ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КАФЕДРА ШТУЧНОГО ІНТЕЛЕКТУ

КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Система відеоспостереження для автоматичного виявлення порушень за допомогою AI-асистента»

зі спеціальності *122 Комп'ютерні науки*
(код, найменування спеціальності)

освітньо-професійної програми *штучний інтелект*
(назва програми)

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело
_____ Марія ПЕРЕСАДА
(підпис)

Виконала: здобувачка вищої освіти групи ШІД-41
Марія ПЕРЕСАДА
(прізвище, ім'я)

Керівник Олександр РАБОТА
(науковий ступінь, вчене звання, прізвище, ім'я)

Рецензент _____
(науковий ступінь, вчене звання, прізвище, ім'я)

Київ 2025

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра Штучного інтелекту
Ступінь вищої освіти Бакалавр
Спеціальність 122 Комп'ютерні науки
Освітньо-професійна програма штучний інтелект

ЗАТВЕРДЖУЮ
Завідувач кафедри ШІ
_____ Ольга ЗІНЧЕНКО
"___" _____ 2025 року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Пересада Марія Денисівна
(прізвище, ім'я)

1. Тема кваліфікаційної роботи: «Система відеоспостереження для автоматичного виявлення порушень за допомогою AI-асистента»

керівник кваліфікаційної роботи Олександр РАБОТА
(прізвище, ім'я, науковий ступінь, вчене звання)
затверджені наказом Державного університету інформаційно-комунікаційних технологій від «24» лютого 2025 року № 56.

2. Строк подання здобувачем вищої освіти кваліфікаційної роботи 23.05.2025 р.

3. Вихідні дані до кваліфікаційної роботи
приклади реалізації систем відеоспостереження з автоматичним виявленням об'єктів та подій;
наукові праці та публікації з тематики застосування штучного інтелекту та машинного навчання у відеоаналітиці

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
1. Аналіз сучасних підходів до побудови систем відеоспостереження та методів автоматичного виявлення подій.
2. Дослідження алгоритмів штучного інтелекту та комп'ютерного зору для розпізнавання порушень у відеопотоці.
3. Розроблення концепції інтеграції AI-асистента для автоматизації процесу виявлення порушень і реагування на них.

4. Перелік графічного матеріалу
Презентація PowerPoint.

5. Дата видачі завдання 25.02.2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Аналіз сучасних систем відеоспостереження та методів автоматичного виявлення об’єктів і подій	21.04.2025 р.	Виконано
2.	Вивчення алгоритмів штучного інтелекту, машинного навчання та комп’ютерного зору для відеоаналітики	31.04.2025 р.	Виконано
3.	Вибір підходів та інструментів для реалізації AI-асистента в системі відеоспостереження	10.05.2025 р.	Виконано
4.	Формування набору даних (відеофрагментів) для навчання та тестування системи	15.05.2025 р.	Виконано
5.	Розроблення системи та навчання моделі виявлення порушень у відеопотоці	26.05.2025 р.	Виконано
6.	Тестування розробленої системи та оцінка результатів	1.06.2025 р.	Виконано
7.	Підготовка звіту та оформлення пояснювальної записки до захисту	10.06.2025 р.	Виконано

Здобувачка вищої освіти _____ Марія ПЕРЕСАДА
(підпис) *(ім’я, прізвище)*

Керівник кваліфікаційної роботи _____ Олександр РАБОТА
(підпис) *(ім’я, прізвище)*

РЕФЕРАТ

Текстова частина бакалаврської роботи: 66 стор., 32 рис., 22 джерела.

Мета роботи – розробити та реалізувати інтелектуальну систему відеоспостереження для автоматичного виявлення порушень за допомогою AI-асистента, забезпечивши ефективність виявлення та обробки подій на основі сучасних технологій штучного інтелекту.

Об'єкт дослідження – процес створення системи відеоспостереження для автоматичного виявлення порушень із використанням штучного інтелекту.

Предмет дослідження – методи комп'ютерного зору, алгоритми машинного навчання для розпізнавання порушень, архітектура побудови AI-асистента для системи відеомоніторингу.

Короткий зміст роботи

1. Аналіз підходів до автоматичного розпізнавання відеопотоків – вивчення методів комп'ютерного зору, CNN, YOLO, Haar Cascade.
2. Формування навчального датасету – підбір та розмітка відеокадрів для тренування моделей розпізнавання.
3. Побудова моделі виявлення порушень – розробка та навчання нейронної мережі з використанням PyTorch та OpenCV.
4. Тестування моделі – оцінка ефективності, precision, recall, F1-score, побудова ROC-кривих та аналіз результатів на практичних прикладах.

Галузь використання – автоматизація систем відеомоніторингу, підвищення ефективності безпеки об'єктів, контроль доступу, розумні міста.

КЛЮЧОВІ СЛОВА: ВІДЕОСПОСТЕРЕЖЕННЯ, АВТОМАТИЧНЕ ВИЯВЛЕННЯ ПОРУШЕНЬ, AI-АСИСТЕНТ, ШТУЧНИЙ ІНТЕЛЕКТ, КОМП'ЮТЕРНИЙ ЗІР, NEURAL NETWORK, CNN, PYTORCH, YOLO, ВІДЕОАНАЛІТИКА.

ABSTRAC

Text part of the bachelor's thesis: 66 pages, 32 figures, 22 sources.

Subject of the study – computer vision methods, machine learning algorithms for violation recognition, and the architecture of an AI assistant for a video monitoring system.

Object of the study – the process of developing a video surveillance system for automatic violation detection using artificial intelligence.

Purpose of the work – to design and implement an intelligent video surveillance system for automatic violation detection using an AI assistant, ensuring efficient event recognition and processing based on modern artificial intelligence technologies.

Summary of the work

1. Analysis of approaches to automatic video stream recognition – study of computer vision methods, CNN, YOLO, Haar Cascade.

2. Formation of a training dataset – selection and annotation of video frames for training recognition models.

3. Construction of a violation detection model – development and training of a neural network using PyTorch and OpenCV.

4. Model testing – evaluation of efficiency, precision, recall, F1-score, building ROC curves, and analyzing results on practical examples.

Field of application – automation of video monitoring systems, enhancement of facility security, access control, smart cities.

KEYWORDS: VIDEO SURVEILLANCE, AUTOMATIC VIOLATION DETECTION, AI ASSISTANT, ARTIFICIAL INTELLIGENCE, COMPUTER VISION, NEURAL NETWORK, CNN, PYTORCH, YOLO, VIDEO ANALYTICS.

ЗМІСТ

ВСТУП

У сучасних умовах зростання рівня загроз для громадської безпеки, об’єктів інфраструктури та приватних територій виникає потреба у вдосконаленні систем відеоспостереження та їх автоматизації. Традиційні системи моніторингу, які ґрунтуються на постійному візуальному контролі з боку операторів, мають низку обмежень, зокрема людський фактор, затримки у реагуванні та значне навантаження при аналізі великого обсягу відеоданих.

Використання AI-асистента в таких системах сприяє своєчасному виявленню подій, що потребують реагування, зменшує ймовірність пропуску критичних моментів та оптимізує роботу персоналу служби безпеки.

Актуальність даного дослідження обумовлена необхідністю створення інтелектуальної системи відеомоніторингу, здатної забезпечити автоматичне виявлення порушень на основі сучасних методів відеоаналітики та штучного інтелекту. Такий підхід дозволяє підвищити рівень безпеки, знизити витрати на обслуговування систем спостереження та забезпечити більш гнучке реагування на потенційні загрози.

Мета роботи – розробити та реалізувати інтелектуальну систему відеоспостереження для автоматичного виявлення порушень за допомогою AI-асистента, забезпечивши ефективність виявлення та обробки подій на основі сучасних технологій штучного інтелекту.

Об’єкт дослідження – процес створення системи відеоспостереження для автоматичного виявлення порушень із використанням штучного інтелекту.

Предмет дослідження – методи комп’ютерного зору, алгоритми машинного навчання для розпізнавання порушень, архітектура побудови AI-асистента для системи відеомоніторингу.

1 АНАЛІЗ СУЧАСНИХ ПІДХОДІВ ДО ПОБУДОВИ СИСТЕМ ВІДЕОСПОСТЕРЕЖЕННЯ

У сучасних умовах зростаючої кількості загроз для об’єктів критичної інфраструктури, підприємств та приватних осіб, системи відеоспостереження відіграють важливу у забезпеченні безпеки та контролю. Ефективне функціонування таких систем дозволяє своєчасно виявляти інциденти, здійснювати моніторинг подій у режимі реального часу та забезпечувати доказову базу для подальшого аналізу [1]. З огляду на стрімкий розвиток цифрових технологій, доцільно дослідити сучасні підходи до проєктування та впровадження систем відеоспостереження, проаналізувати їхні архітектурні особливості, технічні рішення та програмне забезпечення, що використовується для обробки та зберігання відеоданих. Це дозволить визначити найбільш доцільні моделі побудови таких систем з урахуванням актуальних вимог до надійності, масштабованості та захисту інформації.

1.1 Сучасний стан розвитку систем відеоспостереження

Системи відеоспостереження (CCTV – Closed-Circuit Television) пройшли довгий шлях еволюції – від простих аналогових камер до сучасних інтелектуальних рішень з використанням штучного інтелекту та хмарних технологій. Їх розвиток обумовлений постійним зростанням потреби в безпеці, удосконаленням технічних можливостей обробки зображення, збільшенням пропускної здатності мереж та появою новітніх методів зберігання і передачі даних.

Перші системи відеоспостереження з’явилися в середині XX століття та використовувалися переважно для військових і промислових об’єктів. Однією з перших відомих систем була встановлена в Німеччині для спостереження за випробуванням ракет V-2. У 1960–1970-х роках відеоспостереження почали застосовувати в банках, магазинах і на державних підприємствах для базового

контролю безпеки. У цей період використовувалися виключно аналогові камери, що передавали зображення по коаксіальних кабелях на монітори в режимі реального часу без можливості запису (або з використанням магнітофонів на касетах).

Зі зниженням вартості відеокамер та відеомагнітофонів системи відеоспостереження почали активно впроваджуватися в комерційній сфері, торгових центрах, на автостоянках та в громадських місцях. Основною особливістю цього етапу залишалася аналогова передача відео та запис на касетні носії. Якість зображення була обмеженою (часто не перевищувала 480p), а управління камерами здійснювалося вручну [2].

Запровадження цифрових відеореєстраторів (DVR – Digital Video Recorder) дозволило відмовитися від магнітних касет на користь цифрового запису на жорсткі диски, що суттєво покращило якість відео, зручність архівування та пошуку даних. Паралельно розпочалася інтеграція мережевих камер (IP-камер), які передавали відео через комп’ютерні мережі (Ethernet), що забезпечувало більшу гнучкість і масштабованість систем.

Розвиток широкосмугового Інтернету та мережевих технологій призвів до активного поширення IP-відеоспостереження. Камери з підтримкою HD та Full HD якості, функцією нічного бачення, можливістю панорамування, нахилу та масштабування (PTZ-камери) стали стандартом для сучасних систем. Водночас з’явилися мережеві відеореєстратори (NVR – Network Video Recorder), які дозволяли об'єднувати відеодані з кількох камер, забезпечуючи централізоване управління [3].

Системи відеоспостереження почали активно використовувати технології штучного інтелекту (AI) та машинного навчання для автоматичного розпізнавання облич, детекції руху, виявлення підозрілих об’єктів та поведінкової аналітики. Запровадження відеоаналітики дозволило суттєво підвищити ефективність відеоспостереження, мінімізуючи потребу в постійному контролі з боку операторів.

Сучасні системи відеоспостереження дедалі частіше реалізуються у вигляді хмарних сервісів (VSaaS – Video Surveillance as a Service), які дозволяють віддалено керувати камерами, зберігати відео в хмарному сховищі та використовувати аналітичні модулі без потреби в локальному обладнанні. Інтеграція відеоспостереження з іншими елементами систем безпеки (датчиками руху, сигналізацією, IoT-пристроями) забезпечує створення комплексних рішень для моніторингу об’єктів.

Основні тенденції розвитку систем відеоспостереження:

– Перехід від аналогових до цифрових і IP-рішень.

– Застосування штучного інтелекту та відеоаналітики.

– Впровадження хмарних сервісів для гнучкого керування та зберігання даних.

– Інтеграція з IoT-інфраструктурою для розширення можливостей систем безпеки.

– Зростання популярності бездротових та мобільних камер.

Розвиток систем відеоспостереження демонструє поступовий перехід від простих аналогових рішень до складних інтелектуальних екосистем, здатних забезпечити не лише фіксацію подій, а й їх попередження та аналітичну обробку.

Класичними системами відеоспостереження вважаються такі, які не мають вбудованих механізмів автоматичного аналізу відеопотоків за допомогою алгоритмів штучного інтелекту або машинного навчання. Їх основна функція полягає у фіксації та передачі зображення для подальшого перегляду оператором або зберігання на носіях інформації. У таких системах відсутні можливості автоматичного розпізнавання облич, детекції аномалій, поведінкового аналізу тощо.

Основні характеристики класичних систем:

– Використання аналогових або цифрових камер без функцій інтелектуальної обробки, камери лише передають відеосигнал на монітор або реєстратор, не здійснюючи жодної попередньої обробки чи аналізу зображення.

Визначення підозрілих ситуацій та прийняття рішень щодо реагування повністю покладається на оператора, який здійснює візуальний контроль.

– Відсутність автоматизованого аналізу відеозаписів, у процесі роботи не здійснюється автоматичне виявлення або фільтрація подій. Усі записи зберігаються у вигляді суцільного відеопотоку, а перегляд архівних матеріалів проводиться вручну, що потребує значного часу та участі персоналу.

– Наявність базових функцій виявлення руху, у деяких моделях передбачене просте визначення руху на рівні відеореєстратора або самих камер. Однак цей функціонал не є повноцінною відеоаналітикою й не використовує складні алгоритми або нейронні мережі для аналізу ситуацій.

– Локальне зберігання даних, для запису та зберігання відео використовуються локальні носії, здебільшого жорсткі диски у складі DVR (Digital Video Recorder) або NVR (Network Video Recorder), які не підтримують інтелектуальні можливості обробки [4].

– Обмеження у масштабованості та інтеграції з іншими системами, класичні рішення зазвичай мають закриту архітектуру, що створює труднощі при необхідності об’єднання з іншими системами безпеки або інформаційними платформами. Це обмежує їх адаптивність і можливість подальшого розширення.

Переваги класичних систем:

– Простота налаштування та експлуатації.

– Відносно невисока вартість обладнання.

– Надійність та стабільність роботи в базових сценаріях.

– Мінімальні вимоги до пропускної здатності мережі.

Недоліки:

– Відсутність автоматизованого аналізу відео, необхідність постійної участі оператора.

– Складність у швидкому пошуку потрібних фрагментів відео в архіві.

– Низька ефективність при роботі з великим обсягом відеоданих.

– Обмеженість функціоналу для виявлення загроз в режимі реального часу.

Сфера застосування класичних систем:

– Невеликі підприємства, магазини, офіси.

– Парковки, склади, житлові будинки.

– Об’єкти з мінімальними вимогами до аналітики та автоматизації процесів безпеки.

Класичні системи відеоспостереження без штучного інтелекту залишаються актуальними для завдань базового моніторингу, однак у порівнянні з сучасними інтелектуальними рішеннями мають суттєві обмеження у функціональності та ефективності реагування на потенційні загрози [5].

Сучасні системи відеоспостереження активно використовують інформаційні технології для підвищення ефективності та функціональності моніторингу об'єктів. Інтеграція розумних камер, підключених до Інтернету речей (IoT), дозволяє автоматизувати процеси виявлення загроз, зменшити навантаження на операторів та забезпечити більш гнучке управління системами безпеки (рис. 1.1).

Рисунок 1.1 – Структурна схема системи відеоспостереження з інтеграцією IoT-пристроїв

Основні напрямки використання ІТ у відеоспостереженні:

1. Застосування «розумних» відеокамер з вбудованими аналітичними функціями

Сучасні IP-камери обладнані мікропроцесорами, які дозволяють здійснювати попередню обробку відеопотоку безпосередньо на рівні пристрою. Такі камери можуть самостійно виконувати базові аналітичні завдання: виявлення руху, підрахунок людей, визначення зон контролю, а також розпізнавання облич або номерних знаків. Завдяки цьому відпадає необхідність у постійному контролі відеооператором – система самостійно фіксує події, які виходять за межі заданих параметрів.

2. Інтеграція з IoT-пристроями для комплексного моніторингу

Системи відеоспостереження все частіше поєднуються з іншими елементами Інтернету речей – датчиками руху, звуковими сенсорами, тепловізорами, пристроями контролю доступу тощо. Це дає змогу реалізувати єдине інформаційне середовище, де всі компоненти обмінюються даними в режимі реального часу, що підвищує рівень реагування на інциденти.

3. Використання хмарних сервісів та віддаленого доступу

Багато сучасних систем забезпечують можливість зберігання відео в хмарних сховищах та доступу до камер через мобільні додатки або веб-інтерфейси. Це дозволяє здійснювати моніторинг та управління системою з будь-якої точки світу, що особливо зручно для підприємств із розгалуженою інфраструктурою.

4. Підтримка протоколів безпечного з'єднання та стандартизація обміну даними

Системи із підтримкою IoT-архітектури використовують стандартизовані протоколи, такі як MQTT, HTTPS, RTSP та ONVIF, що гарантує безпечну та стабільну взаємодію між різними пристроями незалежно від виробника. Це також полегшує інтеграцію з іншими інформаційними системами підприємства [6].

5. Можливість автоматичного сповіщення та сценарного реагування

Інтелектуальні системи можуть бути налаштовані на автоматичну генерацію сповіщень у разі спрацювання тригерів (детекція вторгнення, зникнення об'єкта, залишені предмети тощо). Сповіщення надходять у вигляді SMS, push-повідомлень або електронних листів, що прискорює реагування відповідальних осіб.

Переваги використання ІТ в системах безпеки:

– Підвищення точності виявлення загроз завдяки аналітичним функціям.

– Автоматизація моніторингу та мінімізація людського фактору.

– Гнучке масштабування та інтеграція з іншими системами.

– Забезпечення віддаленого доступу та управління.

– Оптимізація витрат за рахунок використання хмарних рішень.

Застосування інформаційних технологій, зокрема IoT-підключених пристроїв та розумних камер, дозволяє створювати більш ефективні та адаптивні системи безпеки, здатні своєчасно реагувати на зміну ситуації та знижувати ризики небажаних подій.

1.2 Технології автоматичного виявлення об'єктів та подій

Сучасні системи відеоспостереження все частіше використовують технології автоматичного виявлення об'єктів та подій, що значно підвищує ефективність моніторингу та дозволяє зменшити залежність від постійного контролю з боку людини. Основна ідея таких технологій полягає в автоматизованому аналізі відеопотоку з метою своєчасного виявлення певних ситуацій, які потребують реагування, та подальшої генерації сповіщень або запуску відповідних сценаріїв.

До методів, які застосовуються для автоматичного виявлення об'єктів та подій у відеоспостереженні, належать як класичні алгоритми обробки зображень, так і сучасні підходи комп’ютерного зору та штучного інтелекту. Класичні

алгоритми передбачають використання методів порогової фільтрації, виділення контурів за допомогою алгоритмів Canny або Sobel, фонового віднімання (background subtraction), морфологічних операцій та сегментації зображень. Ці технології дають змогу реалізувати базове виявлення руху, фіксувати зміни в кадрах або визначати присутність об'єктів у контрольованій зоні [7].

Більш складні завдання вирішуються завдяки застосуванню методів комп’ютерного зору, які охоплюють обробку відеопотоків із використанням глибоких згорткових нейронних мереж (CNN), каскадних класифікаторів Haar та методів гістограм орієнтованих градієнтів (HOG). Саме ці підходи дозволяють впроваджувати такі функції, як розпізнавання облич, підрахунок людей, а також класифікацію об’єктів за категоріями, зокрема транспортні засоби, тварини чи люди.

Ще одним напрямом є технології машинного навчання та штучного інтелекту, які базуються на нейронних мережах і алгоритмах глибокого навчання (Deep Learning). Такі системи здатні самостійно навчатися на великих об’ємах даних, що дає змогу суттєво підвищити точність виявлення аномальних або небезпечних ситуацій. Серед найбільш популярних архітектур, які застосовуються в цьому контексті, можна відзначити YOLO (You Only Look Once), SSD (Single Shot Detector), Faster R-CNN та інші. Ці рішення забезпечують високу швидкість обробки відеокадрів та дають можливість точно ідентифікувати об’єкти в реальному часі.

Рисунок 1.2 – Схема методів автоматичного виявлення об'єктів та подій у системах відеоспостереження

Використання описаних методів автоматичного виявлення об'єктів і подій забезпечує реалізацію широкого спектра функцій сучасних систем відеоспостереження. Однією з найпоширеніших задач є визначення появи або переміщення об'єктів у межах контрольованої зони, що дозволяє фіксувати будь-яку активність у потенційно небезпечних або заборонених місцях [8]. Це базова, але водночас ефективна функція, яка широко застосовується для оперативного реагування на вторгнення чи несанкціоноване проникнення (рис. 1.2).

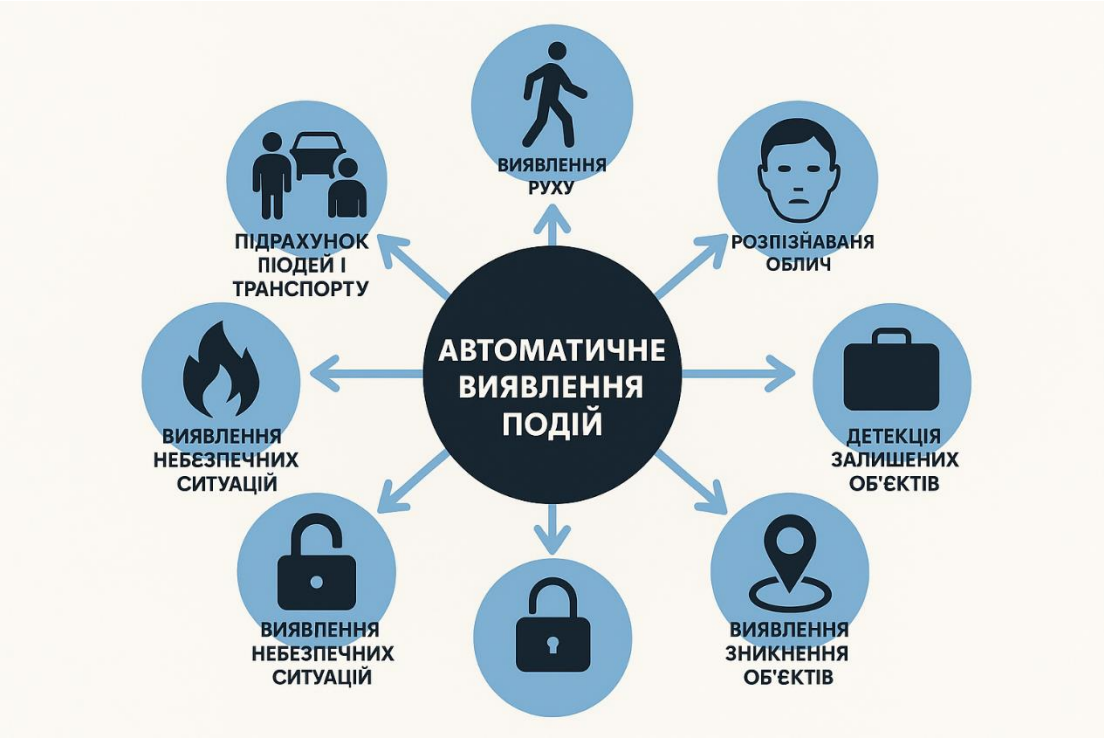

Рисунок 1.3 – Основні задачі, що реалізуються в системах автоматичного виявлення подій

Важливим напрямом є також розпізнавання облич, що реалізується завдяки алгоритмам комп’ютерного зору та глибокого навчання. Такі системи дозволяють ідентифікувати осіб, порівнюючи отримані зображення з базою зареєстрованих користувачів. Це відкриває можливості для автоматизації контролю доступу, пошуку підозрюваних або виявлення порушників.

Серед інших функціональних можливостей варто виділити детекцію залишених або підозрілих об'єктів. У цьому випадку система здатна визначати появу нових предметів на сцені, таких як залишена сумка чи невідомий пакунок, що може свідчити про потенційну загрозу. Аналогічно, система може повідомляти про зникнення об'єктів, наприклад, у разі викрадення обладнання, товарів або інших важливих елементів із контрольованої території [9].

Особливу увагу приділяють автоматичному виявленню небезпечних ситуацій, до яких відносять прояви агресивної поведінки, масові скупчення людей, падіння людини, появу диму або вогню. Завдяки цьому можливо

своєчасно реагувати на інциденти, що становлять загрозу для життя і здоров’я людей чи безпеки об’єкта.

Ще однією актуальною задачею є підрахунок людей та транспортних засобів. Такі функції дозволяють відстежувати кількість осіб або машин, які проходять через певні зони, що є важливим для організації безпеки в громадських місцях, торгових центрах, бізнес-центрах або на парковках. Використання сучасних методів комп’ютерного зору, алгоритмів обробки зображень та технологій штучного інтелекту дозволяє значно підвищити рівень автоматизації систем відеоспостереження. Це забезпечує не лише фіксацію подій, а й їх оперативний аналіз, що істотно покращує якість безпеки, оптимізує витрати на персонал та дозволяє своєчасно реагувати на потенційні загрози.

1.3 Роль штучного інтелекту та комп'ютерного зору в системах відеоаналітики

Серед найбільш поширених алгоритмів, що використовуються для автоматичного виявлення об'єктів у системах відеоспостереження, виділяють згорткові нейронні мережі (CNN), YOLO (You Only Look Once) та каскадні класифікатори Haar.

Згорткові нейронні мережі (CNN) застосовуються для розпізнавання об'єктів та класифікації зображень, зокрема для ідентифікації облич, виявлення транспортних засобів, тварин або інших категорій об'єктів. Вони забезпечують високу точність розпізнавання завдяки багаторівневій обробці та навчальним моделям [9].

Алгоритм YOLO використовується для детекції об'єктів у реальному часі. Він дозволяє швидко та ефективно виявляти об'єкти різних класів на відеокадрах, що робить його актуальним для завдань моніторингу великих територій, виявлення людей, автомобілів або залишених предметів.

Каскадні класифікатори Haar знаходять застосування для базового розпізнавання облич та виявлення певних форм і контурів на зображенні. Цей

метод часто використовується у задачах, де необхідне швидке виявлення без потреби у високих обчислювальних ресурсах.

Застосування цих алгоритмів дозволяє підвищити ефективність систем відеоспостереження, забезпечити точне виявлення об'єктів і своєчасне реагування на потенційні загрози.

У сучасних системах відеоспостереження широке застосування мають алгоритми комп’ютерного зору, що базуються на нейронних мережах та спеціалізованих методах детекції об’єктів. Серед найбільш ефективних підходів використовуються згорткові нейронні мережі (CNN), алгоритм YOLO (You Only Look Once) та каскадні класифікатори Haar (Haar Cascade).

Згорткові нейронні мережі (CNN) є однією з базових технологій у галузі розпізнавання зображень та відео. Завдяки багаторівневій структурі CNN ефективно виділяють ключові ознаки об’єктів, дозволяючи ідентифікувати складні патерни та форми. Ці мережі активно застосовуються для ідентифікації облич, класифікації транспортних засобів, визначення типу об’єктів на відеопотоці, а також в медичних системах для аналізу знімків. У системах відеоспостереження CNN використовуються для розпізнавання осіб на основі біометричних характеристик, виявлення підозрілих предметів та класифікації дій людини.

Алгоритм YOLO (You Only Look Once) є одним із найшвидших і найефективніших методів виявлення об'єктів на зображеннях і відео в реальному часі. Його особливість полягає в тому, що вся сцена аналізується за один прохід через нейронну мережу, що забезпечує високу швидкодію без значних втрат точності. YOLO широко використовується для моніторингу великих територій, виявлення рухомих об'єктів, таких як люди або транспортні засоби, детекції залишених предметів у громадських місцях, а також для підрахунку кількості об'єктів у кадрі [10].

Каскадні класифікатори Haar (Haar Cascade) є класичним підходом для виявлення об'єктів, який базується на використанні наборів простих візуальних ознак, організованих у каскад класифікаторів для швидкої обробки зображення.

Основною перевагою цього методу є низькі вимоги до обчислювальних ресурсів і можливість швидкої роботи навіть на пристроях із обмеженою продуктивністю. Haar Cascade застосовується переважно для розпізнавання облич, виявлення очей, усмішок або інших простих об'єктів, що мають чітко виражені геометричні контури. Цей підхід залишається актуальним для задач, де необхідна базова обробка без використання глибоких нейронних мереж.

Інтеграція CNN, YOLO та Haar Cascade у сучасні системи відеоспостереження дозволяє досягати високої точності виявлення об'єктів, забезпечуючи гнучкість вибору алгоритмів залежно від поставлених завдань, доступних обчислювальних ресурсів і необхідної швидкості реагування. Застосування цих технологій підвищує рівень автоматизації відеомоніторингу, зменшує навантаження на операторів і забезпечує своєчасне виявлення потенційних загроз [11].

Таблиця 1.1 – Приклади використання алгоритмів CNN, YOLO та Haar Cascade у відеоспостереженні

Алгоритм	**Основне призначення**	**Приклади використання**	**Особливості**
CNN (Convolutional Neural Network)	Класифікація та розпізнавання об'єктів	Ідентифікація осіб, розпізнавання номерних знаків, класифікація транспортних засобів, аналіз поведінки	Висока точність, потребує значних обчислювальних ресурсів, адаптивне навчання

Продовження таблиці 2.1

YOLO (You Only Look Once)	Швидке виявлення об'єктів у реальному часі	Виявлення людей, транспортних засобів, залишених речей, підрахунок об'єктів, моніторинг великих площ	Висока швидкодія, ефективність при багатьох об'єктах на сцені, одночасне виявлення кількох класів об'єктів
Haar Cascade	Виявлення об'єктів за геометричними ознаками	Розпізнавання облич, очей, усмішок, базових форм (наприклад, вікон або дверей)	Низькі вимоги до ресурсів, швидка робота, обмежена точність при складних умовах освітлення

1.4 Огляд існуючих рішень та їх порівняльна характеристика

На ринку сучасних систем відеоспостереження представлено низку готових рішень, які відрізняються функціональністю, рівнем інтеграції, можливостями аналітики та масштабованістю[10]. Серед найвідоміших виробників таких систем варто відзначити Hikvision, Dahua Technology, Axis Communications, а також програмні комплекси на основі відкритого програмного забезпечення, зокрема ZoneMinder та Blue Iris [11].

Hikvision є одним із провідних світових виробників систем відеоспостереження. Продукти компанії включають широкий спектр IP-камер, відеореєстраторів та програмного забезпечення для управління відеоспостереженням. Hikvision активно впроваджує технології штучного інтелекту, включаючи розпізнавання облич, детекцію руху, аналітику щодо підрахунку людей і виявлення залишених предметів. Системи підтримують роботу з хмарними сервісами та мають високий рівень інтеграції з іншими безпековими рішеннями (рис. 1.4).

Рисунок 1.4 – Hikvision

Dahua Technology також є одним із лідерів галузі та пропонує широкий вибір камер, NVR та VMS-рішень (Video Management System). Особливістю систем Dahua є застосування власних AI-алгоритмів, зокрема функцій Smart Motion Detection, розпізнавання облич, виявлення перетину лінії та контролю зон. Компанія активно розвиває напрямок інтеграції своїх рішень з IoT-пристроями, що дозволяє створювати більш комплексні системи безпеки (рис. 1.5).

Рисунок 1.5 – Dahua

Axis Communications спеціалізується на IP-камерах та мережевих відеорішеннях високого класу (рис. 1.6). Продукція компанії відома високою

якістю зображення, надійністю роботи та розширеними можливостями відеоаналітики. Axis пропонує відкриті API для розробників, що полегшує інтеграцію з іншими системами безпеки та корпоративною IT-інфраструктурою.

Рисунок 1.6 – Axis

Серед відкритих та програмних рішень варто відзначити ZoneMinder, систему з відкритим вихідним кодом, яка дозволяє організувати базове IP-відеоспостереження. Вона підтримує запис відео, виявлення руху та віддалений доступ, однак поступається комерційним рішенням за функціональністю аналітики [12]. Blue Iris є ще одним популярним програмним продуктом для управління відеоспостереженням, який підтримує широкий перелік камер різних виробників та має функції детекції руху, зонального контролю та автоматичного сповіщення.

Таблиця 1.2 – Порівняльна характеристика популярних систем відеоспостереження

Система	**Тип рішення**	**Підтримка AI/аналітики**	**Інтеграція з IoT**	**Хмарні сервіси**	**Гнучкість налаштування**	**Тип ліцензії**
Hikvision	Апаратура + ПЗ	Розпізнавання облич, підрахунок людей, виявлення вторгнень	Є	Є	Висока	Комерційна

Продовження таблиці 1.2

Dahua Technology	Апаратура + ПЗ	AI-функції Smart Motion, детекція зон, контроль периметра	Є	Є	Висока	Комерційна
Axis Communications	IP-камери + ПЗ	Відеоаналітика, API для розробників	Є	Є	Висока	Комерційна
ZoneMinder	Програмне забезпечення	Базове виявлення руху	Немає	Немає	Середня	Відкрите ПЗ
Blue Iris	Програмне забезпечення	Детекція руху, зони контролю	Частково	Частково	Висока	Комерційна

На основі проведеного аналізу сучасних систем відеоспостереження, технологій автоматичного виявлення подій та застосування алгоритмів штучного інтелекту, можна зробити висновок про доцільність створення інтелектуальної системи, здатної самостійно аналізувати відеопотік та виявляти порушення. Визначені переваги та недоліки існуючих рішень слугували підґрунтям для формування вимог до власної розробки системи відеоспостереження з AI-асистентом.

У наступному розділі буде сформульовано мету та завдання створення системи, визначено функціональні й нефункціональні вимоги, критерії ефективності, а також буде обґрунтовано вибір архітектурного підходу та інструментів реалізації. Це дозволить закласти фундамент для практичної реалізації інноваційного рішення у сфері відеоаналітики.

Висновок до розділу 1

Розглянуто етапи розвитку систем відеоспостереження, що відображають еволюцію від простих аналогових рішень до сучасних інтелектуальних комплексів з використанням штучного інтелекту та IoT-технологій. Визначено,

що сучасні системи відеоспостереження поєднують функції не лише фіксації подій, а й автоматичного виявлення об'єктів, розпізнавання осіб, виявлення аномальних ситуацій і прогнозування потенційних загроз.

Проаналізовано основні методи автоматичного виявлення об'єктів і подій, серед яких виділено класичні алгоритми обробки зображень, методи комп’ютерного зору, машинного навчання та глибокого навчання. Окреслено ключові задачі відеоаналітики, зокрема виявлення руху, розпізнавання облич, детекцію залишених об’єктів, виявлення небезпечних ситуацій та підрахунок людей чи транспортних засобів.

Особливу увагу приділено ролі штучного інтелекту, який суттєво підвищує ефективність систем відеоспостереження за рахунок автоматизації процесів обробки відеопотоку та зменшення впливу людського фактора. Розглянуто приклади використання алгоритмів CNN, YOLO та Haar Cascade, що дозволяють забезпечити точне й швидке виявлення об’єктів та подій.

Проведено огляд найбільш поширених рішень на ринку відеоспостереження, таких як Hikvision, Dahua, Axis Communications, ZoneMinder та Blue Iris, із визначенням їх основних характеристик та можливостей. Сформована порівняльна характеристика дає змогу оцінити переваги та обмеження різних підходів до побудови систем відеомоніторингу, що є підґрунтям для вибору оптимального рішення залежно від конкретних завдань та умов експлуатації.

2 ПОСТАНОВКА ЗАДАЧІ ТА ВИМОГИ ДО СИСТЕМИ

2.1 Методи та моделі штучного інтелекту

У цьому підрозділі описуються основні алгоритми та підходи штучного інтелекту, які можуть бути інтегровані для автоматизації, оптимізації та прогнозування у бездротових мережах високої пропускної здатності.

Одним із важливих напрямків використання ШІ в таких системах є динамічна оптимізація радіочастотного планування. Штучний інтелект може в реальному часі аналізувати параметри, як-от RSSI, SNR та завантаженість каналів, що дає змогу адаптивно підбирати частоти для мінімізації інтерференцій і уникнення утворення «мертвих зон» покриття. Це особливо критично для камер відеоспостереження з бездротовим підключенням, які потребують стабільного зв’язку.

Іншим ключовим аспектом є інтелектуальне керування радіоресурсами. Використання підходів з підкріплювальним навчанням (Reinforcement Learning) дозволяє балансувати навантаження між точками доступу та клієнтськими пристроями, що забезпечує рівномірний розподіл ресурсів, максимізацію пропускної здатності мережі та зменшення затримок передачі відеопотоків.

Прогнозування трафіку на основі моделей, таких як LSTM, дає змогу передбачити періоди пікового навантаження (наприклад, у години великого скупчення людей) і завчасно переналаштувати параметри якості обслуговування (QoS), аби уникнути перевантажень. Це підвищує стабільність відеопотоку та загальну якість обслуговування користувачів або операторів.

Завдяки ШІ система може самостійно виявляти аномалії та потенційні загрози. Наприклад, використання автоенкодерів або методів Isolation Forest дозволяє діагностувати нетипову поведінку в мережі — сплески автентифікаційних помилок, DDoS-атаки або збої в роботі обладнання. При виявленні таких ситуацій можливе автоматичне ініціювання процедур самовідновлення (self-healing), що значно скорочує час реагування на інциденти.

Важливим елементом є й розгортання edge-обчислень за допомогою

TinyML — легковагових моделей, які можна запускати безпосередньо на периферійних пристроях, таких як точки доступу чи шлюзи. Це дозволяє локально фільтрувати шум, агрегувати дані та зменшити потребу в передачі великих обсягів інформації до хмари.

На завершення, для підвищення конфіденційності даних і зниження навантаження на мережу застосовується підхід федеративного навчання. У цьому випадку моделі навчаються безпосередньо на пристроях, і лише оновлення ваг передаються в центральну систему, без пересилання сирих відеоданих або логів. Це важливо в контексті дотримання норм захисту персональних даних.

У сукупності ці напрямки створюють високоефективну, адаптивну й безпечну систему відеоспостереження з підтримкою ШІ, яка здатна не лише фіксувати порушення, а й активно реагувати на зміну ситуації в реальному часі.

2.2 Класичні алгоритми машинного навчання

Класичні алгоритми машинного навчання, а також сучасні методи глибокого навчання відіграють ключову роль у забезпеченні ефективного функціонування систем відеоспостереження з підтримкою ШІ. Одним із базових підходів є кластеризація, наприклад, за допомогою алгоритмів K-Means або DBSCAN. Такі алгоритми дозволяють автоматично групувати зони спостереження за подібними характеристиками — наприклад, за рівнем завантаження або рівнем інтерференції. Ці методи мають низькі обчислювальні вимоги й швидко сходяться до результату, однак їх застосування потребує попереднього налаштування, зокрема задання кількості кластерів чи радіусу сусідства.

Також ефективними є дерева рішень та ансамблеві методи, як-от Random Forest або XGBoost. Вони здатні прогнозувати пропускну здатність, вірогідність виникнення інтерференції або ризик відмов обладнання, спираючись на набір мережевих ознак. Такі моделі автоматично визначають найбільш значущі ознаки, демонструючи високу точність, але водночас можуть бути схильними до

перенавчання та потребують обережного налаштування, особливо при великих ансамблях.

У сфері глибокого навчання особливе місце займають згорткові нейронні мережі (CNN), які застосовуються для аналізу карт радіопокриття — наприклад, для виявлення «мертвих зон» чи зон з високим рівнем завад. Такі карти подаються у вигляді зображень, що дозволяє моделі ефективно витягати просторові патерни. Водночас, CNN мають високі вимоги до ресурсів, зокрема потребують графічних процесорів для тренування.

Для роботи з часовими залежностями застосовуються рекурентні мережі (RNN) та їхня вдосконалена версія — LSTM. Ці моделі здатні прогнозувати періоди пікового навантаження на основі історичних даних про throughput, затримки або кількість підключень. Вони беруть до уваги довготривалі залежності у трафіку, хоча їх ефективність може знижуватись через ефект «затухаючого градієнта» й високу потребу в великих наборах даних для навчання.

Глибокі автоенкодери застосовуються для виявлення аномалій на основі високої помилки реконструкції. Вони не потребують попереднього маркування даних і працюють у режимі без учителя, що дозволяє їм ідентифікувати нетипові умови в системі. Проте, їхні результати важко інтерпретувати, і для стабільної роботи потрібно ретельне калібрування.

Ще одним перспективним напрямком є застосування підходів підкріплювального навчання (Reinforcement Learning) для керування мережевими параметрами. У цьому контексті агентом виступає контролер мережі, який оцінює поточний стан (метрики завантаження, RSSI, кількість клієнтів) і приймає рішення щодо змін параметрів, таких як радіоканал, потужність чи QoS. Отримана нагорода враховує throughput, затримку та споживання енергії. У невеликих середовищах ефективним є Q-Learning, який працює з дискретними станами, тоді як Deep Q-Network використовує нейронні мережі для апроксимації функції корисності у великих просторах. Підходи Policy Gradient і Actor–Critic дозволяють працювати з безперервними просторами дій і мають стабільну збіжність, що робить їх придатними для складних сценаріїв.

Суттєву роль у сучасних рішеннях відіграють edge-обчислення, які дозволяють проводити попередню обробку даних безпосередньо на пристроях — точках доступу або edge-шлюзах. Це значно зменшує затримки та трафік у мережі. Завдяки TinyML можна реалізовувати квантизовані моделі на мікроконтролерах з мінімальними ресурсами. Наприклад, за допомогою фреймворків TensorFlow Lite for Microcontrollers чи Arm CMSIS-NN можна запускати легковагові CNN або LSTM, які виявляють аномалії або приймають рішення на місці, без потреби у хмарній обробці.

Щодо захисту даних і ефективного розподіленого навчання, провідне місце займає федеративне навчання. Його архітектура передбачає, що локальні пристрої (наприклад, AP або IoT-сенсори) тренують модель на власних даних, а сервер виконує лише агрегацію оновлених ваг. Такий підхід забезпечує конфіденційність, оскільки сирі дані не покидають пристрої, а також зменшує навантаження на мережу. Водночас існують обмеження — наприклад, асинхронність оновлень клієнтів або неоднорідність (non-IID) розподілу даних, що ускладнює узагальнення моделі. Крім того, для захисту від потенційних атак потрібне застосування додаткових механізмів, як-от Secure Aggregation або Differential Privacy.

У підсумку, поєднання класичних методів машинного навчання, глибокого навчання, підкріплювального навчання та edge-технологій забезпечує гнучкість, ефективність і безпечність інтелектуальних систем відеоспостереження нового покоління.

2.3 Функціональні та нефункціональні вимоги до системи

З метою забезпечення узгодженості функціональних компонентів системи відеоспостереження з AI-асистентом доцільно представити її загальну архітектуру у вигляді структурної схеми. Такий підхід дозволяє візуалізувати основні елементи системи, їхню взаємодію та послідовність обробки інформації – від отримання відеопотоку до відображення результатів аналізу користувачеві. На

рисунку 2.1 подано спрощену модель, яка охоплює камери спостереження, модуль обробки відеоданих із вбудованими алгоритмами штучного інтелекту, базу даних для зберігання інформації та інтерфейс користувача для взаємодії із системою.

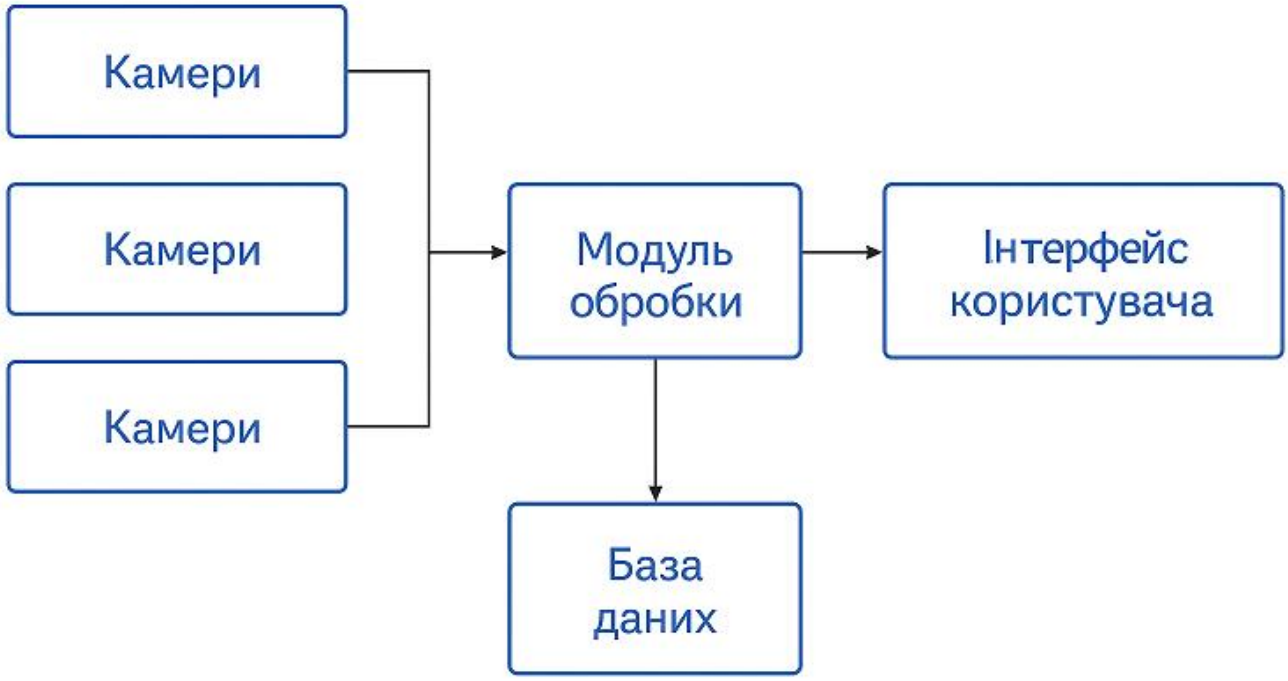

Рисунок 2.1 – Загальна структура системи відеоспостереження з AI-асистентом

Функціональні вимоги описують можливості та очікувану поведінку системи відеоспостереження з використанням ШІ. Система повинна забезпечувати безперервний відеомоніторинг із декількох камер у режимі реального часу. Вона має автоматично виявляти рух і класифікувати об’єкти, зокрема людей, транспортні засоби або залишені предмети. Передбачено виявлення типових порушень, таких як проникнення в заборонену зону, залишений предмет, скупчення людей чи падіння людини. За потреби система повинна мати можливість ідентифікувати обличчя та звіряти їх із наявною базою даних. Усі події фіксуються з можливістю запису відповідних відеофрагментів у локальне або хмарне сховище. При виявленні потенційних загроз система автоматично генерує сповіщення — через email, push-повідомлення або інтерфейс. Користувач має доступ до інтерфейсу, що дозволяє переглядати

активні події, працювати з архівом, керувати налаштуваннями та отримувати аналітичні звіти. Також підтримується експорт журналів подій для подальшого аналізу або звітності.

Нефункціональні вимоги визначають якісні характеристики системи. Важливою є висока швидкодія — затримка обробки відеокадру не повинна перевищувати однієї секунди. Система має бути масштабованою, тобто здатною до розширення — додавання нових камер або функціональних модулів не повинно потребувати значної перебудови архітектури. Надійність означає стійкість до втрати мережевого з'єднання: обробка подій має відбуватись автономно у разі тимчасової недоступності мережі. Особливу увагу слід приділити інформаційній безпеці — відеодані мають бути захищені як під час зберігання, так і при передаванні, наприклад, за допомогою HTTPS та шифрування. Система повинна бути кросплатформенною й сумісною з основними браузерами та мобільними пристроями, забезпечуючи зручний доступ до інтерфейсу. Простота використання є критично важливою — інтерфейс повинен бути інтуїтивно зрозумілим і не вимагати тривалого навчання. Висока точність виявлення подій також є ключовою вимогою: система має забезпечувати рівень F1-score не менше 0.85 на тестових наборах. Окрім цього, вона повинна бути адаптивною до змін навколишнього середовища, функціонуючи як удень, так і вночі, а також за різних погодних умов чи рівнів освітлення.

2.4 Визначення критеріїв ефективності та якості роботи системи

Для об’єктивної оцінки результатів роботи системи відеоспостереження з AI-асистентом необхідно визначити низку кількісних та якісних критеріїв, які дозволяють оцінити її ефективність, точність та продуктивність. Такі критерії є важливими на етапах тестування, вдосконалення та порівняння з альтернативними рішеннями.

У сфері комп’ютерного зору, зокрема при класифікації та виявленні об’єктів у відеоаналітичних системах, широко застосовуються низка ключових метрик

якості. Однією з основних є точність (precision), яка показує, яка частка виявлених системою порушень дійсно була порушеннями. Високе значення precision свідчить про низький рівень хибнопозитивних спрацювань, тобто система рідко помилково ідентифікує нешкідливі події як загрозливі. Повнота (recall), своєю чергою, демонструє здатність системи виявляти реальні інциденти, тобто частку подій, які система не пропустила. Високий recall означає мінімальну кількість хибнонегативних спрацювань, що критично важливо для безпеки.

Збалансовану оцінку ефективності забезпечує F1-score — гармонійне середнє між precision і recall. Цей показник дозволяє оцінити, наскільки добре система виконує свої завдання, враховуючи як помилки першого, так і другого роду. У практичних застосуваннях відеоаналітики значення F1-score не менше 0.85 вважається достатнім для ефективного використання системи.

Окрім метрик точності, важливу роль відіграють технічні параметри продуктивності. Одним з критичних показників є час обробки кадру (latency) — тобто, скільки часу потрібно системі для аналізу одного відеокадру та прийняття рішення. У режимі реального часу цей показник повинен залишатися в межах до 1 секунди, а в ідеалі — не перевищувати 300 мілісекунд. Також має значення ефективність використання апаратних ресурсів: система повинна раціонально працювати з доступним процесором, GPU та оперативною пам’яттю, не створюючи перевантаження, особливо в умовах, коли застосовується розгортання на edge-пристроях з обмеженими можливостями.

Ще одним важливим аспектом є стійкість до навантажень, тобто здатність системи підтримувати стабільну роботу за умови обробки великої кількості відеопотоків одночасно або за високої частоти подій. Нарешті, показник рівня хибнопозитивних та хибнонегативних спрацювань безпосередньо впливає на довіру користувачів до системи. У стабільних системах відеоаналітики частка таких спрацювань повинна залишатися в межах прийнятного рівня, зазвичай не більше 10%.

Усі зазначені критерії мають бути враховані на етапі тестування системи, а також у процесі її вдосконалення. Їх використання забезпечує прозорість оцінки та дозволяє робити обґрунтовані висновки щодо якості розробленого рішення.

Для формалізації очікуваної поведінки та властивостей розроблюваної системи відеоспостереження доцільно чітко окреслити перелік функціональних і нефункціональних вимог. Функціональні вимоги визначають основні завдання, які повинна виконувати система, зокрема виявлення та класифікацію порушень, взаємодію з користувачем, фіксацію подій і реагування на них. Нефункціональні вимоги, своєю чергою, описують якісні характеристики системи, такі як продуктивність, масштабованість, надійність, безпека та зручність використання. У табл. 2.1 наведено узагальнений перелік ключових вимог до системи, які було враховано на етапі проєктування та реалізації.

Таблиця 2.1 – Критерії ефективності та якості роботи системи

Критерій / Метрика	**Опис / Пояснення**	**Оптимальні значення**
Precision (точність)	Частка правильно виявлених порушень серед усіх виявлених системою випадків.	≥ 0.85
Recall (повнота)	Частка реальних порушень, які були успішно виявлені системою.	≥ 0.85
F1-score	Середнє гармонійне між precision та recall; показник загальної точності.	≥ 0.85
Час обробки кадру	Максимальний допустимий час обробки одного кадру в реальному часі (≤ 1 с).	≤ 1 с (оптимально < 300 мс)
Використання апаратних ресурсів	Раціональне використання CPU, GPU та оперативної пам’яті.	Менше 80% використання основних ресурсів
Стійкість до навантажень	Стабільна робота при обробці великого потоку даних або паралельному аналізі.	Без збоїв при ≥ 4 потоках одночасно
Рівень хибнопозитивних спрацювань	Частка помилкових спрацювань серед усіх позитивних рішень системи.	< 10%
Рівень хибнонегативних спрацювань	Частка пропущених реальних порушень серед усіх порушень.	< 10%

2.5 Обґрунтування вибору архітектури та інструментів для реалізації

Вибір архітектури програмної системи та інструментів її реалізації має вирішальне значення для досягнення цілей проєкту та забезпечення його ефективного функціонування. З огляду на поставлені вимоги до системи відеоспостереження з AI-асистентом, обґрунтованим є застосування клієнт-серверної архітектури з чітким розподілом функціональних модулів, що дозволяє забезпечити масштабованість, гнучкість розгортання та простоту супроводу.

Запропонована архітектура системи базується на принципах модульності та розподіленої обробки даних. Вона передбачає розділення функціоналу між окремими компонентами. Клієнтська частина реалізована у вигляді веб-додатку, що слугує інтерфейсом для оператора. Через нього користувач має змогу переглядати відеопотоки, отримувати сповіщення, працювати з архівом подій і керувати системою. Серверна частина відповідає за реалізацію логіки взаємодії: вона обробляє запити, звертається до бази даних, приймає результати з AI-модуля та надсилає відповіді назад клієнту. Сам AI-модуль є автономною підсистемою, що виконує аналіз відеопотоку, виявляє об’єкти та класифікує події. Усі події, лог-файли й налаштування користувача зберігаються в централізованій базі даних.

За необхідності архітектура може бути трансформована у мікросервісну. У такому випадку окремі сервіси, зокрема модулі розпізнавання облич або аналітики подій, працюють незалежно один від одного. Це дає змогу гнучко масштабувати функціональність системи, поліпшити керованість та спростити процес тестування кожного компонента.

Для реалізації системи було використано сучасний стек технологій. Основною мовою програмування є Python, яка забезпечує гнучкість і має велику кількість бібліотек для машинного навчання та обробки зображень. Для реалізації моделей нейронних мереж обрано фреймворк PyTorch, що відзначається високою точністю, швидкою розробкою та підтримкою динамічних обчислень, що є критично важливим для обробки відео в реальному часі. Обробка кадрів на

попередньому етапі здійснюється за допомогою бібліотеки OpenCV, що дозволяє зменшувати шум, виділяти контури та здійснювати трекінг об’єктів.

Серверну частину реалізовано з використанням фреймворку Django, який забезпечує швидке прототипування, надійний захист, підтримку ORM і широке ком’юніті. Контейнери Docker використовуються для ізоляції компонентів системи, що суттєво полегшує розгортання, масштабування та тестування без необхідності змінювати конфігурації в різних середовищах. У якості системи керування базами даних застосовується PostgreSQL, яка забезпечує стабільне зберігання інформації про події, користувачів та параметри системи.

Клієнтська частина побудована за допомогою мов JavaScript, HTML і CSS, що забезпечує інтерактивність та зручність інтерфейсу для оператора. Такий підхід дозволяє створити ефективну, масштабовану та легко підтримувану систему відеоаналітики.

Завдяки такому вибору інструментів можливо створити адаптивну, надійну та зручну у використанні систему відеоспостереження, яка може бути легко розширена або інтегрована з іншими рішеннями у сфері безпеки.

Для наочного розуміння структури та взаємодії компонентів системи відеоспостереження з AI-асистентом доцільно представити її архітектуру у вигляді блок-схеми. Схема ілюструє послідовність обробки даних: відеопотік надходить від камер до модуля обробки, де задіюються алгоритми штучного інтелекту для виявлення порушень. Оброблені дані зберігаються в базі даних, а результати аналітики передаються до інтерфейсу користувача, що дозволяє оперативно реагувати на інциденти. Такий архітектурний підхід забезпечує гнучкість, масштабованість і можливість подальшого вдосконалення системи.

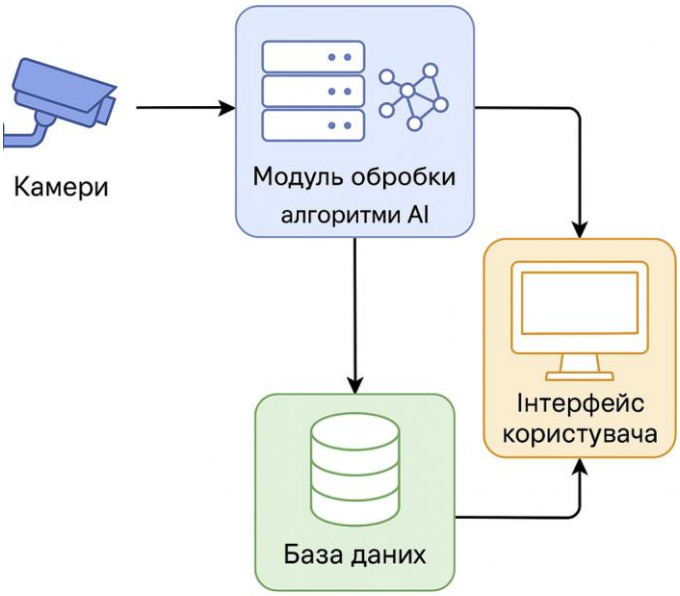

Рисунок 2.2 – Архітектура системи відеоспостереження з AI-модулем

Обґрунтований вибір клієнт-серверної архітектури, а також використання сучасного та перевіреного технологічного стека забезпечує надійність, масштабованість і гнучкість системи відеоспостереження з AI-асистентом. Такий підхід дозволяє ефективно розподілити навантаження між компонентами, спростити розгортання і супровід, а також забезпечити високий рівень адаптивності до змін вимог або інтеграції з іншими системами. Використання інструментів, зокрема Python, PyTorch, OpenCV, Django, Docker і PostgreSQL, дозволяє досягти високої якості реалізації інтелектуальних функцій аналізу відеопотоку в реальному часі. У цілому, обрана архітектура та набір інструментів забезпечують реалізацію всіх функціональних вимог проєкту з урахуванням перспектив подальшого розвитку.

Висновок до розділу 2

У процесі розробки було визначено ключові аспекти, що формують концептуальну основу інтелектуальної системи відеоспостереження з AI-

асистентом. Сформульовано мету та завдання проєкту, які охоплюють виявлення подій, обробку відеопотоку, створення й тестування моделей штучного інтелекту.

Встановлено функціональні та нефункціональні вимоги до системи, що забезпечують її ефективність, надійність і масштабованість. Значну увагу приділено показникам якості, зокрема метрикам точності виявлення (precision, recall, F1-score), а також технічним характеристикам – швидкості обробки кадру й стабільності роботи.

Архітектура проєкту побудована за клієнт-серверною моделлю з потенціалом для переходу до мікросервісного підходу. Обґрунтовано вибір сучасного технологічного стеку, зокрема PyTorch, OpenCV, Django, Docker, які забезпечують гнучку та надійну реалізацію. Усі ці рішення заклали основу для переходу до етапу практичного впровадження системи.

3 ЗАСОБИ ТА ІНСТРУМЕНТИ РОЗРОБКИ СИСТЕМИ

3.1 Використані інструменти, мови програмування та середовище розробки

Для створення інтелектуального веб-застосунку було використано сучасні інструменти та технології, які забезпечують ефективну розробку, тестування та розгортання системи.

Основним середовищем розробки обрано Visual Studio Code, що поєднує функціональність IDE з легкістю текстового редактора. Завдяки підтримці великої кількості розширень, підсвічуванню синтаксису, автодоповненню та налагодженню, забезпечується зручна розробка на JavaScript, Python, HTML/CSS та інших мовах.

Python використовується як основна мова для реалізації обчислювальної логіки та моделей штучного інтелекту. Завдяки великій кількості бібліотек, таких як NumPy, SciPy, Scikit-learn, TensorFlow та PyTorch, Python є де-факто стандартом у сфері Data Science та машинного навчання.

JavaScript застосовується для реалізації інтерактивної частини веб-інтерфейсу. Він дозволяє реалізовувати реакцію на дії користувача, асинхронні запити до сервера (через AJAX або Fetch API), а також підтримує фреймворки на кшталт React, які прискорюють розробку UI-компонентів.

HTML5 та CSS3 забезпечують структуру сторінок і візуальне оформлення інтерфейсу відповідно. HTML5 забезпечує семантичне розмічування контенту, а CSS3 — адаптивність, анімації та кастомізацію інтерфейсу [13].

Для реалізації серверної частини обрано Django — потужний веб-фреймворк на Python, що слідує архітектурі Model–Template–View. Django автоматизує рутинні завдання, забезпечує безпечну автентифікацію, ORM для взаємодії з базами даних та гнучку маршрутизацію запитів [14].

Як систему керування базами даних використано PostgreSQL — потужну, масштабовану об'єктно-реляційну СУБД з відкритим вихідним кодом. Вона

забезпечує підтримку транзакцій, складних запитів, реплікації та повнотекстового пошуку, що критично важливо для надійного зберігання й обробки даних .

Інфраструктура проєкту контейнеризована за допомогою Docker Compose. Це дає змогу легко керувати сервісами (базою даних, сервером, фронтендом) через YAML-файл та автоматично створювати середовище для розгортання застосунку.

Як веб-сервер застосовується NGINX — високопродуктивний реверсивний проксі, що забезпечує балансування навантаження, кешування та підтримку безпечного протоколу HTTPS [15].

Для обробки асинхронних запитів у Python-частині використано Hypercorn — ASGI-сервер, що підтримує сучасні протоколи (HTTP/2, WebSocket), високопродуктивний і легко інтегрується з Django [16].

Система ідентифікації та трекінгу об’єктів (рис. 3.1) складається з послідовно пов’язаних функціональних модулів, кожен з яких виконує окрему задачу в загальному алгоритмі обробки відеопотоку. Загальна логіка роботи системи передбачає такі етапи:

1) Попередня обробка вхідного відео з подальшим розбиттям на окремі кадри;

2) Сегментація зображення кожного кадру з метою виявлення областей, де потенційно присутні номерні знаки [12];

3) Застосування алгоритмів оптичного розпізнавання символів для ідентифікації номерного знаку в межах виділених фрагментів [13];

4) Візуалізація розпізнаного номерного знаку безпосередньо на кадрі, що обробляється;

5) Формування нового відеопотоку з урахуванням відображених результатів розпізнавання;

6) Аналіз і фіксація отриманих даних із кожного відеопотоку для подальшого збереження або передавання;

7) Формування кінцевих результатів ідентифікації та трекінгу для представлення в системі звітності або керування [16].

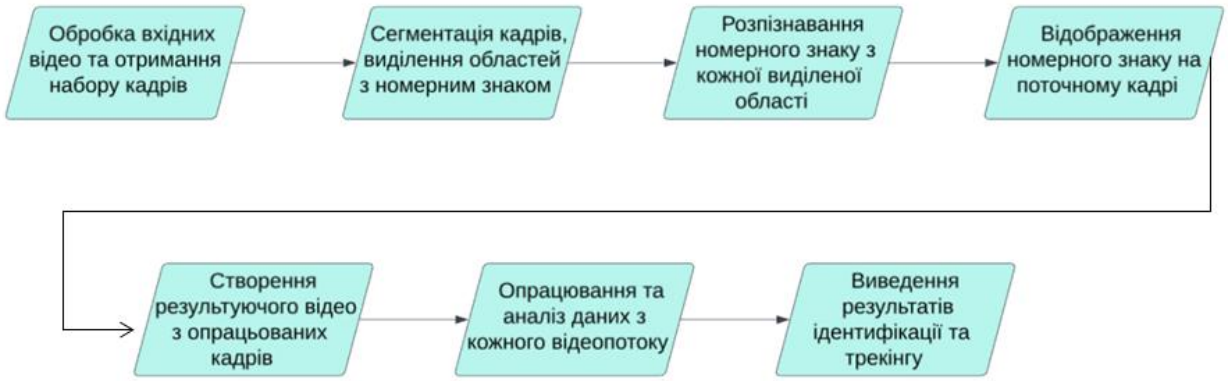

Рисунок 3.1 – Логічна модель функціонування системи розпізнавання номерних знаків

Модуль виявлення номерних знаків на основі YOLO

Для реалізації виявлення номерного знака в системі ідентифікації використано сучасний алгоритм YOLO (You Only Look Once), що належить до категорії нейронних моделей реального часу. Цей підхід дозволяє ефективно виявляти об’єкти на зображенні за один прохід нейронною мережею, поєднуючи класифікацію та локалізацію. YOLO поділяє зображення на сітку розміром S×S і прогнозує, чи розташований центр об'єкта в межах кожної з комірок. Для кожної з них одночасно визначається обмежувальне поле та ймовірність класу.

Перевагами YOLO є надзвичайна швидкість роботи, здатність аналізувати повне зображення одразу (враховуючи контекст), а також узагальнення об’єктів навіть на нових або змінених зображеннях. Разом із цим, модель має низку обмежень: максимальна кількість виявлених об’єктів обмежується кількістю комірок сітки; важко коректно обробити ситуації з кількома об’єктами в одній комірці; а також можливе дублювання результатів при частковому перекритті.

Архітектура YOLO включає згорткові прошарки, зменшувальні 1×1 шари та повнозв’язні шари на виході, що забезпечують прогноз координат обмежувального поля ((x, y), w, h), де x та y – це координати центра об’єкта в межах комірки, а ширина та висота задаються відносно всього зображення. Для відсіву помилкових виявлень застосовується порогове значення довіри: модель

виводить тільки ті обмежувальні поля, значення яких перевищує визначений рівень (наприклад, 0.5).

Модуль трекінгу ідентифікованих об’єктів

Після виявлення об'єктів система переходить до етапу трекінгу – відстеження об'єктів у просторі та часі. У даному проєкті застосовано логіку об'єднання ідентифікованих номерних знаків на основі їхньої появи у межах певного часово-просторового вікна. Якщо один і той самий номерний знак зафіксовано на кількох кадрах або з різних камер із незначною часовою різницею, система розцінює це як наявність одного і того ж транспортного засобу в полі зору.

Для підвищення надійності трекінгу впроваджено алгоритм текстової схожості – він дозволяє згрупувати навіть неповні чи частково нечіткі знаки (наприклад, "AA2345E" і "AA2345EK") в межах одного об’єкта. Завдяки цьому забезпечується стійке відстеження об’єктів навіть у складних умовах, зокрема за наявності часткових перекриттів, тимчасового зникнення з кадру або зміни кута огляду [17-18].

3.2 Архітектура веб-системи відеоспостереження

Взаємодія користувача із системою побудована за класичною схемою запит–відповідь (request–response), що є характерною для більшості вебзастосунків. Клієнт надсилає HTTP-запит, який обробляється серверною частиною, після чого формується відповідь і повертається у вигляді вебсторінки або даних.

Серверна частина застосунку реалізована з використанням архітектури MVT (Model–View–Template), що лежить в основі фреймворку Django. Такий підхід дозволяє чітко розмежувати логіку обробки даних, їх візуальне представлення та структуру веб-сторінок. Компонент Model відповідає за збереження й обробку структурованих даних у базі (наприклад, у PostgreSQL), View виконує логіку обробки запитів і формує відповіді на основі бізнес-правил, а

Template відповідає за формування кінцевої HTML-сторінки на основі даних, отриманих через View.

Запит користувача, що надходить через браузер, спочатку потрапляє на веб-сервер NGINX, який виконує маршрутизацію до доступного екземпляра застосунку, запущеного під керуванням сервера Hypercorn. Далі застосунок зіставляє URL запиту з відповідним компонентом View. За потреби View звертається до моделі для доступу до даних, після чого отримана інформація інтегрується у шаблон, що формує фінальну HTML-сторінку. Ця сторінка повертається користувачеві через веб-сервер як відповідь на його запит. Така архітектура забезпечує гнучкість, зрозумілу логіку обробки запитів та зручне масштабування.

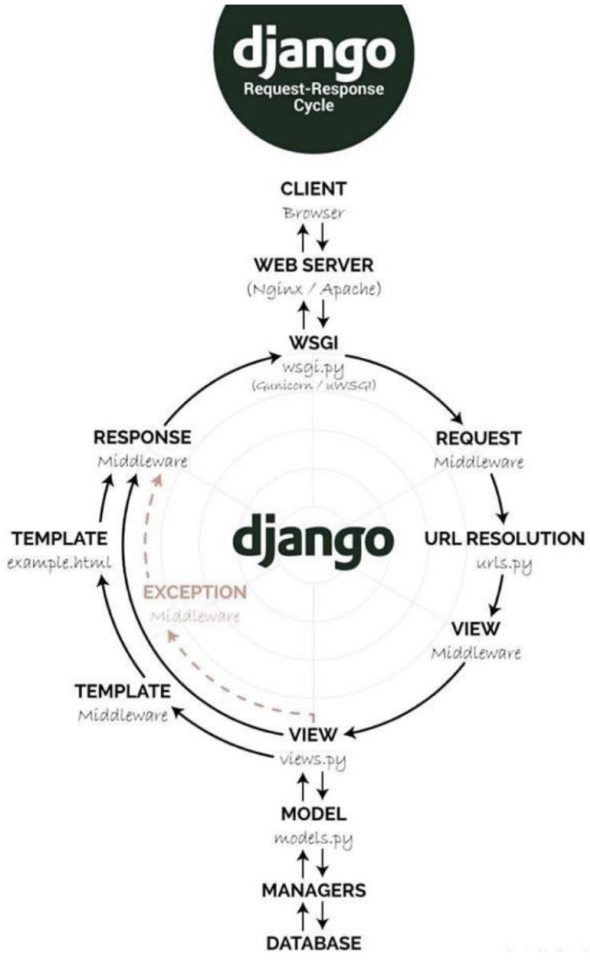

Рисунок 3.2 – Принцип роботи архітектури MVT у веб-додатку

Для наочного представлення способів взаємодії користувачів із системою було побудовано діаграму прецедентів (рис. 3.3), яка відображає основні функціональні можливості, доступні різним категоріям користувачів [17].

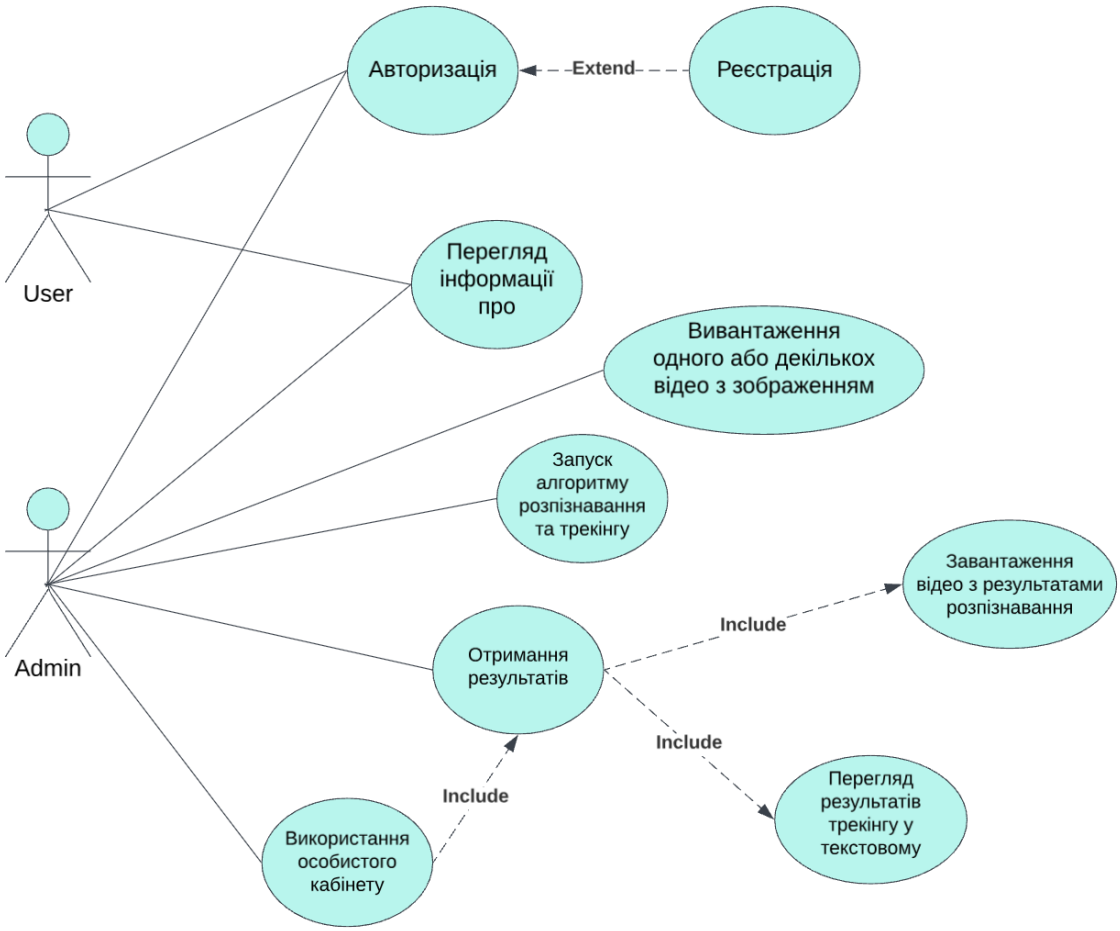

Рисунок 3.3 – Діаграма прецедентів взаємодії користувача із системою ідентифікації та трекінгу

На діаграмі показано акторів системи – зовнішніх користувачів або компонентів, які ініціюють взаємодію із застосунком. Для кожного з них виокремлено відповідні прецеденти (use cases) – типові сценарії використання функціоналу. Така візуалізація дозволяє чітко простежити, хто і які саме дії може виконувати в системі, а також визначити рівень доступу до окремих модулів або функцій.

Дана діаграма є основою для проєктування інтерфейсів і визначення прав доступу, що забезпечує зручність користування та інформаційну безпеку системи. Для деталізації логіки виконання окремих сценаріїв використано діаграму послідовностей — один із ключових інструментів UML, який візуалізує взаємодію між об’єктами в межах конкретного прецеденту [18]. На діаграмі (рис. 3.4) показано, як елементи системи – клієнт, сервер, база даних та інші модулі – обмінюються повідомленнями у визначеній послідовності. Це дозволяє перевірити логіку бізнес-процесів і коректність обробки запитів.

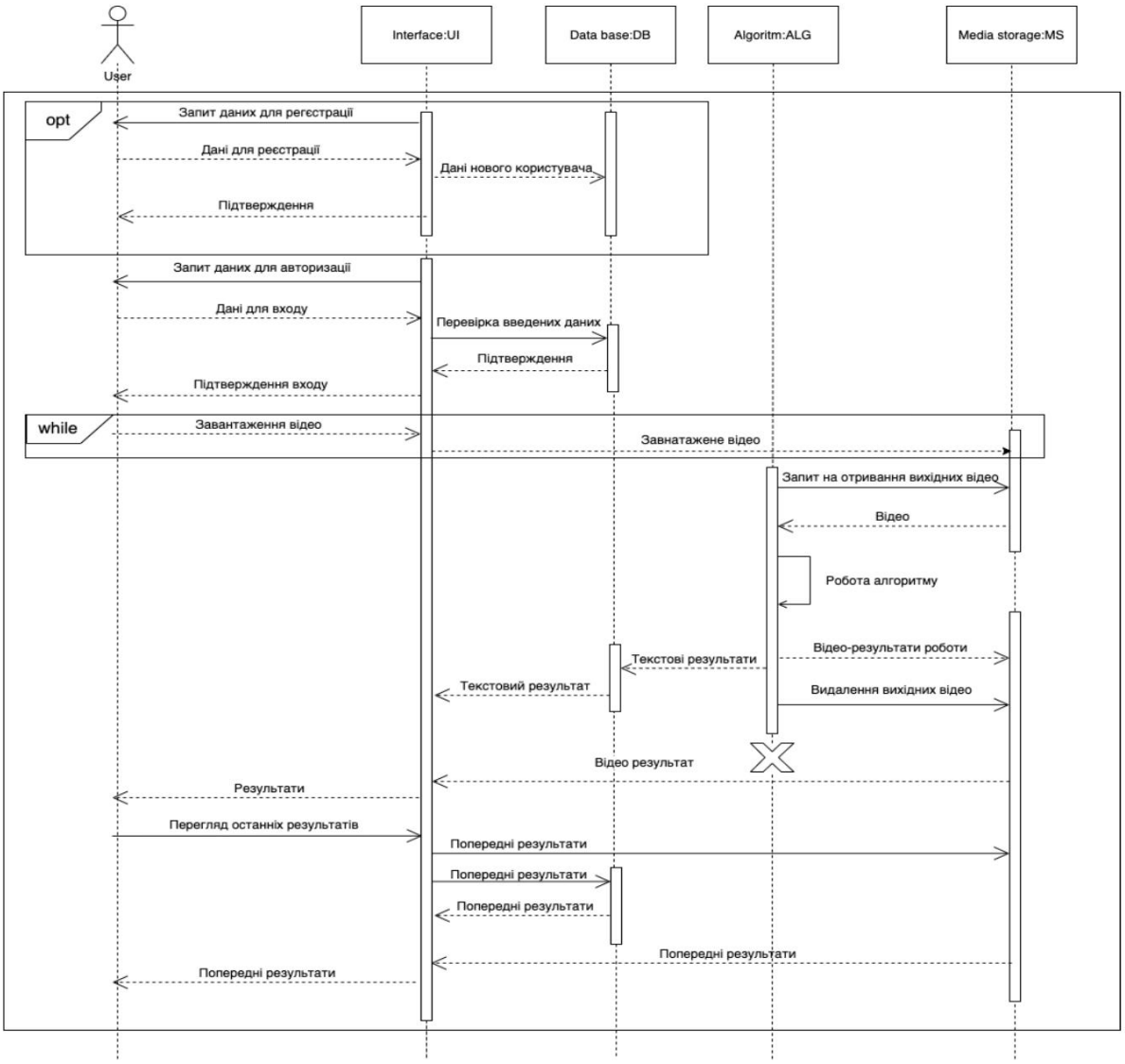

Рисунок 3.4 – Модель послідовних дій системи під час взаємодії з користувачем

Дана діаграма послідовностей моделює типовий випадок використання: надсилання користувачем запиту, його обробку на сервері, взаємодію з базою даних і повернення відповіді – відображаючи повний життєвий цикл виконання одного запиту [20-21].

3.3 Інсталяція системи, вимоги до апаратного забезпечення та тестування

Система розроблена із застосуванням сучасних вебтехнологій, тому не потребує окремої інсталяції на пристрої користувачів. Доступ до функціоналу здійснюється через будь-який сучасний веб-браузер, що підтримує стандарти HTML5, CSS3 та JavaScript, за умови стабільного підключення до Інтернету.

Перед початком роботи користувач створює персональний обліковий запис через інтерфейс реєстрації, доступний з головного меню веб-застосунку (див. рис. 3.5). Форма реєстрації (рис. 3.6) передбачає введення базових даних, необхідних для ідентифікації та авторизації в системі. Цей процес є частиною початкового тестування системи на коректність взаємодії з користувачем і перевірки базового функціоналу.

Рисунок 3.5 – Інтерфейс переходу до сторінки реєстрації користувача

У разі, якщо користувач вже зареєстрований у системі, для входу необхідно перейти до сторінки авторизації, доступ до якої здійснюється через головне меню веб-застосунку (див. рис. 3.5). На рисунку 3.7 представлено форму авторизації, яка дозволяє ввести облікові дані (логін та пароль) з метою доступу до персоналізованого функціоналу системи [20].

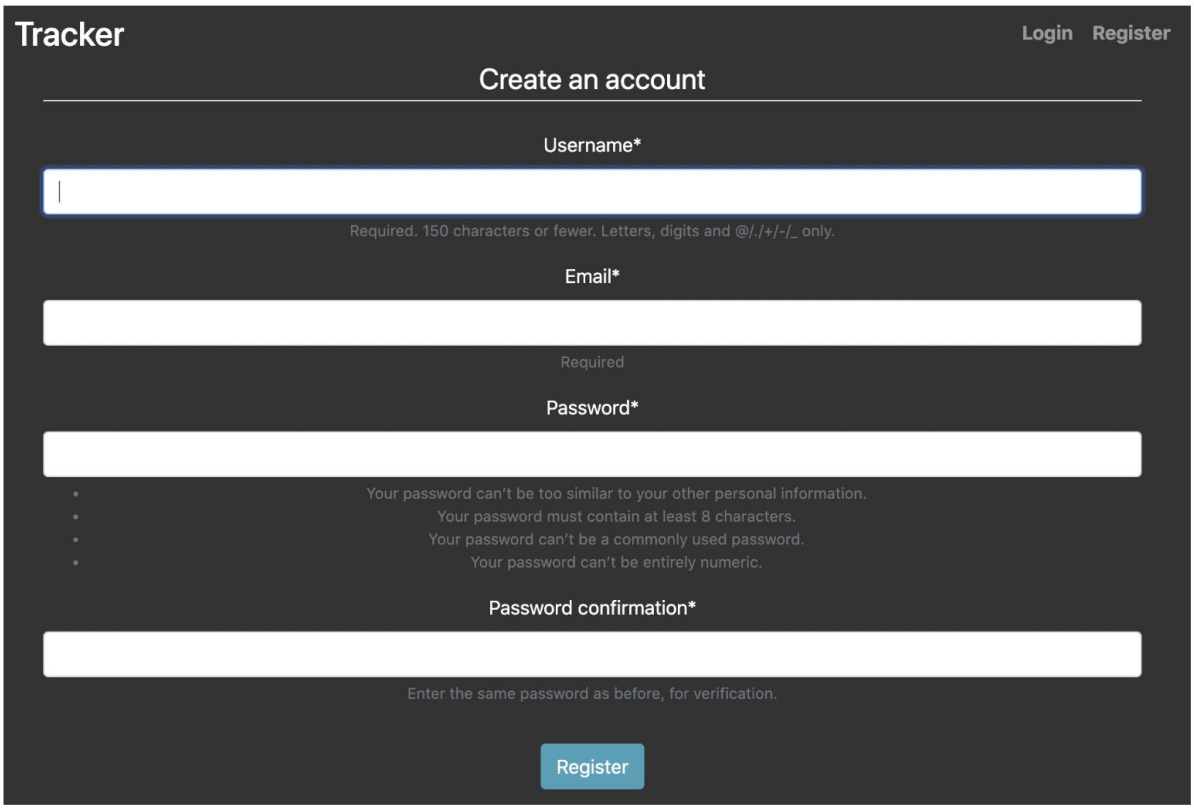

Рисунок 3.6 – Форма створення нового облікового запису в системі

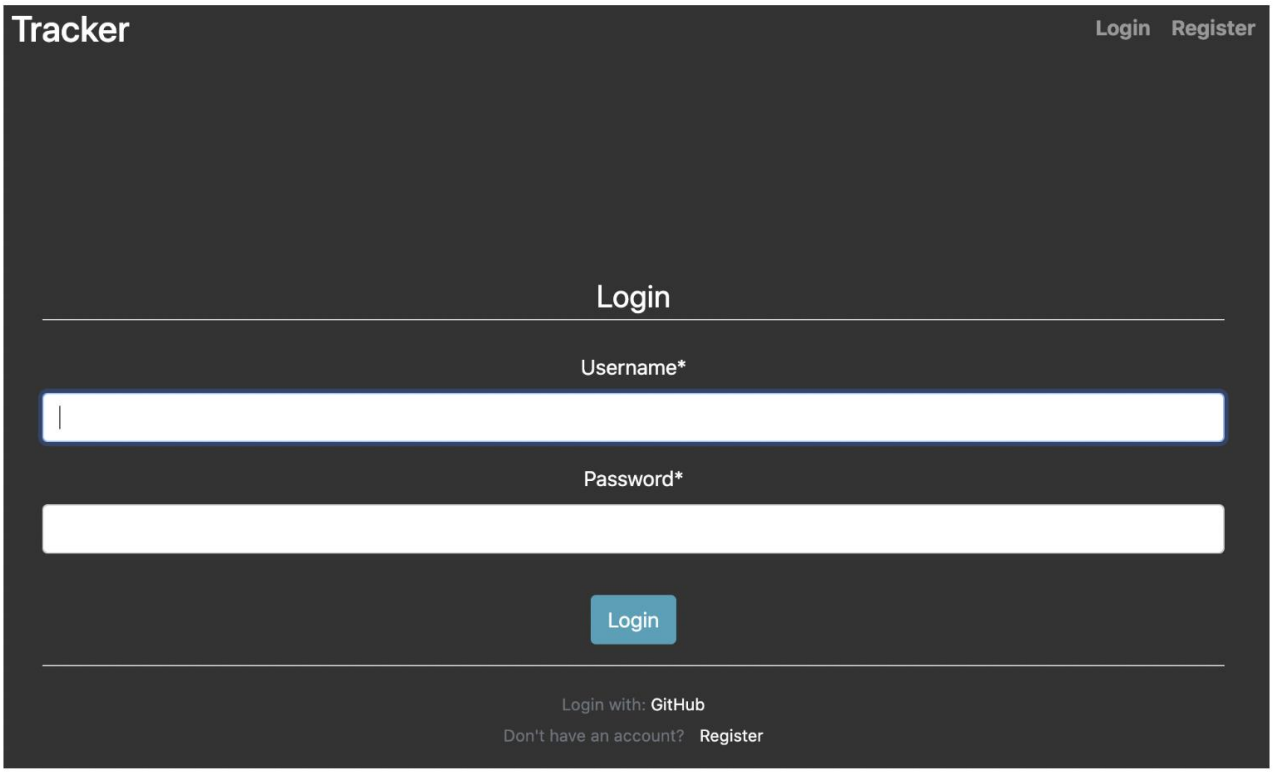

Рисунок 3.7 – Форма входу до облікового запису користувача

Крім стандартної форми реєстрації та входу, система також підтримує автентифікацію через соціальну мережу GitHub. Для цього користувач може перейти на сторінку авторизації або реєстрації та обрати відповідний варіант входу через GitHub (див. рис. 3.8). Такий спосіб дозволяє значно спростити процес ідентифікації користувача, скорочуючи час доступу до функціоналу системи.

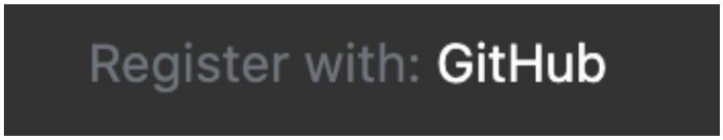

Рисунок 3.8 – Можливість входу в систему через обліковий запис GitHub

Після вибору варіанту входу через GitHub система автоматично перенаправляє користувача на офіційний сайт цієї платформи для проходження процедури автентифікації. На рисунку 3.9 представлено приклад інтерфейсу GitHub, через який користувач надає дозвіл на доступ до свого профілю для подальшої реєстрації або авторизації в системі.

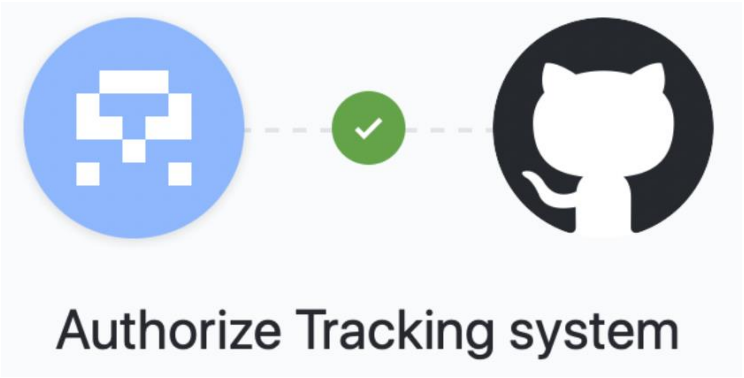

Рисунок 3.9 – Вікно автентифікації користувача через платформу GitHub

Після успішної авторизації користувач отримує доступ до персоналізованого інтерфейсу та розширеного функціоналу системи. Зокрема, активуються додаткові пункти меню, які дозволяють завантажувати відео, переглядати результати обробки та керувати обліковим записом. На рисунку 3.10 представлено оновлений вигляд головного меню після входу до системи.

Рисунок 3.10 – Інтерфейс користувача після авторизації

Користувач має можливість завантажити один або кілька відеофайлів у форматах .MOV або .MP4 для запуску основного функціоналу системи (див. рис. 3.11). Після завантаження відображається перелік відеофайлів із зазначенням їх назв та форматів, що дозволяє перевірити правильність вибраних даних (рис. 3.12). У разі помилкового завантаження передбачена можливість очищення списку – усі відеофайли можуть бути видалені одним натисканням (рис. 3.13).

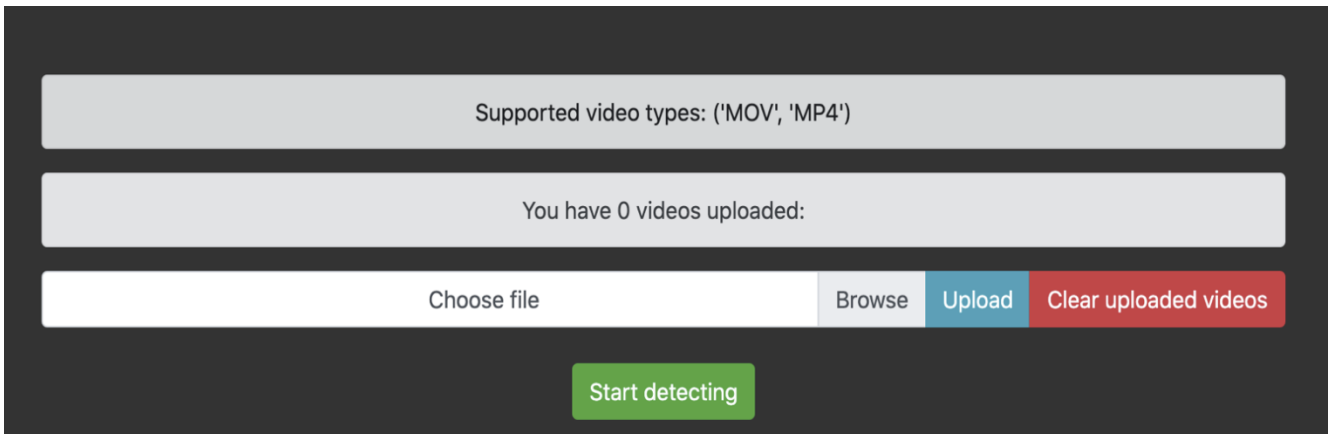

Рисунок 3.11 – Завантаження відеофайлів для обробки

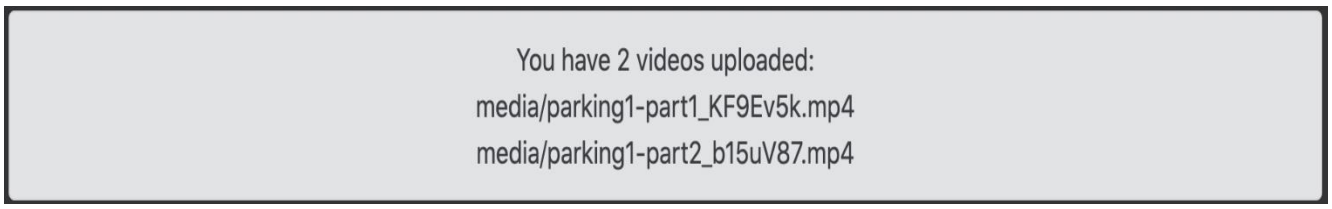

Рисунок 3.12 – Перевірка завантажених відео: назва та формат

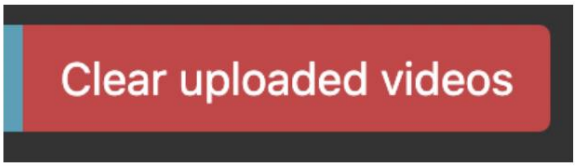

Рисунок 3.13 – Очищення списку завантажених відеофайлів

Після перевірки коректності завантажених відеофайлів користувач може ініціювати процес обробки вхідних даних. Для цього передбачено окремий елемент інтерфейсу, який запускає алгоритми розпізнавання та трекінгу об’єктів у відеопотоці (див. рис. 3.14).

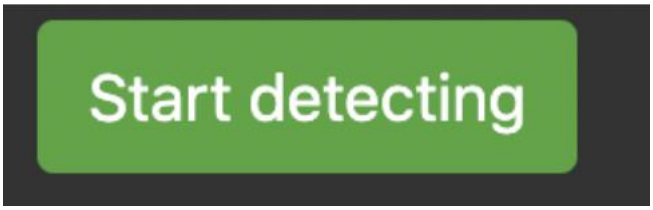

Рисунок 3.14 – Ініціація обробки вхідних відеофайлів

У процесі обробки даних користувач спостерігає індикатор активності у вигляді анімації, яка свідчить про те, що система виконує розрахунки. Анімація залишається активною до моменту завершення обробки всіх завантажених відеофайлів. На початковому етапі система виконує розбиття відео на послідовність окремих кадрів для подальшого аналізу (див. рис. 3.15, рис. 3.16) [20].

Рисунок 3.15 – Початковий етап обробки: розбиття відео на кадри

Рисунок 3.16 – Візуалізація процесу сегментації відеопотоку

Після завершення попередньої обробки система переходить до етапу локалізації та розпізнавання номерного знака. Алгоритм виявлення визначає точне розташування номеру на зображенні, після чого застосовується модель оптичного розпізнавання, яка ідентифікує символи та відображає результат безпосередньо на кадрі (див. рис. 3.17, рис. 3.18).

У процесі формування остаточних результатів ідентифікації та трекінгу система генерує парні зображення, на яких відображено автомобіль разом із розпізнаним реєстраційним номером (рис. 3.19). Додатково формується таблиця результатів, у якій користувач може переглянути повну інформацію про всі виявлені транспортні засоби на відео, включаючи номерні знаки та часові мітки (рис. 3.20).

Рисунок 3.17 – Локалізація номерного знака на кадрі відео

Рисунок 3.18 – Відображення розпізнаного номеру на зображенні

Рисунок 3.19 – Генерація парного зображення: транспортний засіб і номер

Total results for camera

Plate	Seconds of video	Total seconds
CA4668GE	3.1	0.0
CAM688C	3.15	0.0
CA6668BE	3.2 3.25 3.3 3.35 3.4 3.45 3.5 3.55 3.6 3.65	0.44999999999999973
CA16480	4.35	0.0
CA8186BI	4.45 4.5 4.55 4.6 4.65 4.7	0.25
CA1708CA	6.55 6.6	0.04999999999999982
CA8700CA	6.65 6.7 6.75	0.09999999999999964
CA81348	10.95	0.0
CA4315AX	11.05	0.0
CA6115AX	11.15	0.0
CAAJ11AM	11.2	0.0
CA8115	11.25	0.0
CA2728HP	16.35 16.45 16.5 16.6 16.65 16.7	0.34999999999999787
CA7728HF	16.55	0.0

Рисунок 3.20 – Таблиця результатів ідентифікації та трекінгу авто

3.4 Тестування системи: модульне та ручне (з прикладами)

Модульне тестування (англ. *Unit Testing*) – це процес перевірки окремих компонентів програмного забезпечення в ізольованому середовищі. Його головна мета – підтвердити коректність роботи кожного окремого модуля незалежно від інших частин системи. Такий підхід дозволяє локалізувати помилки на ранньому етапі розробки, полегшуючи налагодження та рефакторинг коду [21].

Кожен тест виконується окремо в штучно створеному середовищі за допомогою спеціальних засобів – драйверів і заглушок, що імітують поведінку інших компонентів. Це дозволяє зосередитись виключно на поведінці конкретної функції або класу. Модульні тести, як правило, створюються безпосередньо розробником і є надійним інструментом для контролю якості коду під час усієї розробки. Крім того, тести виконують роль «живої документації», що описує очікувану поведінку кожного класу або методу.

У межах даної дипломної роботи було реалізовано набір **модульних тестів** з використанням стандартної тестової інфраструктури, вбудованої в Django. Для кожної окремої підсистеми системи ідентифікації та трекінгу було створено окремі тести, які охоплюють основні функціональні сценарії. Це дозволило не лише перевірити стабільність роботи ключових компонентів, але й гарантувати правильність виконання логіки програми при оновленнях та розширенні функціоналу.

Для перевірки працездатності окремих модулів системи було проведено модульне (юніт) тестування. Нижче наведено приклади тестових випадків, що демонструють поведінку ключових функціональних компонентів.

Тестовий випадок 1 – Перевірка формату вхідних файлів

Необхідно перевірити, чи система коректно фільтрує вхідні файли за форматом.

Умова: для запуску обробки користувач має завантажити відеофайли у форматі .mp4 або .mov. Система повинна ігнорувати всі інші типи файлів.

Вхідні дані:

- hello.txt
- one.mp4
- 12_day.png
- work.pdf
- video.mov

Очікуваний результат:

– до обробки допущено лише файли one.mp4 та video.mov.

Фактичний результат:

– система коректно обробила лише дозволені формати.

Результат тесту: пройдено успішно.

Тестовий випадок 2 – Перевірка розбиття відео на кадри

Необхідно перевірити правильність роботи модуля розбиття відеопотоку на послідовність зображень.

Вхідні дані:

– відеофайл у форматі .mp4 тривалістю 10 секунд

Очікуваний результат:

– генерація 20 кадрів у форматі .jpg (за умов частоти 2 кадри на секунду)

Фактичний результат:

– система створила некоректну кількість кадрів

Результат тесту: непройдено (див. рис. 3.21)

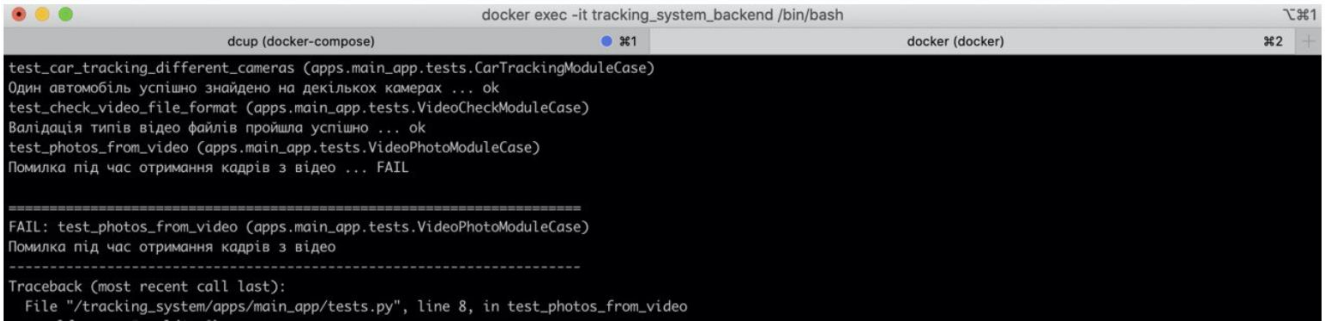

Рисунок 3.21 – Некоректний результат розбиття відео на кадри під час тестування

Після отримання негативного результату під час модульного тестування функції розбиття відео на кадри було проведено налагодження модуля та оптимізацію відповідного алгоритму. З метою підтвердження виправлення помилки було виконано регресійне тестування, яке продемонструвало, що модуль функціонує коректно й відповідає очікуваній поведінці.

Тестовий випадок 3 – Перевірка функціоналу трекінгу об’єкта

Метою тесту є перевірка здатності системи коректно відстежувати один і той самий транспортний засіб на відео з різних камер, а також уникати

помилкової ідентифікації подібних номерів. Відповідно до вимог, модуль трекінгу повинен визначати повторну появу автомобіля незалежно від джерела відео та формувати точні часові мітки без врахування схожих, але різних об’єктів.

У рамках тестування було використано три відеокадри: на першому з камери 1 зафіксовано авто з номером AA0000TO о 12:45:16, на другому (також з камери 1) – інший автомобіль із номером CA0000TE о 12:47:09, а на третьому – повторне фіксування авто AA0000TO вже на камері 2 о 12:47:34.

Очікувалося, що система об’єднає появи автомобіля з номером AA0000TO як один об’єкт із двома часовими мітками та не включить до цього треку інший транспортний засіб зі схожим номером.

Фактичні результати підтвердили правильність роботи: обидва кадри з AA0000TO були об’єднані, а авто CA0000TE не було помилково асоційоване з ними.

Ручне тестування є важливою складовою комплексної перевірки програмного забезпечення і, як правило, виконується перед фінальним впровадженням системи. Цей тип тестування передбачає перевірку функціональності додатку без використання автоматизованих інструментів – тестувальник безпосередньо взаємодіє з інтерфейсом системи, імітуючи дії кінцевого користувача.

Процес ручного тестування може бути як неформальним, так і структурованим – з використанням формалізованих тест-планів, сценаріїв дій та контрольних списків. Перевірка може здійснюватися з різних ролей: як від імені адміністратора, так і звичайного користувача, що дозволяє оцінити працездатність системи з різних точок зору.

Хоча ручне тестування зазвичай виконують окремі тестувальники, воно також може здійснюватися безпосередньо розробниками. У цьому випадку воно спирається на глибоке розуміння внутрішньої логіки системи та дозволяє виявити неочевидні помилки, які важко зафіксувати автоматизованими тестами.

Тестовий випадок 1 – Використання системи неавторизованим користувачем

Метою першого тестування було перевірити, чи система блокує доступ до основного функціоналу для неавторизованих користувачів. У цьому сценарії гість, який не пройшов авторизацію, намагався ініціювати запуск алгоритму розпізнавання та трекінгу об’єктів. Очікувалося, що система автоматично перенаправить його на сторінку реєстрації або авторизації. У результаті перевірки було підтверджено, що перенаправлення відбулося коректно (див. рис. 3.22), отже, тест завершено успішно.

Другий тестовий випадок мав на меті перевірити, як система обробляє помилкові дії користувача при завантаженні файлу з недопустимим форматом. Користувач здійснив спробу завантажити файл у форматі .jpg, хоча система підтримує лише .mp4 або .mov. Очікувалося, що система відхилить файл, відобразить відповідне повідомлення про помилку й повідомить про допустимі формати. Фактичний результат підтвердив правильність реалізації — повідомлення з поясненням було показано коректно, тому тест також вважається пройденим успішно [20].

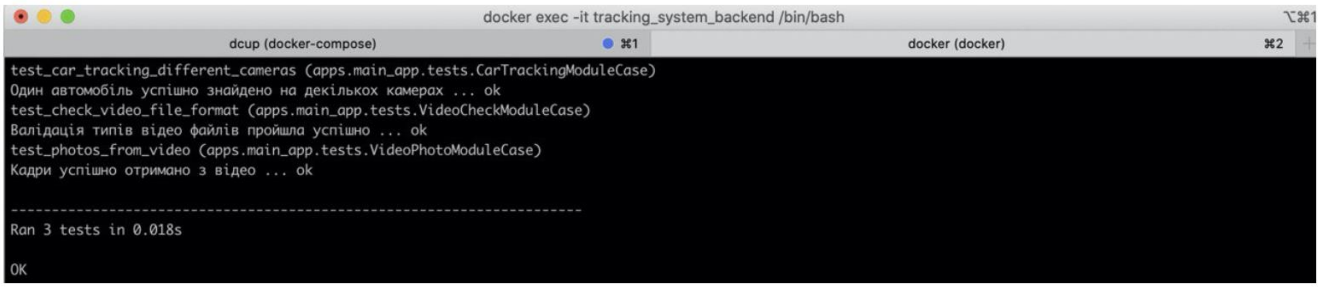

Рисунок 3.24 – Інтерфейс завершення тестування з позитивним результатом

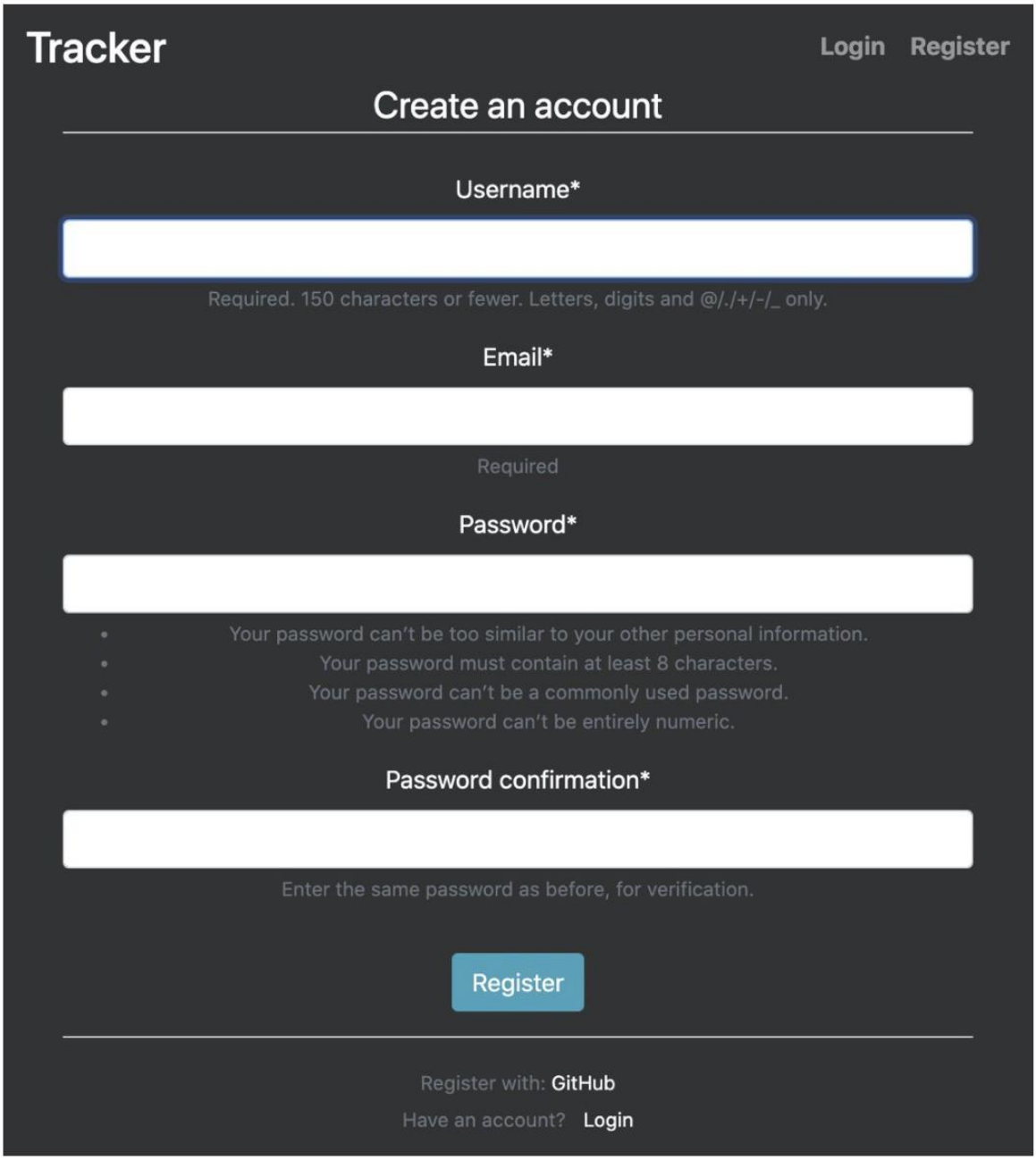

Рисунок 3.22 – Веб-сторінка реєстрації нового користувача

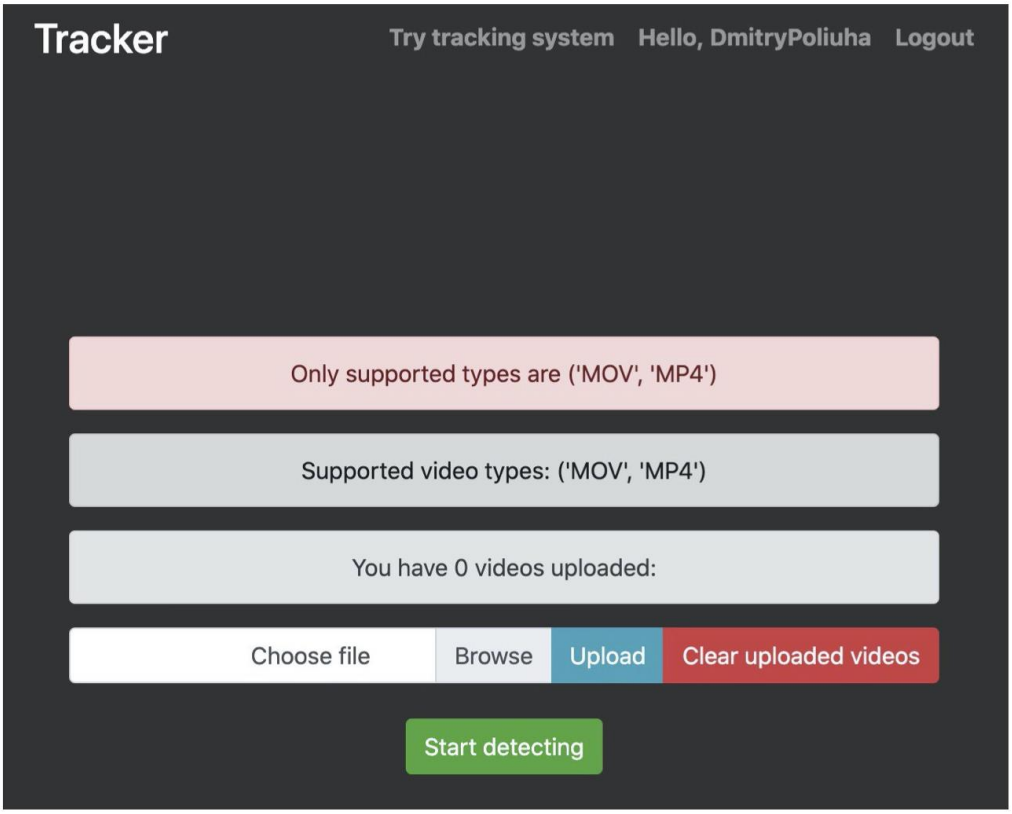

Рисунок 3.23 – Інтерфейсне сповіщення про помилку користувача

Проведені тести – як модульні (unit), так і ручні (manual) – засвідчили, що розроблена система функціонує стабільно та відповідає визначеним вимогам якості (див. рис. 3.24). У ході тестування було виявлено помилку в одному з компонентів, що була успішно усунена завдяки своєчасному запуску юніт-тесту.

Систематичне тестування дозволило на ранніх етапах розробки виявити критичні недоліки, запобігти їхньому впливу на подальшу роботу системи та забезпечити правильне функціонування ключових підсистем. У результаті користувачі можуть бути впевнені в надійності, стабільності та якості програмного продукту [21-22].

Висновок до розділу 3

У цьому розділі розроблено та застосовано комплексну стратегію тестування для оцінки надійності та стабільності функціонування створеної системи. Було впроваджено модульне (unit) тестування з використанням

вбудованих засобів Django, що дозволило перевірити логіку окремих підсистем у ізольованому середовищі. Також реалізовано ручне тестування інтерфейсів із позиції кінцевого користувача, що охопило найбільш поширені сценарії взаємодії з додатком.

Продемонстровано, що система відповідає заявленим вимогам – успішно обробляє вхідні відео, здійснює розпізнавання та трекінг об’єктів, фіксує час появи транспортних засобів і забезпечує доступність результатів через зручний інтерфейс. Проведені тести виявили низку помилок на ранніх етапах, які були усунені, що підтверджує ефективність вибраної моделі тестування.

У результаті реалізовано програмний продукт, який пройшов повний цикл перевірки функціональності, включаючи регресійне тестування після усунення виявлених дефектів. Отримані результати засвідчують готовність системи до подальшого впровадження та її відповідність критеріям якості, що є важливою передумовою для її використання в реальному середовищі.

ВИСНОВКИ

У процесі виконання кваліфікаційної роботи було розроблено інтелектуальну систему відеоспостереження з використанням AI-асистента, що здатна автоматично виявляти та трекати об’єкти у відеопотоці.

У першому розділі роботи проаналізовано сучасні підходи до побудови систем відеоспостереження, розкрито стан розвитку відповідних технологій, а також розглянуто роль комп’ютерного зору та AI у відеоаналітиці.

У другому розділі розглянуто постановку задачі, сформульовано вимоги до функціоналу, обґрунтовано вибір архітектурного підходу та програмних інструментів.

У третьому розділі деталізовано засоби реалізації, серед яких Python та бібліотеки PyTorch і NumPy для реалізації штучного інтелекту, Django для серверної частини, HTML/CSS/JavaScript для фронтенду, а також Docker, Hypercorn і NGINX для оптимізації розгортання. Побудовано архітектуру програмного комплексу на основі MVT-моделі, що забезпечує модульність, масштабованість і зручність підтримки.

У четвертому розділі продемонстровано роботу системи, описано алгоритм обробки відео, сценарії взаємодії з користувачем, а також діаграми прецедентів та послідовностей. Проведено повноцінне тестування: як юніт-тести на рівні окремих модулів, так і мануальне тестування ключових сценаріїв. Виявлені під час тестування помилки було оперативно усунуто, що свідчить про ефективність обраного підходу до перевірки якості.

Таким чином, робота виконана відповідно до поставлених цілей та завдань, створено повнофункціональний прототип системи, що може бути адаптований до різних сфер – від безпеки до логістики. Отримані результати підтверджують практичну цінність запропонованого рішення та його готовність до подальшого вдосконалення.

ПЕРЕЛІК ПОСИЛАНЬ

1. Долішній, А. І. Системи відеоспостереження: принципи побудови та технології [Текст] / А. І. Долішній, М. В. Величко. – Львів: Видавництво Львівської політехніки, 2020. – 224 с.

2. Пахомов, В. П. Відеоспостереження та відеоаналітика: основи побудови інтелектуальних систем безпеки [Текст] / В. П. Пахомов. – Київ: Наукова думка, 2021. – 312 с.

3. Redmon, J., Farhadi, A. YOLOv3: An Incremental Improvement [Електронний ресурс]. – Режим доступу: https://arxiv.org/abs/1804.02767. – Дата звернення: 25.04.2025.

4. OpenCV documentation. Cascade Classifier [Електронний ресурс]. – Режим доступу: https://docs.opencv.org/4.x/db/d28/tutorial_cascade_classifier.html. – Дата звернення: 25.04.2025.

5. Krizhevsky, A., Sutskever, I., Hinton, G. E. ImageNet Classification with Deep Convolutional Neural Networks [Текст] // Communications of the ACM. – 2017. – Vol. 60, No. 6. – P. 84–90.

6. Hikvision. Product Overview and AI Solutions [Електронний ресурс]. – Режим доступу: https://www.hikvision.com. – Дата звернення: 25.04.2025.

7. Dahua Technology. Smart Video Surveillance Solutions [Електронний ресурс]. – Режим доступу: https://www.dahuatech.com. – Дата звернення: 25.04.2025.

8. Axis Communications. Video Surveillance Solutions [Електронний ресурс]. – Режим доступу: https://www.axis.com. – Дата звернення: 25.04.2025.

9. Графік захисту бакалаврських кваліфікаційних робіт на кафедрі штучного інтелекту [Електронний ресурс]. – Режим доступу:https://duikt.edu.ua/ua/news-1-576-14398-grafik-zahistu-bakalavrskih-kvalifikaciynih-robit-na-kafedri-shtuchnogo-intelektu_kafedra-shtuchnogo-intelektu– Дата звернення: 25.04.2025.

10. Zhang, Z., Zhang, J., Wang, H. Real-Time Object Detection Using Convolutional Neural Networks [Текст] // Journal of Visual Communication and Image Representation. – 2021. – Vol. 78. – Article 103145.

11. Rosebrock, A. Deep Learning for Computer Vision with Python [Текст] / A. Rosebrock. – New York: PyImageSearch, 2019. – 528 p.

12. Hussain, S., Muhammad, K. AI-Powered Video Surveillance: Challenges and Solutions [Текст] // IEEE Access. – 2022. – Vol. 10. – P. 95436–95452.

13. Подгаєцький О. О. Проблема штучного інтелекту / О. О. Подгаєцький // Україна і світ: гуманітарно- технічна еліта та соціальний прогрес [зб. тез Міжнар. наук.–теор. конференції студ. та аспір.

14. Глинський Я.М. Штучний інтелект. Інтелектуальні роботи /Я.М.Глинський, В.А. Ряжська В.А. – Львів: Деол, 2002. - 168 с.

15. Малиновський Б. М. Відоме і невідоме в історії інформаційних технологій в Україні / Б. М. Малиновський – К.: Академперіодика, 2001.– 214 с.

16. The Future of AI in the Telecom Industry[Електронний ресурс]URL:https://medium.com/@venkat34.k/the-future-of-ai-in-the-telecomindustry-5d4c83a60054.

17. Про затвердження Технічних вимог до телекомунікаційних мереж загального користування [Електронний ресурс]//URL:https://zakon.rada.gov.ua/laws/show/z0872-15.

18. Напрямки використання інтелектуальних систем//Копустинський К.В., Шаров С.В.//2019-ст.132.

19. Top 10 AI-Powered Telecom Companies in World//URL:https://www.aithority.com/technology/analytics/top-10-aipowered-telecom-companies-in-world(дата звертання 29.05.2025).

20. Liang Han, Jijuan Gong A Study on Exploring the Path of Psychology and Civics Teaching Reform in Universities Based on Artificial Intelligence (2022) Computional intelligence and neuroscience https://doi.org/10.1155/ 2022/4841387

21. Gabo S., Kempen K., Lingelbach K., Bipp T. Artificial intelligence in psychology: How can we enable psychology students to accept and use artificial intelligence? (2021) Psychology Learning and teaching, 21(1), pp. 37–56.

22. Grassini S. Shaping the Future of Education: Exploring the Potential and Consequences of AI and ChatGPT in Educational Settings (2023) Education Sciences, 13(7), art. no. 692. DOI: 10.3390/educsci13070692.

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ

Державний університет інформаційно-комунікаційних технологій

Кафедра штучного інтелекту

Кваліфікаційна робота на тему:

«СИСТЕМА ВІДЕОСПОСТЕРЕЖЕННЯ ДЛЯ АВТОМАТИЧНОГО ВИЯВЛЕННЯ ПОРУШЕНЬ ЗА ДОПОМОГОЮ AI-АСИСТЕНТА »

на здобуття освітнього ступеня бакалавра
зі спеціальності 122 Комп'ютерні науки
освітньо-професійної програми Штучний інтелект

Виконав: ПЕРЕСАДА Марія ШІД-41
Науковий керівник роботи: РАБОТА О.В.

КИЇВ - 2025

Мета роботи – розробити та реалізувати інтелектуальну систему відеоспостереження для автоматичного виявлення порушень за допомогою AI-асистента, забезпечивши ефективність виявлення та обробки подій на основі сучасних технологій штучного інтелекту.

Об'єкт дослідження – процес створення системи відеоспостереження для автоматичного виявлення порушень із використанням штучного інтелекту.

Предмет дослідження – методи комп'ютерного зору, алгоритми машинного навчання для розпізнавання порушень, архітектура побудови AI-асистента для системи відеомоніторингу.

Наукові завдання:

1. Проаналізувати сучасний стан систем відеоспостереження та виявити їхні основні обмеження в контексті виявлення порушень безпеки.
2. Дослідити методи комп'ютерного зору та машинного навчання, що можуть бути застосовані для автоматичної обробки відеопотоків і розпізнавання подій.
3. Розробити архітектуру інтелектуальної системи відеомоніторингу, яка включає модулі відеоаналітики, обробки даних та інтеграції з AI-асистентом.
4. Реалізувати прототип системи відеоспостереження з використанням алгоритмів розпізнавання порушень на основі штучного інтелекту.
5. Сформулювати практичні рекомендації щодо впровадження інтелектуальних систем відеомоніторингу в сферу громадської, приватної або критичної інфраструктури.

Рис. 1.1 – Структурна схема системи відеоспостереження з інтеграцією IoT-пристроїв

Технології автоматичного виявлення об'єктів та подій

Рис. 1.2 – Схема методів автоматичного виявлення об'єктів та подій у системах відеоспостереження

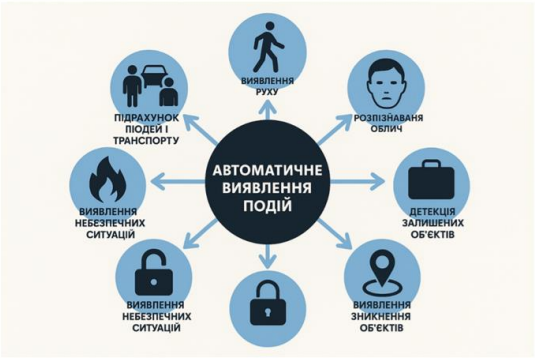

Рис. 1.3 – Основні задачі, що реалізуються в системах автоматичного виявлення подій

Роль штучного інтелекту та комп'ютерного зору в системах відеоаналітики

Таблиця 1.1 – Приклади використання алгоритмів CNN, YOLO та Haar Cascade у відеоспостереженні

Алгоритм	Основне призначення	Приклади використання	Особливості
CNN (Convolutional Neural Network)	Висока точність, потребує значних обчислювальних ресурсів, адаптивне навчання	Ідентифікація осіб, розпізнавання номерних знаків, класифікація транспортних засобів, аналіз поведінки	Висока точність, потребує значних обчислювальних ресурсів, адаптивне навчання
YOLO (You Only Look Once)	Швидке виявлення об'єктів у реальному часі	Виявлення людей, транспортних засобів, залишених речей, підрахунок об'єктів, моніторинг великих площ	Висока швидкодія, ефективність при багатьох об'єктах на сцені, одночасне виявлення кількох класів об'єктів
Haar Cascade	Виявлення об'єктів за геометричними ознаками	Розпізнавання облич, очей, усмішок, базових форм (наприклад, вікон або дверей)	Низькі вимоги до ресурсів, швидка робота, обмежена точність при складних умовах освітлення

6

Огляд існуючих рішень та їх порівняльна характеристика

Таблиця 2.1 – Порівняння ефективності та якості роботи відео- систем

Система	Тип рішення	Підтримка AI/аналітики	Інтеграція з IoT	Хмарні сервіси	Гнучкість налаштування	Тип ліцензії
Hikvision	Апаратура + ПЗ	Розпізнавання облич, підрахунок людей, виявлення вторгнень	Є	Є	Висока	Комерційна
Dahua Technology	Апаратура + ПЗ	AI-функції Smart Motion, детекція зон, контроль периметра	Є	Є	Висока	Комерційна
Axis Communications	IP-камери + ПЗ	Відеоаналітика, API для розробників	Є	Є	Висока	Комерційна
ZoneMinder	Програмне забезпечення	Базове виявлення руху	Немає	Немає	Середня	Відкрите ПЗ
Blue Iris	Програмне забезпечення	Детекція руху, зони контролю	Частково	Частково	Висока	Комерційна

Визначення критеріїв ефективності та якості роботи системи

Таблиця 2.2 – Критерії ефективності та якості роботи системи

Критерій / Метрика	Опис / Пояснення	Оптимальні значення
Precision (точність)	Частка правильно виявлених порушень серед усіх виявлених системою випадків.	≥ 0.85
Recall (повнота)	Частка реальних порушень, які були успішно виявлені системою.	≥ 0.85
F1-score	Середнє гармонійне між precision та recall; показник загальної точності.	≥ 0.85
Час обробки кадру	Максимальний допустимий час обробки одного кадру в реальному часі (≤ 1 с).	≤ 1 с (оптимально < 300 мс)
Використання апаратних ресурсів	Раціональне використання CPU, GPU та оперативної пам'яті.	<80% використання основних ресурсів
Стійкість до навантажень	Стабільна робота при обробці великого потоку даних або паралельному аналізі.	Без збоїв при ≥ 4 потоках одночасно
Рівень хибнопозитивних спрацювань	Частка помилкових спрацювань серед усіх позитивних рішень системи.	< 10%
Рівень хибнонегативних спрацювань	Частка пропущених реальних порушень серед усіх порушень.	< 10%

Використання допоміжного ПЗ

NumPy і PyTorch – використовуються для обробки даних та реалізації моделей глибокого навчання, що здійснюють аналіз відео.

Django (Python) – фреймворк для серверної частини, керує маршрутизацією, автентифікацією, базами даних та логікою запитів.

Docker Compose – автоматизує запуск усіх компонентів (сервер, БД, фронтенд, проксі) в одному середовищі.

Hypercorn – ASGI-сумісний сервер для асинхронної обробки запитів, що забезпечує високу продуктивність та підтримку WebSocket.

PostgreSQL – це потужна об'єктно-реляційна система управління базами даних з відкритим вихідним кодом, яка поєднує надійність, масштабованість і високий рівень відповідності сучасним стандартам СУБД.

Опис алгоритму виявлення порушень за допомогою штучного інтелекту

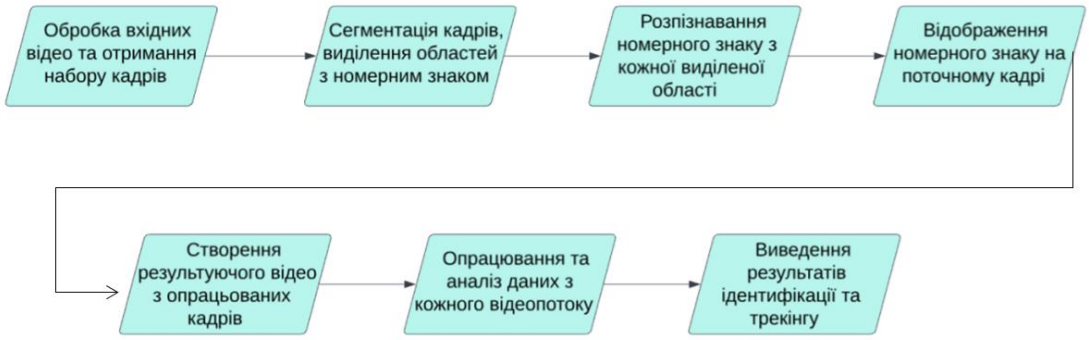

Рис. 2.1 – Логічна модель функціонування системи розпізнавання номерних знаків

Опис алгоритму виявлення порушень за допомогою штучного інтелекту

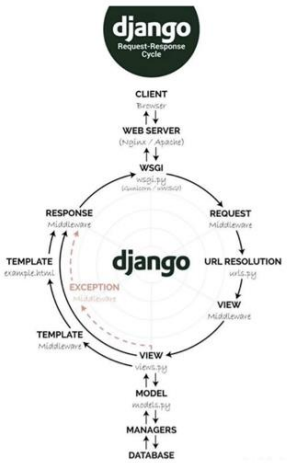

Рис. 2.2 – Принцип роботи архітектури MVT у веб-додатку

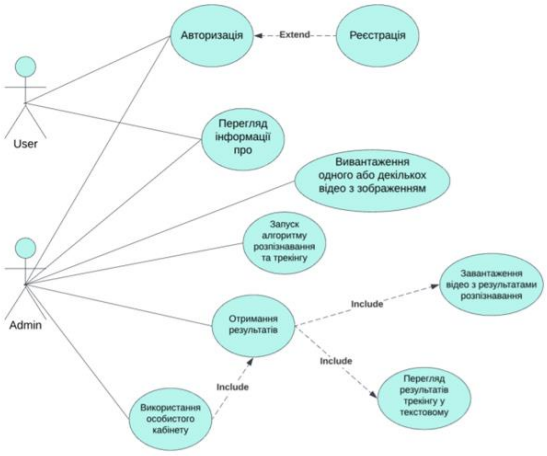

Рис. 2.3 – Діаграма прецедентів взаємодії користувача із системою ідентифікації та трекінгу

Демонстрація роботи програми

Рис. 3.1 – Початковий етап обробки: розбиття відео на кадри та локалізація номерного знака на кадрі відео

Демонстрація веб-застосунку та інтерфейсу користувача

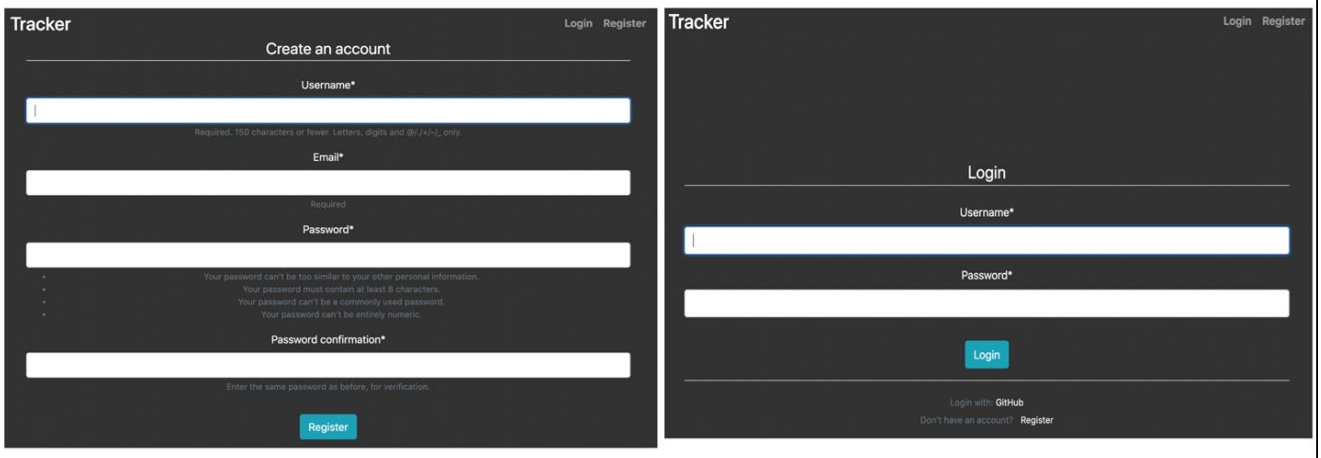

Рис. 3.2 – Форма створення нового та входу в обліковий запис в системі

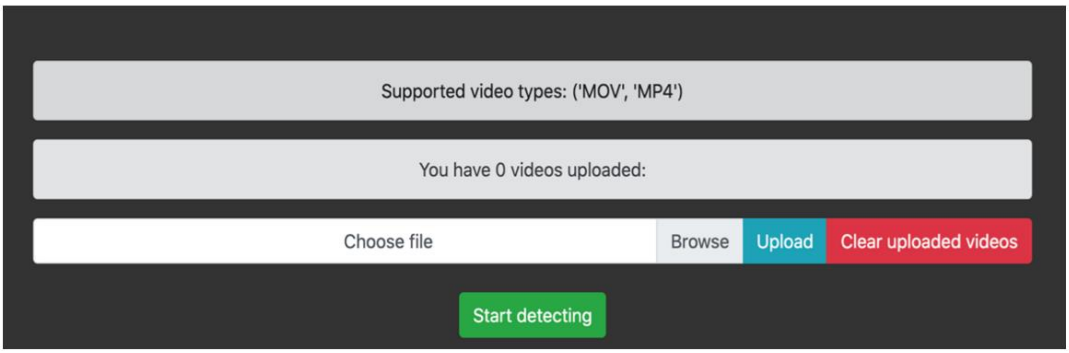

Рис. 3.3 – Завантаження відео файлів для обробки

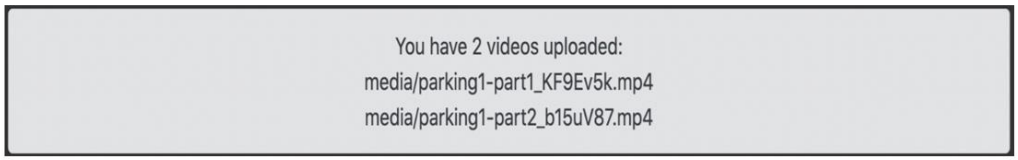

Рис. 3.4 – Перевірка завантажених відео: назва та формат

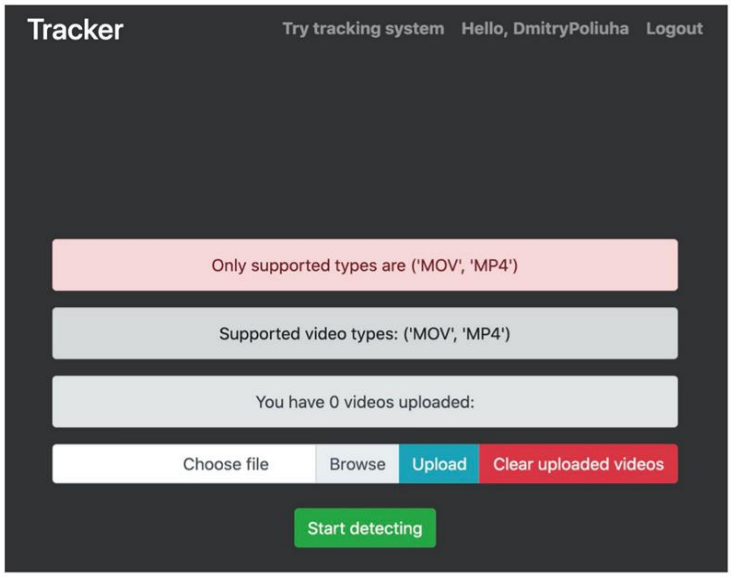

Рис. 3.5– Інтерфейс сповіщення про помилку користувача

Опис процесу мануального тестування

Рис. 3.6 – Генерація парного зображення: транспортний засіб і номер

Total results for camera

Plate	Seconds of video	Total seconds
CA4668GE	3.1	0.0
CAM688C	3.15	0.0
CA6668BE	3.2 3.25 3.3 3.35 3.4 3.45 3.5 3.55 3.6 3.65	0.44999999999999973
CA16480	4.35	0.0
CA8186BI	4.45 4.5 4.55 4.6 4.65 4.7	0.25
CA1708CA	6.55 6.6	0.04999999999999982
CA8700CA	6.65 6.7 6.75	0.09999999999999964
CA81348	10.95	0.0
CA4315AX	11.05	0.0
CA6115AX	11.15	0.0
CAAJ11AM	11.2	0.0
CA8115	11.25	0.0
CA2728HP	16.35 16.45 16.5 16.6 16.65 16.7	0.34999999999999787
CA7728HF	16.55	0.0

Рис. 3.7 – Таблиця результатів ідентифікації та трекінгу авто

16

Опис процесу unit тестування

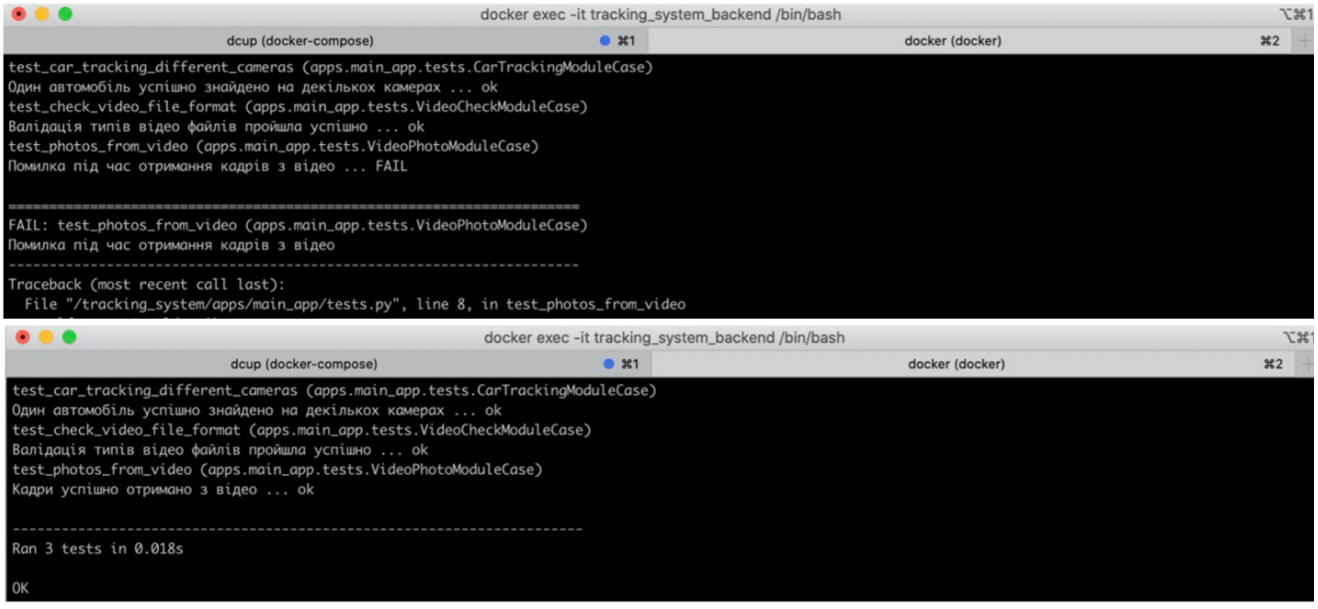

Рис. 3.8 – Результати розбиття відео на кадри під час тестування

ВИСНОВКИ

- Сформульовано технічне завдання, визначено функціональні вимоги до системи, обґрунтовано вибір архітектури та програмних засобів на основі принципів масштабованості й модульності.

- Реалізовано повнофункціональний прототип системи, що включає серверну частину на Django, фронтенд на HTML/CSS/JS та модулі обробки відео з AI на Python з використанням PyTorch і NumPy.

- Розроблено архітектуру на основі MVT-моделі, що забезпечує чітке розділення компонентів, спрощує підтримку та розширення системи.

- Проведено всебічне тестування: реалізовано юніт-тести для перевірки логіки модулів та ручне тестування ключових сценаріїв взаємодії з користувачем.

- Розроблена система демонструє високу ефективність роботи при виявленні порушень. Результати оцінювання показали ефективну та стабільну роботу розробленої системи в умовах реального часу. Система досягла точності виявлення порушень (precision, recall, F1-score) на рівні ≥ 0.85, а час обробки одного кадру не перевищує 1 секунди.