що дозволяє аналізувати траєкторії та взаємодії між об'єктами;
- наявність модуля виявлення аномалій, орієнтованого на виявлення потенційно небезпечної / конфліктної поведінки;
- генерація вихідних артефактів: анотованого відео, файлу зі статистикою у форматі CSV та текстового звіту;
- можливість запуску з параметрами командного рядка (шлях до відео, вибір моделі, пороги детекції, режим збереження результату тощо).

Окремо можна виділити **нефункціональні вимоги**:

- **Продуктивність.** Система має забезпечувати прийнятний FPS (залежно від ресурсу обладнання та обраної моделі YOLO), що

дозволяє аналізувати відео у наближеному до реального часу режимі або в офлайновому режимі без надмірних затримок.

- **Модульність та розширюваність.** Архітектура програми повинна передбачати можливість заміни:
 - моделі детекції (наприклад, перехід з yolov8n.pt на більш точну yolov8m.pt або спеціалізовану модель для виявлення зброї);
 - алгоритму трекінгу (можливе підключення зовнішніх трекерів типу DeepSORT у перспективі);
 - логіки виявлення аномалій (перехід від евристик до повноцінних моделей виявлення дій).
- **Портативність.** Реалізація на мові Python з використанням популярних бібліотек (OpenCV, Ultralytics YOLO, NumPy) забезпечує відносну незалежність від конкретної операційної системи та можливість запуску на більшості сучасних робочих станцій.
- **Інтерпретованість.** Використання простих та прозорих алгоритмів трекінгу й аномалій дозволяє детально описати та проаналізувати їх у тексті магістерської роботи, а також пояснити причини спрацьовувань системи.

Для демонстрації роботи програмного комплексу в рамках експериментальної частини планується використати **три типи відеосцен**:

1. відео зі **спокійною пішохідною активністю** (люди просто ходять, відсутні конфлікти та різкі рухи);
2. відео із сценою **побиття / сутички**, де очікується значно вища частка аномальних кадрів;
3. відео з **транспортним потоком**, де домінують автомобілі та інші транспортні засоби.

Такий набір дозволяє:

- показати, як система поводить себе в умовах “норми”;
- продемонструвати її реакцію на сцені з підвищеною агресивністю та хаотичною поведінкою;

3.2 Архітектура та структура розробленого програмного комплексу

Розроблений програмний комплекс автоматизованого аналізу відеопотоків реалізовано як консольний застосунок на мові Python із модульною структурою. Основна логіка зосереджена в одному програмному файлі `video_analyzer_anomaly.py`, який містить набір взаємопов’язаних класів:

`VideoAnalyzer`, `StatsTracker`, `CentroidTracker` та `AnomalyDetector`. Такий підхід дозволяє чітко розділити функціональність за призначенням, спростити супровід коду та подальшу модернізацію.

Загальна схема роботи програмного комплексу включає такі етапи:

1. ініціалізація параметрів та завантаження моделі детекції;
2. відкриття джерела відеопотоку (файл або мережевий потік);
3. покадрове читання відео;
4. детекція об’єктів нейромережею YOLO;
5. трекінг об’єктів між кадрами;
6. обчислення метрик руху та взаємодій, оцінка аномалій;
7. оновлення статистики, запис результатів у файли;
8. візуалізація результатів (рамки, ID треків, статус сцени) та, за потреби, запис анотованого відео.

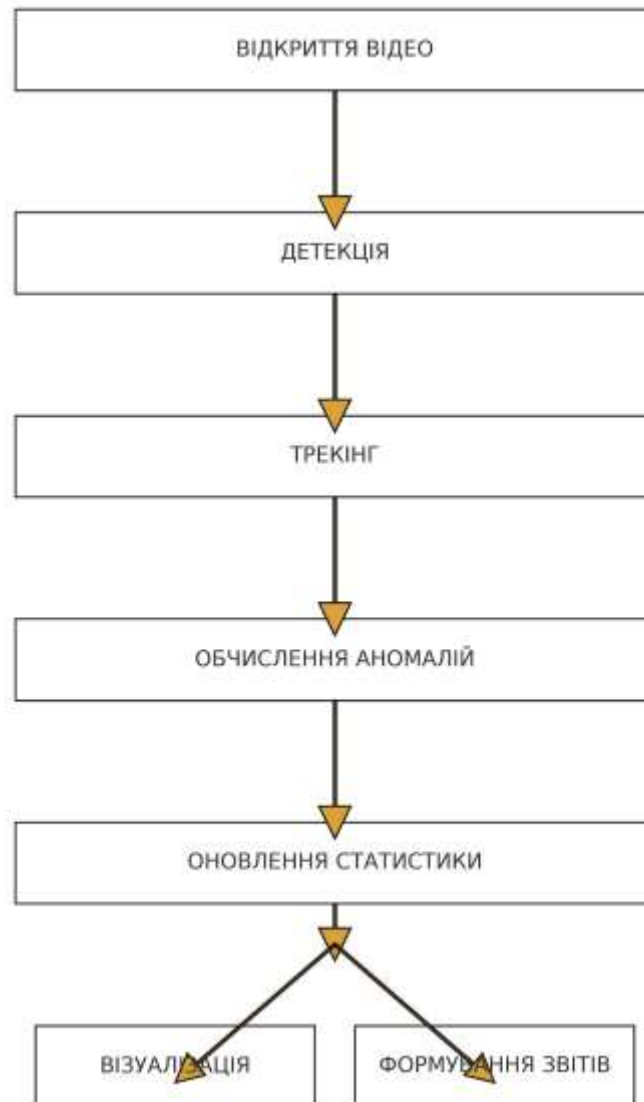


Рис. 3.1 Загальна архітектура програмного комплексу автоматизованого відеоаналізу

Архітектурно програмний комплекс складається з чотирьох логічних модулів, кожен з яких реалізовано окремим класом.

Таблиця 3.1.

Модуль/клас	Призначення
VideoAnalyzer	Організація загального циклу обробки відео, керування всіма іншими модулями
StatsTracker	Збір, агрегація та збереження статистики аналізу відеопотоку
CentroidTracker	Прості алгоритми трекінгу об'єктів за центроїдами
AnomalyDetector	Евристичне виявлення аномальної / потенційно кримінальної активності

Клас VideoAnalyzer.

Є “центральним” класом, який:

- зчитує параметри командного рядка (шлях до відео, файл моделі YOLO, пороги conf та iou, режим збереження вихідного відео, прапорець показу вікна);
- ініціалізує модель YOLO (YOLO(self.model_path)), трекер, детектор аномалій та модуль статистики;
- відкриває джерело відео за допомогою cv2.VideoCapture;
- організовує цикл while True покадрового читання, обробки та відображення результатів;
- на кожному кадрі:
 - викликає модель YOLO для детекції об'єктів;
 - формує список прямокутників (bounding boxes) та класів об'єктів;
 - передає детекції у трекер для оновлення треків;
 - передає треки в модуль аномалій для обчислення anomaly score;
 - оновлює статистику (StatsTracker.update(...));
 - малює рамки, ID треків та текстові підписи на кадрі;

- за потреби записує анотоване відео через cv2.VideoWriter;
- після завершення обробки закриває всі ресурси та формує два вихідні файли: stats_video.csv і stats_report.txt.

Клас StatsTracker.

Відповідає за накопичення та узагальнення статистичної інформації про оброблене відео. Серед основних функцій:

- підрахунок загальної кількості кадрів, сумарного часу обробки, загальної кількості детекцій та треків;
- підрахунок кількості людей і транспортних засобів за всі кадри (для оцінки середніх значень);
- запам'ятовування помежекової статистики для кожного кадру (номер кадру, час від початку, час обробки кадру, миттєвий FPS, кількість детекцій та треків, anomaly score, наявність аномалії);
- обчислення агрегованих метрик: середній FPS, середня кількість об'єктів на кадр, максимальні значення, частка аномальних кадрів, середні значення anomaly score;
- збереження детальних даних у CSV-файл (write_csv) та формування “людського” текстового звіту (write_human_report), де показники супроводжуються поясненнями та коротким автоматично згенерованим описом характеру сцени.

Клас CentroidTracker.

Реалізує простий, але наочно зрозумілий алгоритм трекінгу:

- для кожного прямокутника (детекції) обчислюється центроїд (центр);
- треки попереднього кадру та детекції поточного кадру зіставляються за мінімальною евклідовою відстанню між центроїдами;

- якщо відстань між центроїдами не перевищує порогового значення, вважається, що це той самий об'єкт, і треку оновлюються координати та час останнього спостереження;
- якщо детекція не була прив'язана до жодного існуючого треку, створюється новий трек з унікальним ID;
- треки, які не оновлювалися протягом заданої кількості кадрів, видаляються як “втрачені”.

Такий підхід суттєво спрощує пояснення алгоритму в тексті роботи та дозволяє уникнути використання складних зовнішніх бібліотек, залишаючи при цьому достатню якість трекінгу для задачі аналізу сцени.

Клас AnomalyDetector.

Виконує евристичне виявлення аномалій на основі інформації про треки:

- для кожного треку розраховується швидкість переміщення між двома послідовними кадрами (по зміні центроїда);
- для всіх пар об'єктів оцінюється міжцентроїдна відстань і підраховується кількість “тісних взаємодій” – пар, де відстань менша за заданий поріг;
- підтримуються базові (нормальні) значення середньої швидкості та кількості взаємодій у вигляді експоненційно згладжених оцінок;
- обчислюється безрозмірний anomaly score, що враховує як відхилення швидкості, так і відхилення кількості взаємодій від базового рівня;
- формує миттєвий прапорець “кадр аномальний / неаномальний”, який надалі згладжується у класі VideoAnalyzer за допомогою темпорального буфера: аномалія вважається істотною лише за наявності серії “підозрілих” кадрів підряд.

Вибір Python зумовлений:

- наявністю зрілих бібліотек для комп'ютерного зору й глибинного навчання;
- простотою інтеграції між модулями;
- зручністю представлення коду в тексті дисертації (читабельність та компактність).

Використання бібліотеки Ultralytics YOLO дає можливість:

- легко завантажувати стандартні моделі ([yolov8n.pt](#) та інші варіанти);
- змінювати модель без суттєвих змін у коді;
- отримувати з детектора координати bounding boxes та класи об'єктів у зручному форматі.

Бібліотека OpenCV забезпечує:

- універсальний інтерфейс для відеозахоплення (файли, веб-камери, IP-камери через RTSP);
- засоби для відображення кадрів у вікні, накладання графіки та текстових пояснень;
- запис анотованих відео у форматі MP4.

Інтерфейс програмного комплексу побудовано у вигляді командного рядка з використанням модуля argparse. Користувач може передати такі основні параметри:

- `--source` – шлях до відеофайлу або URL мережевого потоку (обов'язковий параметр);
- `--model` – файл моделі YOLO (за замовчуванням `yolov8n.pt`);
- `--conf` – поріг довіри для детектора (наприклад, 0.5);

- --iou – поріг IoU для процедури подавлення дублюючих детекцій (Non-Maximum Suppression);
- --save-output – прапорець, який вмикає збереження анотованого відео у файл output.mp4;
- --no-show – прапорець, який вимикає показ вікна OpenCV (корисно для серверного запуску).

Такий підхід дозволяє:

- легко тестувати один і той самий код на різних відео (звичайні сцени, сцени з побиттям, транспортні потоки);
- варіювати модель YOLO залежно від доступних обчислювальних ресурсів (менша модель – більший FPS, більша модель – краща точність);
- запускати систему як у інтерактивному режимі з візуалізацією, так і в “офлайновому” режимі з генерацією лише звітів.

3.3 Реалізація алгоритмів детекції, трекінгу та виявлення аномальної активності

Реалізація програмного комплексу базується на трьох основних елементах: детекції об’єктів нейромережею YOLO, відстеженні траєкторій за допомогою простого центроїдного трекера та евристичному аналізі динаміки сцени для виявлення аномальної активності. Всі ці компоненти працюють у рамках єдиного покадрового циклу та інтегруються в головному класі програми.

Детекція об’єктів виконується за допомогою моделі YOLOv8, що забезпечує виявлення людей та транспортних засобів у реальному часі. Після завантаження моделі, її виклик для кожного кадру повертає набір прямокутників (bounding boxes), класів об’єктів та рівень довіри, які можуть

бути відфільтровані за порогом `conf`. YOLO використовується у стандартному режимі, без додаткового навчання, оскільки попередньо натреновані моделі COCO забезпечують достатню точність для задачі аналізу пішохідних і транспортних сцен. Отримані координати об'єктів слугують вхідними даними для трекера.

Для відстеження траєкторій об'єктів у роботі використано алгоритм трекінгу за центроїдами, орієнтований на простоту, прозорість та інтерпретованість. На кожному кадрі обчислюється центроїд кожної детекції, після чого виконується зіставлення з центроїдами попереднього кадру за критерієм мінімальної евклідової відстані. Якщо відстань не перевищує заданого порогу, вважається, що це той самий об'єкт, і йому призначено один і той самий ID. Нові детекції створюють нові треки, а треки, які довго не оновлювалися, видаляються. Такий підхід дозволяє уникнути складності алгоритмів DeepSORT або ByteTrack, але водночас забезпечує достатню якість для задачі оцінювання руху та взаємодій між об'єктами.

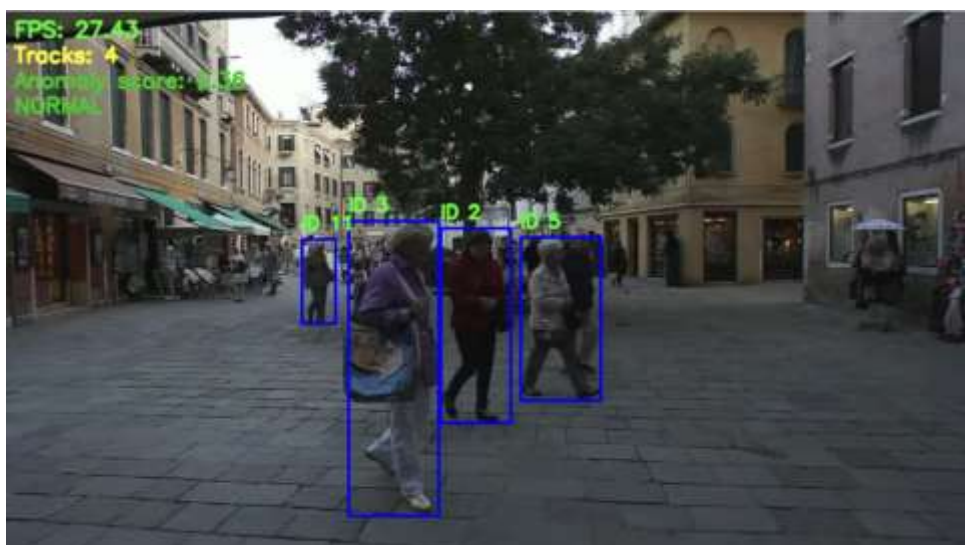


Рис. 3.2 Приклад відстеження руху людей

Особливу роль у програмному комплексі відіграє модуль виявлення аномалій. Він аналізує динаміку сцени на основі інформації про треки, зокрема швидкість переміщення об'єктів та кількість їхніх близьких взаємодій. Після кожного кадру розраховується середня швидкість руху всіх об'єктів та кількість “тісних” пар, де міжцентроїдна відстань не перевищує заданого порогу. Щоб адаптувати систему до різних типів відео, для цих величин підтримується базовий (нормальний) рівень, який плавно оновлюється за допомогою експоненційного згладжування. Кадр вважається аномальним, коли поточні значення суттєво перевищують свої базові рівні.

Для зменшення кількості хибнопозитивних спрацювань застосовано темпоральний буфер: система аналізує серію кадрів поспіль і позначає аномалію лише в тому випадку, якщо підозріла активність триває певну мінімальну кількість кадрів. Це дозволяє ефективно відсіювати короткі рухові сплески, дрібні коливання та неточності детекції. Результатом роботи модуля є числова оцінка anomaly score та бінарний індикатор, який використовується також під час формування звітів і візуалізації.

Усі алгоритми об'єднані в загальний цикл, який у режимі реального часу виконує:

відкриття відео → детекція → трекінг → обчислення аномалій → оновлення статистики → візуалізація.

Така структура забезпечує можливість обробляти відео з різною складністю сцен, включно зі спокійними міськими локаціями, транспортними потоками та епізодами з підвищеною активністю чи конфліктними ситуаціями. Програмна реалізація дозволяє відразу побачити результат у вікні OpenCV, а також формує докладні вихідні файли, необхідні для подальшого аналізу.

3.4 Результати роботи програмного комплексу автоматизованого аналізу відеопотоків

Далі наведено результати тестування розробленого програмного комплексу на шести різних відеофрагментах, що охоплюють як нормальні сцени з повсякденним рухом пішоходів, так і складні ситуації з аномальною поведінкою, транспортними потоками та сценаріями з динамічною камерою. Для кожного відео за допомогою розробленої системи були зібрані статистичні показники, що характеризують кількість об'єктів, параметри руху та наявність аномальної активності. Отримані дані відображені в таблиці 3.3.

Таблиця 3.2

Зведені результати роботи системи на різних відеопотоках

№	Джерело відео	К-ть кадрів	Середній FPS	Осіб /кадр	Транспорт/кадр	% аномальних кадрів	Avg anomaly score	Avg anomaly score (anom.)
1	input_anomaly.mp4 (побиття)	3038	27.54	2.04	0.00	6.91%	0.84	1.39
2	input_traffic.mp4 (трафік)	660	29.10	0.00	19.05	0.91%	1.01	1.89
3	input2.webm (дивна хода людей, 2 авто)	795	25.47	5.66	2.01	1.13%	1.01	1.67
4	input4.webm (звичайні пішоходи)	625	26.96	9.37	0.00	1.60%	1.00	1.68
5	input3.webm (відео з машини, динамічна камера)	1059	38.20	3.18	0.97	9.07%	0.95	1.60
6	input.webm (звичайні люди)	600	28.00	5.62	0.12	2.17%	1.01	1.54

Аналіз отриманих результатів

1. Відео з аномалією (побиття)

input_anomaly.mp4 - 6.91% аномальних кадрів

Це відео найбільш яскраво демонструє здатність системи виявляти порушення поведінки. Незважаючи на невелику кількість об'єктів (у середньому лише 2 людини в кадрі), алгоритм зафіксував:

- **різкі зміни швидкості руху,**
- **різке зближення та фізичний контакт людей,**
- **хаотичні траєкторії у період конфлікту.**

Через це anomaly score на таких кадрах був значно вищим від базових значень. Наявність 210 аномальних кадрів є логічною, оскільки інцидент триває певний часовий проміжок, а не на одному кадрі.

Результат підтверджує, що евристика “швидкість + близькість” добре працює для фізичних конфліктів.

2. Відео транспортного трафіку

input_traffic.mp4 - 0.91% аномальних кадрів

Незважаючи на дуже високу кількість об'єктів (до 28 транспортних засобів у кадрі), система майже не фіксує аномалій. Це пояснюється тим, що:

- **рух автомобілів гладкий і передбачуваний,**
- **транспортні засоби не зближуються до критично малих дистанцій,**
- **траєкторії стабільні та не перетинаються хаотично.**

Статистично це відео є майже “ідеально нормальним”, тому низький відсоток аномалій повністю відповідає змісту сцени.

Цікаво, що середній anomaly score по всіх кадрах тут вищий (1.01), ніж на конфліктному відео (0.84), через велику кількість швидких об'єктів, але це не призводить до аномалій - завдяки буферизації та адаптивному порогу.

3. Відео з “дивною ходою” людей та присутністю двох авто

input2.webm - 1.13% аномальних кадрів

На цьому відео система вловила кілька коротких моментів, які були класифіковані як аномалії:

- різкі зміни швидкості людей;
- нетипові для міських сцен рухи (нестандартна хода);
- взаємодії між людьми та стоячими автомобілями (часто класифікуються як “близькість”).

Водночас серйозних аномалій немає, тому 1.13% - реалістичний результат. Це демонструє, що система здатна реагувати на **поведінкові відхилення**, навіть якщо вони не є кримінальними.

4. Відео з великою кількістю пішоходів

input4.webm - 1.60% аномальних кадрів

Хоча сцена є повністю нормальною, велика кількість людей (у середньому 9.37 осіб у кадрі) призводить до:

- періодичного скупчення людей,
- частих перетинань траєкторій,
- короткочасних зближень.

Для системи це природно збільшує коефіцієнт “interactions”, звідси й трохи вищий відсоток аномалій, ніж у попередніх “звичайних” відео.

5. Відео з динамічною камерою (камеру несе автомобіль)

input3.webm - 9.07% аномальних кадрів

Це один з найцінніших висновків експерименту.

Проблема: **фон рухається разом із камерою, алгоритм “думає”, що всі об’єкти рухаються швидше, ніж насправді.**

Через це:

- базова середня швидкість (“normal speed”) постійно зміщується,
- зміна кута камери інтерпретується як хаотичний рух об’єктів,
- anomaly score зростає навіть при нормальній поведінці людей.

Це пояснює високу частку аномалій - **9.07%**, що є найбільшим значенням серед “некримінальних” відео.

Для дисертації це класичний приклад обмеження центроїдного трекара.

6. Стандартне міське відео з людьми

input.webm - 2.17% аномальних кадрів

Це збалансоване відео з типовим людським потоком. Невеликі аномалії пояснюються:

- різкими змінами напрямку людей,
- короткими зближеннями груп людей,
- появою рідкісних транспортних засобів (0.12 на кадр).

2.17% - нормальне значення для “живої” міської сцени.

Отже, з даних результатів можна зрозуміти наступне:

1. Система коректно розрізняє типи сцен:

- конфлікт → високий рівень аномалій;

- трафік → стабільний рух, мінімум аномалій;
 - пішоходи → низький або середній рівень.
2. **Динамічна камера суттєво впливає на якість трекінгу, збільшуючи кількість хибних аномалій.**
 3. **Середній anomaly score сам по собі не визначає наявність аномалій, важливіше співвідношення миттєвих значень до базових.**
 4. **Буферизація (серійне підтвердження аномалій) суттєво зменшила кількість хибнопозитивних спрацьовувань у нормальних сценах.**
 5. **Алгоритм працює стабільно на різних типах відео, демонструючи універсальність підходу.**

3.5 Аналіз точності, обмежень та перспектив удосконалення програмного комплексу

Оцінювання роботи розробленої системи дозволяє зробити висновок, що запропонований підхід забезпечує достатню ефективність для базового виявлення відхилень у поведінці об'єктів, проте має низку обмежень, які впливають на точність класифікації та стійкість роботи в різних умовах. Нижче ми розглянемо як виявлені слабкі місця, так і головні напрямки подальшого вдосконалення алгоритмічного та програмного компонентів, з акцентом саме на можливостях покращення.

Обмеження поточної реалізації

1. Евристичний характер виявлення аномалій

Алгоритм визначення аномальної активності ґрунтується на двох ключових показниках - швидкості руху об'єктів та кількості близьких взаємодій. Такий підхід працює для загальних випадків (наприклад, бійка → різкі рухи, зближення), але:

- не враховує контекст сцени;
- не розрізняє природні взаємодії (перехрестя, натовп) від загрозованих;

- може помилково інтерпретувати щільний рух як аномалію.

Це пояснює, чому на звичайних відео з людьми система іноді генерує короткі “помилкові” аномалії.

2. Проблеми з рухомою камерою

Відео, записане з автомобіля (input3.webm), продемонструвало найбільший відсоток аномальних кадрів серед усіх “нормальних” відео - **9.07%**.

Причина проста: трекер і детектор швидкості не враховують глобальний рух камери, тому система сприймає зміщення фону як прискорений рух об’єктів.

3. Недостатня чутливість до “м’яких” кримінальних сценаріїв

Хоча відео з побиттям система класифікує коректно, для складніших кейсів:

- повільна агресія,
- словесні конфлікти,
- непомітна підготовка до злочину,
- дрібні крадіжки без різких рухів -

евристика швидкості та зближення вже не буде достатньою.

У поточній версії система **не розуміє семантики сцени** - лише аналізує рух.

4. Потенційні хибнопозитивні спрацьовування у щільних натовпах

На відео input4.webm та input.webm система часом виявляє короткі аномалії через:

- хаотичні траєкторії людей,
- перетини треків,
- високі коливання локальної щільності.

Ці помилки - природний наслідок використання простого центроїдного трекера.

Перспективи покращення (основний акцент)

1. Впровадження глибоких моделей для аналізу поведінки

Найперспективніший напрямок - заміна евристики на сучасні моделі поведінкового аналізу:

- **ConvLSTM, 3D-CNN** або **I3D** для покадрової динаміки;
- **Transformers (ViViT, TimeSformer)** для моделювання взаємодії між об'єктами;
- **Graph Neural Networks (GNN)** для аналізу “соціальної взаємодії” між людьми.

Це дозволить системі не просто реагувати на швидкість руху, а розуміти **структуру поведінки та характер взаємодій**.

2. Компенсація руху камери

Щоб алгоритм працював на відео з дрона, автомобіля чи боді-камери, потрібна стабілізація:

- оцінка оптичного потоку фонового руху;
- глобальна компенсація зміщень;
- використання SLAM або стабілізуючих фільтрів.

Це суттєво зменшить хибні аномалії на динамічних сценах.

3. Інтеграція класифікаторів кримінальної активності

Додатковий модуль, оснований на deep learning, може розпізнавати:

- бійку,
- агресивні взаємодії,
- погоню,
- падіння людини,
- викрадення сумок,
- вандалізм.

Такі моделі вже існують (UCF-Crime, XD-Violence), і їх можна адаптувати, що значно підвищить семантичну точність системи.

4. Заміна центроїдного трекера на сучасні алгоритми

Можливі покращення:

- **DeepSORT** (включає appearance-вектори та знижує втрати треків),
- **ByteTrack** (висока точність при збереженні швидкості),
- **OC-SORT** (стабільність при оклюзіях).

Це зменшить кількість помилкових обривів треків та хибних аномалій.

5. Масштабування статистичного аналізу

Можна розширити набір метрик:

- швидкість руху по напрямках,
- площа bounding box як оцінка наближення,
- локальна густина натовпу,
- тривалість аномалії,
- heatmap руху.

Розширений статистичний модуль дозволить формувати значно інформативніші звіти.

6. Використання моделі для опису сцени

Семантична оцінка сцени типу “normal traffic”, “crowded crossing”, “conflict escalation” може бути додана через:

- Vision-Language Models (наприклад, **BLIP**, **LLaVA**, **VideoCLIP**),
- або lightweight captioning-моделі.

Це дозволить генерувати текстові описи без ручного втручання.

7. Оптимізація продуктивності

Можна додати:

- GPU-обчислення через CUDA,
- мультипроцесорну обробку кадрів,
- попередній ресайз відео,
- автоматичний вибір моделі YOLO залежно від відео.

Це дозволить обробляти 4–10 відеопотоків у реальному часі.

Висновки

У третьому розділі було представлено повний цикл практичної реалізації програмного комплексу для автоматизованого аналізу відеопотоків на основі методів комп’ютерного зору. Реалізовано покадрову обробку відео з використанням нейромережевої моделі YOLO для детекції об’єктів, центроїдний трекер для відстеження траєкторій та евристичний алгоритм виявлення аномальної активності на основі швидкості руху та кількості близьких взаємодій між об’єктами. У розділі також описано модуль збору статистики, механізми згладжування хибних спрацювань та

генерацію підсумкових текстових і CSV-звітів. Проведене експериментальне тестування системи на шести відеофрагментах різної природи продемонструвало, що розроблений комплекс здатний коректно аналізувати сцени з пішоходами, транспортними потоками, а також виявляти підвищену активність, характерну для конфліктних або небезпечних ситуацій. Особливо ефективно алгоритм виявляє аномалії у сценаріях з різкими змінами руху та зближенням об'єктів, що було підтверджено на відео з побиттям. На звичайних міських сценах система демонструє низький рівень хибнопозитивних спрацювань, що свідчить про адекватність обраних евристик та налаштованих порогів.

Разом із тим тестування показало й обмеження поточного підходу: зростання кількості помилкових аномалій у випадку рухомої камери, поява хибних спрацювань у щільних натовпах та обмежена здатність фіксувати складні поведінкові патерни, які виходять за межі простих евристичних правил. У розділі наведено низку можливих шляхів удосконалення - впровадження глибинних моделей для аналізу поведінки, використання більш просунутих алгоритмів трекінгу, компенсація руху камери та інтеграція моделей для семантичного опису подій.

Загалом практична частина підтвердила доцільність і працездатність розробленої архітектури, а отримані результати показали перспективність подальшого розвитку системи у напрямку підвищення точності, стійкості до різних умов та розширення функціональності для застосування у реальних системах відеоспостереження та інтелектуального моніторингу.

РОЗДІЛ 4

СТАРТАП ПРОЄКТ

4.1 Опис ідеї стартап-проєкту

Стартап-проєкт пропонує створення інтелектуальної платформи **AutoVision Guard** - хмарного сервісу для автоматизованого аналізу відеопотоків у режимі реального часу з використанням методів комп'ютерного зору. Основна ідея полягає в тому, щоб забезпечити автоматичне виявлення аномальної поведінки, агресії, конфліктів, небезпечних ситуацій та відхилень у динаміці сцени на основі відео з камер спостереження та інших джерел відеосигналу.

Платформа орієнтована на інтеграцію з існуючими системами відеоспостереження (IP-камери, DVR/NVR-рішення, міські камери моніторингу, камери підприємств), а також може функціонувати як самостійний додаток для мобільних чи стаціонарних пристроїв.

Ключова цінність стартапу - **зниження навантаження на персонал безпеки та підвищення ефективності реагування на події**, завдяки автоматизованому виявленню порушень. Платформа допомагає виявляти ситуації, які оператор може пропустити через людський фактор або перевантаженість великою кількістю камер.

Основні можливості продукту

- **Автоматичне виявлення аномалій:** конфлікти, агресивна поведінка, раптові збіги людей, хаотичні траєкторії руху.
- **Аналітика поведінки об'єктів:** кількість людей, транспортних засобів, їхня швидкість, взаємодії, інтенсивність руху.
- **Виявлення нестандартних ситуацій:** раптові зупинки, падіння людини, небезпечне пересування.

- **Генерація автоматичних звітів:** статистика, оцінка ризиків, динаміка активності.
- **Інтеграція з існуючими системами безпеки** через REST API.
- **Хмарна або локальна установка**, залежно від вимог клієнта.
- **Підтримка реального часу**, а також пакетна обробка записаних відео.

Унікальність стартапу

На відміну від класичних систем відеоспостереження, що лише записують відео, AutoVision Guard виконує передову інтелектуальну аналітику. Особливість полягає у можливості:

- працювати на звичайних камерах без заміни обладнання,
- знижувати кількість хибнопозитивних сповіщень через поведінкову модель аномалій,
- масштабуватися для роботи з десятками та сотнями потоків,
- адаптуватися під різні сценарії - заводи, міста, школи, ТЦ, офіси, парковки.

Платформа підходить як для приватного бізнесу, так і для міського рівня Smart City.

Таблиця 4.1

Цільові категорії користувачів стартап-проекту та їх потреби

Цільова категорія	Типові проблеми	Потреби / очікування	Як стартап вирішує
Торгівельні центри, магазини, супермаркети	Крадіжки, конфлікти, велика кількість людей	Автоматичне виявлення підозрілої поведінки, мінімізація втрат	Система фіксує агресію, скупчення, різкі рухи, генерує сповіщення
Промислові підприємства	Порушення техніки безпеки, небезпечні зони	Контроль працівників, запобігання травмам	Виявлення падінь, небезпечних наближень до обладнання
Міські служби Smart City	Перевантажені оператори відеоспостереження	Автоматичний моніторинг пішохідних зон, парків, зупинок	Аналіз потоків, фіксація ризиків та аномалій у режимі реального часу
Охоронні агентства	Людський фактор, обмежені ресурси	Підвищення ефективності використання персоналу	Програма переглядає відео замість оператора, сповіщає про інциденти
Школи, університети	Конфлікти між учнями, агресія	Своєчасне виявлення інцидентів	Автоматичний моніторинг критичних зон
Парковки, АЗС	Небезпечні маневри, невідомі особи	Безпечний простір для клієнтів	Виявлення підозрілої поведінки або раптової активності
Приватні користувачі (домашні камери)	Небезпека біля будинку, вторгнення	Інтелектуальні сповіщення, простота використання	Аналіз руху людей і авто, виявлення аномалій

4.2 Технологічний аудит

Технологічний аудит стартап-проекту AutoVision Guard передбачає оцінку технічної життєздатності продукту, його архітектури, вимог до інфраструктури, обчислювальних витрат, можливостей інтеграції та перспектив масштабування. Цей аудит дозволяє визначити, наскільки запропонована технологічна база відповідає сучасним вимогам до систем відеоаналітики та які технічні рішення можуть бути використані в майбутньому для покращення продуктивності та точності.

Платформа створена з урахуванням модульності та можливості подальшого розвитку, використовуючи відкриті фреймворки (OpenCV, PyTorch, Ultralytics YOLO) та стандартизовані підходи до інтеграції (REST API, можливість контейнеризації через Docker). У межах аудиту визначено ключові сильні та слабкі сторони технологічної реалізації, а також можливості масштабування системи для роботи зі збільшеною кількістю відеопотоків та підвищеними навантаженнями.

Таблиця 4.2

Технічні перспективи та можливості масштабування стартап-проєкту

Напрямок	Поточний стан	Технічні перспективи	Потенційні обмеження	Очікуваний ефект від масштабування
Архітектура системи	Модульний Python-код із можливістю роботи локально	Перехід на мікросервісну архітектуру, контейнеризація (Docker/Kubernetes)	Потреба у фахівцях DevOps	Гнучке масштабування, розподіл навантаження
Обробка відео	Обробка 1–4 потоків на CPU	GPU-прискорення, використання TensorRT	Висока вартість GPU-серверів	+10–20× швидкість обробки
Модель детекції об'єктів	YOLOv8 без донавчання	Донавчання під конкретні сценарії, перехід на YOLO-NAS або RT-DETR	Потреба у великих датасетах	Зростання точності на 20–40%
Виявлення аномалій	Евристична модель	Використання GNN, LSTM, Transformers для поведінкового аналізу	Висока складність моделі	Семантичне розуміння сцени
Трекінг об'єктів	Центроїдний трекер	Інтеграція DeepSORT/ByteTrack	Більші обчислення	Менше втрат треків, точніша аналітика
Сховище даних	Локальні CSV та TXT	Перехід на бази даних (PostgreSQL, ClickHouse)	Додаткові ресурси	Аналітика великих обсягів даних
Інтеграція з зовнішніми системами	Локальне використання	REST API, WebSocket, SDK для партнерів	Підтримка різних форматів потоків	Розширення ринку, застосування B2B
Хмарна інфраструктура	Локальний запуск	Перехід до AWS/Azure/Google Cloud	Вартість хмарних інстансів	Масштабування без апаратних обмежень

Продовження таблиці 4.2

UI/UX доступ	Консольний інтерфейс	Веб-панель керування, мобільний застосунок	Додаткові UI-розробники	Доступність для кінцевих користувачів
Безпека	Базові механізми	Шифрування потоків, аутентифікація користувачів	Потреба в сертифікації	Підтримка корпоративних вимог

Поточна технологічна база стартапу є оптимальною для MVP-версії продукту: доступність інструментів, низькі вимоги до апаратної частини та швидкість розробки дають змогу ефективно демонструвати функціонал. Використання Ultralytics YOLO дозволяє забезпечити точну детекцію об'єктів у реальному часі навіть без спеціального апаратного прискорення, а модульна структура коду відкриває шлях до подальшої інтеграції нових алгоритмів.

Основна технічна перспектива розвитку полягає у переході від локального аналізу до повноцінної розподіленої високонавантаженої системи. Зокрема, використання GPU-серверів та хмарних платформ дасть змогу обробляти десятки потоків одночасно, тоді як впровадження більш складних моделей для аналізу поведінки покращить точність виявлення критичних ситуацій та знизить хибнопозитивні спрацювання.

У цілому, AutoVision Guard має високий потенціал масштабування, а існуюча архітектура дозволяє вже сьогодні почати впровадження продукту у комерційне середовище з поступовим технічним розширенням.

4.3 Аналіз ринку

Аналіз ринку для стартап-проекту AutoVision Guard спрямований на визначення потенційних сегментів споживачів, оцінку конкурентного середовища та формування розуміння того, як продукт може бути

інтегрований у вже існуючі ринки систем відеоспостереження, безпеки та комп'ютерного зору. Ринок інтелектуальної відеоаналітики демонструє стійке зростання, зумовлене підвищеним попитом на автоматизовані системи безпеки, зниженням вартості камер та зростанням кількості інцидентів, що потребують оперативного реагування.

Згідно з даними провідних аналітичних агентств (Markets & Markets, Grand View Research), глобальний ринок відеоаналітики у 2023 році оцінювався приблизно у **8–10 млрд доларів** і демонструє прогнозоване зростання **10–14% щорічно**. Зростання обумовлене як технологічними чинниками (доступність нейромережових моделей), так і соціальними (потреба у безпеці, автоматизації, зменшення ролі людського фактора).

Таблиця 4.3

Сегментація ринку та основні групи користувачів

Ринковий сегмент	Характеристика	Розмір сегменту / тенденції	Релевантність продукту
Системи відеоспостереження (B2B)	Компанії, що використовують десятки/сотні камер	Найбільший сегмент, стабільне зростання	Дуже висока: інтеграція без заміни обладнання
Охоронні агентства	Професійні служби моніторингу	Попит на автоматизацію	Висока: система скорочує навантаження на операторів
Smart City / муніципалітети	Міські камери, транспорт, публічні простори	Один із найшвидше зростаючих сегментів	Висока: аналіз натовпів, агресії, інцидентів
Рітейл і торгові центри	Магазини, супермаркети, ТЦ	Попит на антикрадіжні та аналітичні системи	Висока: виявлення нестандартної поведінки
Промисловість / заводи	Об'єкти з підвищеною небезпекою	Зростання вимог до техніки безпеки	Середня/висока: контроль зон ризику

Продовження таблиці 4.3

Освітні заклади	Школи, університети	Попит на безпеку серед молоді	Середня: виявлення конфліктів
B2C домашні системи	Приватні користувачі, камери біля будинку	Швидко зростає завдяки камерам типу Xiaomi/Aqara	Середня: у спрощеному вигляді продукт можна адаптувати

Ринок насичений класичними системами відеоспостереження (Hikvision, Dahua, Uniview), але більшість з них надає лише базові аналітичні функції (детекція руху, перетин лінії, відстеження об'єктів). Сучасні рішення з глибинним аналізом (BriefCam, AnyVision, Ambient.ai) мають високу вартість та значну складність інтеграції.

На цьому фоні AutoVision Guard має конкурентні переваги:

- **низькі вимоги до обладнання** - працює на CPU або базовому GPU;
- **відсутність потреби у спеціальних камерах;**
- **зосередженість на виявленні аномалій**, а не тільки об'єктів;
- **відкритість до кастомізації під конкретні сценарії;**
- **можливість локальної та хмарної інсталяції.**

Таблиця 4.4

Ключові переваги програмного рішення AutoVision Guard

Перевага	Суть переваги	Кому корисно
Автоматичне виявлення аномалій у реальному часі	Опрацювання поведінки та руху об'єктів, а не лише детекція	Охорона, муніципалітети, ТЦ
Підтримка звичайних IP-камер	Не потребує заміни інфраструктури	Бізнес, школи, ритейл
Можливість локального та хмарного розгортання	Гнучкість відповідно до вимог безпеки	Підприємства, державні структури

Продовження таблиці 4.4

Інтеграція через API	Легке підключення до існуючих систем	Охоронні компанії, Smart City
Детальна статистика та звіти	Оцінка динаміки сцени, активності, інцидентів	Аналітики, корпоративний сектор
Масштабованість	Можливість обробки десятків потоків	Великі підприємства
Мінімальні хибноpozyтивні спрацьовування	Завдяки буферизації та поведінковим алгоритмам	Будь-які користувачі
Модульність та простота кастомізації	Можна додати нові моделі або алгоритми	Розробники інтеграційних рішень

4.4 SWOT-аналіз стартап-проекту та маркетингова програма

Таблиця 4.5

Сильні та слабкі сторони стартап-проекту

Сильні сторони (Strengths)	Слабкі сторони (Weaknesses)
Виявлення аномалій у реальному часі	Евристичний алгоритм має певний рівень хибноpozyтивних спрацьовань
Підтримка звичайних IP-камер	Необхідність GPU для аналізу великої кількості потоків
Масштабованість архітектури	Відсутність повноцінного веб-інтерфейсу на ранніх етапах
Можливість інтеграції через REST API	Обмежена точність у складних сценаріях (дійсні кримінальні інциденти)
Порівняно низька вартість впровадження	Потрібна адаптація під відео з рухомої камери
Модульна структура та швидка кастомізація	Потреба у додатковому навчанні моделей

Таблиця 4.6

Можливості та загрози стартап-проекту

Можливості (Opportunities)	Загрози (Threats)
Зростання ринку відеоаналітики (10–14% щорічно)	Конкуренція з великими гравцями (Hikvision, Dahua, BriefCam)
Впровадження у Smart City проекти	Юридичні обмеження щодо конфіденційності
Попит на автоматизацію систем безпеки	Можливе підвищення вартості хмарних сервісів
Використання хмарних платформ для масштабування	Кіберзагрози та вимоги до захисту даних
Інтеграція поведінкових моделей (ML/AI)	Залежність від точності нейромережевих алгоритмів
Ринок B2C і домашніх камер	Бар'єр довіри у приватних користувачів

Маркетингова стратегія стартапу ґрунтується на доступності рішення, його практичній цінності та можливості використання з існуючими системами відеоспостереження.

Основними елементами програми є: позиціонування, ціноутворення, канали просування, модель монетизації.

1. Позиціонування продукту

AutoVision Guard позиціонується як:

“Інтелектуальна відеоаналітика для виявлення аномальної поведінки у режимі реального часу, що працює на звичайних камерах та не потребує дорогих рішень”

2. Цінова політика

Для стартапу пропонується гібридна модель монетизації: передплата (SaaS) + встановлення локального серверного рішення.

Таблиця 4.7

Орієнтовна цінова політика стартап-продукту

Модель	Опис	Ціна
SaaS (хмара)	Облік за кількістю камер, хмарна обробка	10–25 €/місяць за одну камеру
Локальний сервер	Встановлення на обладнання клієнта	800–2500 € за інсталяцію
Корпоративний тариф	50+ камер, інтеграція через API	договірна (від 1000 €/міс)
B2C версія	Домашні IP-камери, обробка у хмарі	4.99–7.99 €/міс
Доналаштування/навчання моделей	Під конкретні сценарії	300–2000 € за модель

Канали просування

- **В2В-прямі продажі:** контакти з охоронними агентствами, ТЦ, офісними центрами.
- **Інтеграційні партнери:** компанії, що встановлюють системи відеоспостереження.
- **Онлайн-канали:** сайт, демонстраційне відео, таргетована реклама.
- **Участь у виставках та конференціях безпеки (IFSEC, Intersec).**
- **Freemium-версія:** тест на 3 камери протягом 14 днів.
- **Публікації у технічних блогах, GitHub demo,** що залучає розробників та інтеграторів.

4. Модель масштабування

- Створення маркетингових пакетів для різних сегментів: рітейл, Smart City, приватний сектор.
- Впровадження партнерської програми для інтеграторів систем відеонагляду.

- Масштабування через хмарні платформи (AWS, Google Cloud, Azure).
- Додавання API-модуля для розробників сторонніх рішень.

Висновок

Проведений аналіз у межах стартап-проекту показав, що технологія автоматизованого відеоаналізу, розроблена в практичній частині роботи, має реальні передумови для комерціалізації та масштабування. Сучасний ринок відеоспостереження та інтелектуальної аналітики демонструє стійку потребу в сервісах, які здатні зменшувати вплив людського фактора та забезпечувати оперативне виявлення аномальних ситуацій. Це формує сприятливе середовище для впровадження платформи AutoVision Guard.

Технологічний аудит підтвердив, що обрана архітектура дозволяє масштабувати систему як вертикально (підвищення продуктивності шляхом використання GPU та оптимізованих моделей), так і горизонтально (розподіл потоків між сервісами), що створює можливість виходу на B2B та Smart City сегменти. SWOT-аналіз показав переваги проекту у вигляді модульності, низької вартості впровадження та здатності працювати з існуючими камерами, а також окреслив критичні ризики, які можуть стати обмеженням - конкуренція великих виробників, юридичні вимоги до обробки відеоданих та залежність від точності алгоритмів.

Логічним наслідком аналізу ринку є те, що найбільш перспективними для стартапу є сегменти охоронних агентств, ритейлу, муніципальних служб та підприємств із підвищеними вимогами до безпеки. Саме ці групи отримують найбільшу цінність від автоматизації моніторингу та зниження операційних витрат. Водночас наявність B2C-сегменту створює додатковий потенціал для масштабування та диверсифікації послуг у майбутньому.

ЗАГАЛЬНИЙ ВИСНОВОК ПО РОБОТІ

У межах дослідження було розроблено та практично реалізовано систему автоматизованого аналізу відеопотоків, здатну виявляти аномальну поведінку на основі методів комп'ютерного зору. Проведена теоретична частина дала можливість сформулювати цілісне уявлення про сучасні підходи до детекції об'єктів, трекінгу та поведінкового аналізу, а практична реалізація підтвердила, що навіть на базі доступних моделей та відносно простих алгоритмів можна досягти коректної роботи системи на реальних відеоматеріалах.

Експериментальні результати показали, що запропонований підхід здатний фіксувати підвищену активність, конфліктні ситуації та різкі зміни у поведінці людей. Система адекватно реагує на різні типи відеосцен: від звичайних міських локацій до транспортного трафіку та реальних епізодів із агресією. Зібрана статистика підтвердила, що алгоритм працює стабільно, а використання буферизації дозволило суттєво зменшити кількість хибнопозитивних спрацьовувань.

Разом з тим результати також окреслили природні обмеження камеральної та евристичної аналітики: система чутлива до руху камери, інколи реагує на щільні натовпи та не може повноцінно розпізнавати «м'які» або контекстні сценарії. Однак ці недоліки не зменшують цінності проведеної роботи - навпаки, вони чітко позначають напрямки, у яких варто розвивати систему, зокрема: застосування більш глибоких моделей поведінкового аналізу, покращення трекінгу, інтеграція семантичних моделей та стабілізація відеопотоків.

Стартап-аналіз продемонстрував, що створене рішення має не лише дослідницький інтерес, а й реальну комерційну перспективу. Платформа може бути корисною для різних секторів - від бізнесу до державних

структур - оскільки дозволяє автоматизувати моніторинг і зменшувати навантаження на персонал безпеки. Оцінка ринку, його сегментація та SWOT-аналіз показали, що технологія може знайти своє місце серед існуючих рішень і поступово розширювати функціональність відповідно до потреб користувачів.

У підсумку, робота демонструє, що поєднання теоретичних засад комп'ютерного зору, практичної реалізації алгоритмів та аналізу ринкових умов дає змогу створити повноцінну основу для сучасної системи відеоаналітики. Результатом дослідження стало не лише програмне рішення, а й чітке бачення того, яким чином його можна вдосконалювати та використовувати в реальних умовах - як технічно, так і комерційно.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Goodfellow I., Bengio Y., Courville A. *Deep Learning*. MIT Press, 2016.
<https://www.deeplearningbook.org/>
2. Bishop C. *Pattern Recognition and Machine Learning*. Springer, 2006.
<https://www.springer.com/gp/book/9780387310732>
3. Szeliski R. *Computer Vision: Algorithms and Applications*. Springer, 2022.
<https://szeliski.org/Book/>
4. Gonzalez R., Woods R. *Digital Image Processing*. Pearson, 2018.
<https://www.pearson.com/en-us/subject-catalog/p/digital-image-processing/P200000006612/9780133356724>
5. O'Shea K., Nash R. *An Introduction to Convolutional Neural Networks*. arXiv, 2015. <https://arxiv.org/abs/1511.08458>
6. Krizhevsky A., Sutskever I., Hinton G. *ImageNet Classification with Deep Convolutional Neural Networks*. NeurIPS, 2012.
<https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks>
7. Redmon J., Farhadi A. *YOLOv3: An Incremental Improvement*. arXiv, 2018.
<https://arxiv.org/abs/1804.02767>
8. Tan M., Le Q. *EfficientDet: Scalable and Efficient Object Detection*. CVPR, 2020. <https://arxiv.org/abs/1911.09070>
9. Bewley A. et al. *SORT: Simple Online and Realtime Tracking*. ICIP, 2016. <https://arxiv.org/abs/1602.00763>
10. Wojke N., Bewley A., Paulus D. *DeepSORT: Simple Part-based Tracking with Deep Association Metrics*. 2017.
https://github.com/nwojke/deep_sort

11. Zhang Y. et al. *ByteTrack: Multi-Object Tracking by Associating Every Detection Box*. arXiv, 2021. <https://arxiv.org/abs/2110.06864>
12. Medel J., Savakis A. *Anomaly Detection in Video Using Predictive Convolutional Long Short-Term Memory Networks*. arXiv, 2016. <https://arxiv.org/abs/1612.00390>
13. Hasan M. et al. *Learning Temporal Regularity in Video Sequences*. CVPR, 2016. https://openaccess.thecvf.com/content_cvpr_2016/html/Hasan_Learning_Temporal_Regularity_CVPR_2016_paper.html
14. Tran D. et al. *Learning Spatiotemporal Features with 3D Convolutional Networks*. ICCV, 2015. https://openaccess.thecvf.com/content_iccv_2015/html/Tran_Learning_Spatio-Temporal_Features_ICCV_2015_paper.html
15. ITU-T Recommendation P.1203: *Video Quality Assessment Methods*. <https://www.itu.int/rec/T-REC-P.1203>
16. Cisco Systems. *Cisco Visual Networking Index: Forecast and Trends*. <https://www.cisco.com/>
17. Zhang K., Zuo W., Zhang L. *FFDNet: Toward a Fast and Flexible Solution for CNN-based Image Denoising*. arXiv, 2017. <https://arxiv.org/abs/1710.04026>
18. IEEE. *Real-Time Streaming Protocol (RTSP), RTP and RTCP Standards*. <https://standards.ieee.org/>
19. Vakili V. *WebRTC Architecture & Protocols Overview*. <https://webrtc.org/>
20. Amazon Web Services. *Edge Computing and Real-Time Video Analytics Whitepaper*. <https://aws.amazon.com/whitepapers/>
21. OpenCV Documentation. <https://docs.opencv.org/>
22. PyTorch Documentation. <https://pytorch.org/>

- 23.23. Chollet F. *Deep Learning with Python*. Manning, 2017.
<https://www.manning.com/books/deep-learning-with-python>
24. Karpathy A. *CS231n: Convolutional Neural Networks for Visual Recognition*. Stanford, 2021. <http://cs231n.stanford.edu/>
25. Xu D. et al. *Video Anomaly Detection with Spatiotemporal Autoencoder*. BMVC, 2015. <https://www.bmvc2015.ie/>