

Центральноукраїнський національний технічний університет
Механіко-технологічний факультет
Кафедра кібербезпеки та програмного забезпечення

”Допущено до захисту”
Завідувач кафедри кібербезпеки
та програмного забезпечення
д.т.н., професор
_____ Олексій СМІРНОВ
« ____ » _____ 2024 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за другим (магістерським) рівнем вищої освіти
на тему
“Дослідження та програмна реалізація системи відеоаналітики
на основі систем машинного зору”

КБПЗ - 2024

Виконав здобувач вищої освіти
II курсу, групи КІ-23М
ОПП «Комп’ютерна інженерія»
спеціальності 123 «Комп’ютерна інженерія»
_____ Рибка О.С.
« ____ » _____ 2024 р.

Керівник проекту
доктор філософії (PhD)
_____ Дреєва Г.М.
« ____ » _____ 2024 р.
Рецензент _____

Центральноукраїнський національний технічний університет
Факультет Механіко-технологічний
Кафедра Кібербезпеки та програмного забезпечення
Рівень вищої освіти магістр
Галузь знань 12 "Інформаційні технології"
Спеціальність 123 "Комп'ютерна інженерія"
Освітньо-професійна (освітньо-наукова) програма "Комп'ютерна інженерія"

ЗАТВЕРДЖУЮ
Завідувач кафедри
д.т.н., проф.
Олексій СМІРНОВ
« 6 » вересня 2024 року

ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗА ДРУГИМ (МАГІСТЕРСЬКИМ) РІВНЕМ ВИЩОЇ ОСВІТИ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

Рибі Олексію Сергійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи *Дослідження та програмна реалізація системи відеоаналітики на основі систем машинного зору*

2. Керівник роботи *Дреєва Ганна Миколаївна, доктор філософії (PhD)*

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 19-13 від 07.08.2024 року

3. Строк подання студентом роботи до захисту *2.12.2024 р.*

4. Мета та завдання випускної кваліфікаційної роботи: *Метою розробки є дослідження та програмна реалізація системи відеоаналітики на основі систем машинного зору*

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

- | | |
|--|--|
| <i>1. Призначення та область використання.</i> | <i>6. Наукова новизна.</i> |
| <i>2. Перегляд аналогічних існуючих систем.</i> | <i>7. Маркетингове та економічне обґрунтування ІТ-проєкту.</i> |
| <i>3. Опис і обґрунтування проектних рішень.</i> | <i>8. Заходи з охорони праці та техніки безпеки.</i> |
| <i>4. Етапи програмування системи.</i> | <i>9. Висновки.</i> |
| <i>5. Впровадження системи в промислову експлуатацію</i> | |

6. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

- | | |
|--|-----------------|
| <i>Наукова новизна</i> | <i>1 аркуш</i> |
| <i>Структурна схема системи</i> | <i>1 аркуш</i> |
| <i>Функціональна схема системи</i> | <i>1 аркуш</i> |
| <i>Діаграма процесів</i> | <i>1 аркуш</i> |
| <i>Блок-схема алгоритму роботи додатку</i> | <i>2 аркуша</i> |
| <i>Показники економічної ефективності</i> | <i>1 аркуш</i> |

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний	Доренська А.О.	05.10.2024	14.11.2024
Охорона праці	Марченко К.М., к.т.н., доцент	06.10.2024	16.11.2024

7. Дата видачі завдання « 6 » вересня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем	10.10.2024 р.	
2.	Постановка задачі, оформлення ТЗ	15.10.2024 р.	
3.	Розробка моделі компонента	20.10.2024 р.	
4.	Розробка структур даних	25.10.2024 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.10.2024 р.	
6.	Програмування алгоритмів	10.11.2024 р.	
7.	Розрахунок економічної ефективності	13.11.2024 р.	
8.	Розрахунки з охорони праці та техніки безпеки	15.11.2024 р.	
9.	Оформлення ПЗ	17.11.2024 р.	
10.	Попередній захист роботи	2.12.2024 р.	

Дата видачі завдання
« 6 » вересня 2024 р.

Підпис керівника

(прізвище та ініціали)Завдання прийнято до виконання
« 6 » вересня 2024 р.

Підпис здобувача

(прізвище та ініціали)

АНОТАЦІЯ

Риба О.С. Дослідження та програмна реалізація системи відеоаналітики на основі систем машинного зору. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2024.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи відеоаналітики на основі систем машинного зору.

Метою розробки є дослідження та програмна реалізація системи відеоаналітики на основі систем машинного зору.

Об'єктом дослідження є процес відеоаналітики на основі систем машинного зору.

Предметом дослідження є методи відеоаналітики на основі систем машинного зору.

Методи дослідження базуються на методах теорії розпізнання образів, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація системи відеоаналітики на основі систем машинного зору.

В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

Розроблено зручний інтерфейс користувача. Наведені інструкції по роботі з програмними засобами.

Програма може використовуватися на ПЕОМ з ОС Windows 10/11.

Програму розроблено в середовищі Python.

Ключові слова: комп'ютерна інженерія, відеоаналітика, машинний зір

ABSTRACT

Ryba O.S. Research and software implementation of a video analytics system based on machine vision systems. 123 Computer engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2024.

In this graduation thesis for the second (master's) level of higher education, software is developed, which is intended for a video analytics system based on machine vision systems.

The goal of the development is the research and software implementation of a video analytics system based on machine vision systems.

The object of the study is the process of video analytics based on machine vision systems.

The subject of the research is the methods of video analytics based on machine vision systems.

Research methods are based on pattern recognition theory methods, mathematical statistics methods, and software development methods.

The result of the work is a software implementation of a video analytics system based on machine vision systems.

In the process of working on the software model, an analysis of existing hardware and software was performed. All components of the developed software are fully described.

A convenient user interface has been developed. Instructions for working with software tools are provided.

The program can be used on a PC with Windows 10/11 OS.

The program was developed in the Python environment.

Keywords: computer engineering, video analytics, machine learning

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	3
ВСТУП.....	4
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	7
1.1 Призначення системи.....	7
1.2 Область застосування.....	9
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	11
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	11
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	18
2.3 Розгорнута постановка завдання	23
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	25
3.1 Опис функціонування системи	25
3.2 Розробка структурної схеми.....	35
3.3 Розробка функціональної схеми	43
3.4 Розробка діаграми процесів.....	46
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	48
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	48
4.2 Захист розробленого програмного забезпечення.....	62
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	64
6 НАУКОВА НОВИЗНА	69

						ВКРМ-123.24.0036.00.00.ПЗ		
Вим	Арк.	№ докум.	Підп.	Дата				
Розроб.	Риба О.С.				Дослідження та програмна реалізація системи відеоаналітики на основі систем машинного зору	Літ.	Аркуш	Аркушів
Перев.	Дресва Г.М.					М	1	97
Н.контр.	Коваленко А.С.					ЦНТУ КІ-23М		
Затв.	Смірнов О.А.							

7	МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ ІТ-ПРОЄКТУ	70
7.1	Визначення цільової аудиторії кінцевого готового продукту	70
7.2	Оцінка привабливості шляхом застосування методів експертних оцінок ...	70
7.3	Вибір методу оцінки вартості ПЗ	73
7.4	Розрахунок економічної ефективності від впровадження реалізованого ПЗ як фактору його привабливості.....	74
7.5	Пропозиція алгоритму просування проєкту розробки ПЗ	76
7.6	Оптимізація каналів збуту та шляхів реалізації ПЗ	76
7.7	Визначення ключових факторів успіху конкретного проєкту.....	78
8	ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ	80
8.1	Вступ.....	80
8.2	Аналіз умов праці на робочому місці ІТ-фахівця	81
8.3	Пропозиції щодо підвищення працездатності ІТ-фахівців.....	84
8.4	Розрахункова частина	86
8.5	Висновки до розділу.....	88
9	ОСНОВНІ ВИСНОВКИ.....	89
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	91

КБПЗ-2024

					ВКРМ-123.24.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		2

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

АРП	–	автоматичне регулювання посилення
АТМ	–	автоматичний телевізійний моніторинг
ЕПТ	–	електронно-променева трубка
ІБ	–	інформаційна безпека
ІС	–	інформаційна система
НСД	–	несанкціонований доступ
ПЗ	–	програмне забезпечення
ПЗЗ	–	прилади із зарядовим зв'язком
СКД	–	системи контролю даних
СЦК	–	система централізованого керування
ТВЛ	–	телевізійні лінії
ІР	–	Internet Protocol. Протокол міжмережної взаємодії
РІН-код	–	персональний ідентифікаційний код
ТСР	–	Transmission Control Protocol. Протокол керування передачею (даних) Основний транспортний протокол у стеці протоколів ТСР/ІР. Встановлює зв'язок між двома ПК й організує обмін даними в дуплексному режимі
ТСР/ІР	–	Transmission Control Protocol/Internet Protocol. Стек протоколів, що забезпечують організацію зв'язку між комп'ютерами в мережі Інтернет

ВСТУП

Актуальність теми. Відеоспостереження й відеоаналітика сьогодні використовуються не тільки для забезпечення безпеки. Все частіше вони позиціонуються в якості сучасних і ефективних інструментів для бізнесу, при цьому поряд із традиційними IP-камерами для рішення певного кола завдань застосовуються системи машинного зору. Технології, які ще недавно здавалися дорогими і складними, поступово стають усе більше масовими й доступними. Камери машинного зору й відеоаналітичні функції вже широко інтегруються в різні додатки, а області їхнього застосування виходять за рамки нішевих, спеціалізованих сегментів. Недорогі камери машинного зору можуть істотно поліпшити відеоспостереження на транспорті, у торгівлі й у ряді інших областей. Нові технології CMOS дозволяють одержати зображення кращої якості й підвищити пропускну здатність системи. Пропонуються платформи з убудованим програмним забезпеченням для розробки різних вертикальних рішень – зокрема, для потреб медицини, біології, роздрібної торгівлі, для застосування в банкоматах і на транспорті. Нерідко в таких додатках потрібно ідентифікувати дрібні деталі, і на виторг приходять камери машинного зору. Причому вартість такого рішення, включаючи друковану плату, камеру й оптику до неї, часом не перевищує 200 євро. Наприклад, у роздрібній торгівлі системи машинного зору (у тому числі стерео – 3D) застосовуються в торговельних автоматах для спрощення процесу продажів (ними навіть можна управляти за допомогою жестів), для підрахунку відвідувачів, їхнього профілювання й класифікації (визначення підлоги, віку й т.д.), а також для трекінгу й розпізнавання (у біометричних системах).

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи відеоаналітики на основі систем машинного зору.

					ВКРМ-123.24.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

- Огляд існуючих систем відеоаналітики на основі систем машинного зору.
- Дослідження системи відеоаналітики на основі систем машинного зору.
- Програмна реалізація системи відеоаналітики на основі систем машинного зору.

Об'єктом дослідження є процес відеоаналітики на основі систем машинного зору.

Предметом дослідження є методи відеоаналітики на основі систем машинного зору.

Методи дослідження базуються на методах теорії розпізнання образів, методах математичної статистики, методах розробки програмного забезпечення.

Наукова новизна отриманих результатів. У процесі рішення завдань, обумовлених цілями дослідження, отримані наступні результати:

- Удосконалено метод відеоаналітики на основі систем машинного зору.
- Розроблено вітчизняний продукт відеоаналітики на основі систем машинного зору, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі відеоаналітики на основі систем машинного зору.

Достовірність наукових результатів підтверджена теоретичними викладеннями, даними комп'ютерного моделювання, коректними дослідженнями параметрів на функціонуючій обчислювальній мережі, а також відповідністю отриманих результатів окремим результатам, наведеним у науковій літературі.

					ВКРМ-123.24.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

Робота апробована на LVII Науково-технічній конференції здобувачів вищої освіти LV науково-технічної конференції «Наука в ЦНТУ: основні досягнення та перспективи розвитку» (2024 р.), основні положення випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти надруковані у статті збірника праць молодих науковців ЦНТУ, випуск №15.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи відеоаналітики на основі систем машинного зору, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

КБПЗ – 2024

					VKPM-123.24.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Відеоаналітика або аналіз відео – це процес вилучення корисної інформації з відеозапису. Це може бути що завгодно: від підрахунку кількості людей на відео до ідентифікації конкретних об'єктів чи осіб. Сучасна відеоаналітика використовує комп'ютерний зір, який є областю штучного інтелекту, яка займається аналізом цифрових зображень і відео.

У різних галузях існує кілька різних програм для відеоаналітики. У багатьох випадках використання аналізу відеовмісту дає змогу автоматизувати завдання, які інакше виконувалися б вручну, наприклад, підрахувати кількість людей у відео або ідентифікувати конкретні об'єкти на кількох камерах прямого ефіру.

Аналіз відео з глибоким навчанням

Глибоке навчання – це підмножина машинного навчання, яке використовує нейронні мережі для вивчення шаблонів у даних. Нейронні мережі складаються з кількох шарів взаємопов'язаних вузлів обробки.

Швидкий прогрес глибокого навчання продемонстрував великий успіх у застосуванні ШІ для аналізу відео. Зокрема, алгоритми глибокого навчання використовуються для виявлення та відстеження об'єктів у відео, а також для розпізнавання конкретних дій.

Виявлення об'єктів у відеоаналітиці

Одним із найпоширеніших застосувань глибокого навчання для аналізу відео є виявлення та відстеження об'єктів. Це передбачає виявлення та відстеження конкретних об'єктів у послідовності розпізнавання відео. Популярні методи включають використання згорткової нейронної мережі (CNN) для вивчення складних шаблонів із даних.

					ВКРМ-123.24.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

Виявлення об'єктів у відеопотоках у реальному часі було одним із найважливіших завдань комп'ютерного зору. Найпопулярніші алгоритми виявлення об'єктів у відеоданих включають Mask R-CNN, YOLOv3, YOLOR і YOLOv7. Їх можна навчити на основі зібраних відеоданих або попередньо навчити на великих наборах даних зображень, таких як MS COCO.

Такі моделі глибокого навчання використовуються в програмному забезпеченні для аналізу відео та виявлення та відстеження об'єктів для підготовлених класів, таких як транспортні засоби, люди, світлофори тощо, у режимі реального часу. Більш розширене програмне забезпечення відеоаналітики надає функціональність для підрахунку об'єктів і аналізу на основі правил, наприклад, для підрахунку людей у місцях з великим скупченням людей.

Розпізнавання дій у відеоаналізі

Іншим типовим застосуванням глибокого навчання для аналізу відео є розпізнавання дій. Це передбачає розпізнавання певних дій у відеоряді або відеопотоках у реальному часі. Моделі глибокого навчання можна навчити класифікувати дії, що виконуються в різних контекстах або середовищах.

Розпізнавання відео або аналіз виявлення руху є дуже популярним для виявлення дій у сцені шляхом аналізу серії відеокадрів. Методи виявлення руху відео або аналізу прогресу включають прив'язку кадрів або зіставлення пікселів для виявлення горизонтальних і вертикальних змін між набором зображень або відеокадрів.

Більш просунуті методи застосовують розпізнавання/розуміння відео, оцінку пози, аналіз емоцій або розпізнавання обличчя для аналізу та розуміння контексту відеоданих. Однак ці складні завдання ШІ вимагають значних обчислювальних ресурсів і складної програмної інфраструктури.

					ВКРМ-123.24.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

1.2 Область застосування

Система управління відео (VMS) і штучний інтелект

Нова відеоаналітика на основі штучного інтелекту швидко набирає популярності. Основні користувачі включають тих, хто працював із традиційними системами відеоспостереження ще до появи штучного інтелекту (ШІ), і тих, хто хоче застосувати проривні технології для автоматизації.

Важливо розрізнити системи керування відео, які в основному керують потоками камер, і системи комп'ютерного бачення, які зосереджені на розпізнаванні відео та застосуванні розширеного аналізу відео в реальному часі для вирішення бізнес-завдань.

Система керування відео (VMS)

VMS – це програма, яка керує та записує відео з камер безпеки. VMS може забезпечити єдиний інтерфейс для перегляду живого та записаного відео з багатьох камер, а також можливості пошуку, відтворення та експорту. Більшість додатків VMS розроблено для роботи з певним типом камер, наприклад IP-камерами або камерами відеоспостереження. До популярних постачальників VMS належать Milestone, Avigilon, Axis, Bosch, Dahua, Hikvision, Honeywell і Pelco.

Більшість провайдерів VMS не є хмарними компаніями чи компаніями, що використовують штучний інтелект, і надавали продукти для відеоспостереження до появи штучного інтелекту. Проте все більше і більше систем керування відео стикаються з необхідністю додати можливості відеоаналітики для підтримки ручних операторів, які мають контролювати відеопотоки. Такі функції включають розпізнавання обличчя та людини та автоматичне позначення тегом або сповіщення.

Системи комп'ютерного зору

Система комп'ютерного бачення використовує алгоритми обробки зображень у багатоетапних конвеєрах комп'ютерного бачення для аналізу

					ВКРМ-123.24.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

зображень з метою отримання інформації з відеоданих. Системи комп'ютерного зору можуть вирішувати складні та специфічні для бізнесу завдання, які передбачають виявлення особи чи об'єкта, розпізнавання обличчя, розпізнавання діяльності, перевірку якості тощо. Системи комп'ютерного бачення можуть отримувати відеовхід від камер або VMS. Перегляньте наш список популярних компаній, що займаються комп'ютерним зором.

З розвитком технологій штучного інтелекту комп'ютерне бачення стає стратегічним питанням для всіх галузей. У результаті компанії почали впроваджувати портфоліо програм комп'ютерного зору для автоматизації завдань за допомогою відеоаналітики ШІ. Хоча існують точкові рішення, компанії, як правило, розробляють спеціальні системи комп'ютерного бачення, щоб відповідати вимогам бізнесу щодо інтеграції системи, гнучкості, економічності, конфіденційності даних, продуктивності та економічності.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи відеоаналітики на основі систем машинного зору, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-123.24.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

2.1 Огляд існуючих систем, технологій, архітектур, програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти

Способи визначення тривожних ситуацій

Розрізняють наступні способи налаштування відеоаналітики для автоматичного розпізнавання тривожних ситуацій:

– **Відеоаналітика на основі правил** (rule-based video analytics) є найпоширенішим типом відеоаналітики, у якому користувач заздалегідь визначає правила, що визначають тривожні події. Наприклад, користувач може задати заборонну зону, де не дозволено перебувати людям або заборонній зоні для паркування автомобілів. Система Kiproд являє приклад реалізації відеоаналітики на основі правил.

– **Відеоаналітика на основі статистичного навчання** (statistical learning video analytics) накопичує статистичні дані про поведження об'єктів і формує сигнал тривоги у випадку нестандартного поведження. Комплекс CasRetail і відеосервер CasNVR являє приклад реалізації відеоаналітики на основі статистичного навчання.

Міжнародні стандарти

Інтелектуальне відеоспостереження в цілому й відеоаналітика особливо перебувають на ранньому етапі стандартизації. Наступні міжнародні організації розробляють інтерфейси передачі даних і керування для систем відеоаналітики:

ONVIF (Open Network Video Interface Forum) поєднує близько 500 виробників і має сильні позиції в Росії, Європі і Японії. Описує інтерфейси взаємодії між IP-камерами, серверами відеоаналітики, відеореєстраторами, системами контролю доступу (СКД) і іншими компонентами. Відмітна риса ONVIF – використання протоколу SOAP для взаємодії між компонентами систем безпеки.

					ВКРМ-123.24.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

PSIA (Physical Security Interoperability Alliance) поєднує близько 65 виробників і має сильні позиції в Північній Америці. Відмітна риса PSIA – використання протоколу POST. У порівнянні с ONVIF, інтерфейс PSIA більше простий у реалізації, але менш гнучкий і масштабуємий.

CAP (Common Alerting Protocol) є протоколом передачі повідомлень про тривожні ситуації. На відміну від ONVIF і PSIA, інтерфейс CAP не спеціалізований для відеоспостереження й відеоаналітики.

Області застосування відео аналітики

Економічний ефект від впровадження відеоаналітики більшою мірою помітний у великих територіально-розподілених мережах відеоспостереження. У міру здешевлення технології відеоаналітика стає привабливою для домогосподарств і малого бізнесу.

Камери машинного зору – альтернатива IP-камерам

Смуга пропускання й ступінь стиску

IP-камери розроблені для роботи в середовищі з низькою пропускнуою здатністю, тому їх легко інтегрувати в існуючі комп'ютерні системи без ризику для цілісності таких систем. IP-камери налаштовуються віддалено й передають стислий відеопотік (MPEG-4, MJPEG або H.264). Пріоритет в оптимізації зображення віддається ефектній картинці на моніторі оператора. Камери машинного зору, виступаючи компонентом закритих систем обробки зображень, навпроти, призначені для роботи в середовищі з високою пропускнуою здатністю й забезпечують оптимальна якість відео. Цим пояснюється одне з основних технічних розходжень – стиск зображення. Тобто якщо IP-камера сильно стискає картинку, щоб заощадити ресурси при її передачі, промислова камера відправляє "сирі" (raw) дані, які будуть згодом оброблені потужним процесором на сервері. Такий підхід дозволяє зберегти дрібні деталі вихідного зображення, що незамінно, наприклад, у системах візуального контролю якості поверхонь. Якщо дефект перебуває на тій ділянці зображення, дані для якого відсутня, то система контролю якості помилково пропустить дефектний компонент як відповідному стандарту якості, тобто не впорається зі своїм завданням.

					ВКРМ-123.24.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

Таблиця 2.1 – Основні галузі застосування відеоаналітики із вказівкою використуваних аналітичних функцій.

Громадський транспорт	<ul style="list-style-type: none"> – Підрахунок пасажирів. – Виявлення падіння людей і сторонніх предметів на рейки. – Виявлення людей у забороненій зоні(наприклад, на території аеропорту). – Виявлення залишеного предмета. – Розпізнавання розшукуваних осіб. – Розпізнавання номерних знаків.
Безпечне місто й ЖКГ	<ul style="list-style-type: none"> – Виявлення неправильного паркування. – Виявлення скупчення людей. – Виявлення бійки. – Контроль якості відео. – Розпізнавання номерних знаків.
Небезпечні об'єкти, у тому числі ПЕК Промислове виробництво	<ul style="list-style-type: none"> – Охорона периметру. – Виявлення вогню й диму. – Виявлення парного проходу. – Моніторинг активності персоналу.
Торговельні й банківські мережі. Побутові послуги (кафе, салон стільникового зв'язка, перукарня, автомойка)	<ul style="list-style-type: none"> – Підрахунок клієнтів. – Аналіз довжини черги. – Класифікація клієнтів. – Оцінка уваги.
Розваги й спорт	<ul style="list-style-type: none"> – Підрахунок відвідувачів (уболівальників). – Виявлення скупчення людей. – Виявлення бійки або бійки. – Аналіз довжини черги. – Оцінка уваги.

Багатопотоковість

IP-камери можуть передавати відразу кілька потоків, стислих по різних алгоритмах (H.264, MPEG-4, MJPEG). Кожний кінцевий пристрій потім приймає сумісний відеопотік. Оператор може вибрати відеопотік з високим розв'язною здатністю MJPEG, що дозволить розрізняти на моніторі навіть дрібні деталі, тоді як максимально стислий відеопотік у форматі H.264 передається на архівування.

Камери машинного зору передають дані вже після обробки й аналізу, наприклад для цілей ведення виробничої статистики, оптимізації продуктивності встаткування й налаштування встаткування для усунення повторюваних помилок. По суті, зображення зберігаються тільки після перегляду.

Сумісність із системами реального часу

Сумісність із системами реального часу припускає, що камера почне захвата зображення негайно після одержання сигналу від тригера (тобто через заздалегідь відомий і добре прогнозований відрізок часу), після чого зображення буде відразу передано в систему, що відправила запит. Прийнятний діапазон затримки може варіюватися від мікросекунд до секунд. У камерах машинного зору можливість передачі зображень у режимі реального часу є обов'язковою умовою. Наприклад, при здійсненні візуального контролю якості виробів, що рухаються по конвеєрі з високою швидкістю, затримка в роботі системи приведе до збою контролю якості. Таким чином, затримка між одержанням сигналу й відправленням зображення є критично важливою. Не менш важлива й передбачуваність часу відгуку, тобто консистентність роботи комплексу. Наприклад, для системи, що генерує 300 кадрів/с, затримки більше декількох мікросекунд категорично неприйнятні. Однак і за межами промислових об'єктів пред'являється ця вимога – зокрема, у системах контролю дорожнього руху камери можуть бути синхронізовані зі світлофором (фіксація проїзду на червоне світло).

Багато які IP-камери не передбачають роботу в режимі реального часу. Звичайно системи відеоспостереження, як, наприклад, ті, що встановлено в

					ВКРМ-123.24.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

банках для моніторингу приміщень, експлуатуються в режимі автоматичного захвату й передачі зображень для одержання безперервного відеопотоку, без необхідності спрацьовування по тригері. Камера, настроєна на одержання 30 кадр/с, використовує внутрішній генератор імпульсів для захвату кадру з інтервалом 1/30 секунди. У деяких же випадках бажано запускати захват кадру в момент, обумовлений зовнішніми подіями, наприклад зовнішнім тригером, що спрацьовує при перетинанні автомобілем обраної границі.

Генерування сигналів уведення-виводу

Більшість камер машинного зору оснащені інтерфейсом уведення-виводу для керування зовнішніми пристроями. Наприклад, для включення світла в приміщенні тільки на час захвату зображення, коли людина входить у зону дії камери.

Області застосування

Промислові камери

Системи обробки зображень часто використовуються для контролю якості різної продукції, наприклад харчових продуктів. Завдяки цим зображенням виробник може приймати обґрунтовані рішення, що без промислової камери було б неможливо.

Іншими областями застосування камер машинного зору є електроніка, робототехніка й напівпровідникова промисловість.

Промислові камери передають незжаті (raw) зображення на ПК, де вони обробляються за допомогою потужних процесорів, які неможливо розмістити усередині камери. При цьому якість зображень при передачі не страждає, так як повністю відсутнє стиск.

ІР-камери

Через свої технічні характеристики ІР-камери прекрасно підходять для рішення завдань відеоспостереження. Вони можуть застосовуватися як класичні камери відеоспостереження для запобігання крадіжок, злочинства й вандалізму. Крім того, їх використовують для моніторингу виробничих процесів, а також

					ВКРМ-123.24.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

контролю обстановки на дорогах або виявлення порушень правил дорожнього руху. IP-камери служать і для дистанційного керування машинами, автомобілями й роботами. Залежно від області застосування вирішальними з погляду оптимальної експлуатації будуть різні характеристики камери.

Машинний зір на практиці

Ідеальним рішенням для систем біометричної ідентифікації по зображенню особи є камери машинного зору. Розглянемо одну з таких систем на прикладі контролю входу й виходу учнів на прохідних коледжу зв'язку.

Завдання

Керівництво коледжу було зацікавлено в пошуку альтернативи картам доступу з RFID-мітками як способу ідентифікації відвідувачів. Цілями побудови системи машинного зору були:

- контроль відвідуваності;
- запобігання передачі карт іншій особі;
- забезпечення безпеки об'єкта.

Принцип роботи

Система складається з комп'ютерів (одного сервера й одного робітника місця оператора), камер (по двох на прохід – одна на вхід і одна на вихід), контролерів доступу, турнікетів і ПЗ. Учні, викладачі й персонал занесені в біометричну базу даних.

Зображення, які щонайкраще підходять для наступного порівняння з картинкою, отриманою камерою, а також кут зйомки визначаються й вибираються системою автоматично. Процедура реєстрації була проведена з мінімальним і незручностями для учнів і викладачів – їм не довелося позувати, повертати голову певним чином і т.п.

Система використовує графічні співпроцесори (акселератори), що дозволяє домогтися величезної швидкості обробки на встаткуванні порівнянної ціни. З кожним обробленим зображенням лінійно збільшується точність роботи системи, так як помилки ідентифікації не накопичуються.

					ВКРМ-123.24.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

З метою нагромадження й збору реальної статистики учням і викладачам було запропоновано як і раніше носити із собою карту доступу й вирішити самостійно, що використовувати надалі: розпізнавання осіб або карти з RFID-міткою. Для підвищення залученості учнів і викладачів у роботу системи точки доступу були обладнані відкритими моніторами, щоб кожна людина могла бачити процес ідентифікації в режимі реального часу.

У результаті 91% учасників проекту зволіли використовувати систему розпізнавання осіб, а не карту доступу. При проведенні опитування вони назвали наступні причини: "так як так швидше", "карту легко забути будинку, а особа нікуди не дінеться"; "набагато простіше подивитися в камеру, чим шарити по кишенях або в сумці в пошуках карти доступу".

У багатьох спеціальних додатках камери машинного зору можуть виявитися прекрасною альтернативою традиційним IP-камерам. Вони пропонують ряд переваг, зокрема в сфері розпізнавання осіб, де необхідні незжаті зображення. Камери машинного зору передають достатній обсяг даних для надійної ідентифікації особистості. Забезпечують чудова якість зображення. Висока швидкість зйомки, що відіграє важливу роль у багатьох областях, – це, скоріше, перевага камер машинного зору, ніж IP-камер

Камери високого дозволу

Нова версія стандарту EN 50 132-7 враховує наявність на ринку відеоспостереження камер високого дозволу. Попередні вказівки були розраховані на камери стандарту PAL (576i). Вони містили наступні критерії: для цілей моніторингу або контролю юрби було потрібно, щоб силует людини в кадрі займав не менш 5% від висоти кадру, для детектування – 10%, для цілей розпізнавання відомого операторові людини – 50% і для ідентифікації – 100% від висоти екрана.

Стандарт роз'ясняє, що при використанні мегапіксельної камери й дисплея високого дозволу можна забезпечити ту ж розв'язну здатність кадру навіть у випадку, коли силует людини займає менший відсоток екрана.

					ВКРМ-123.24.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

Оскільки це завдання трудомістка, допомога проєктувальникові роблять калькулятори, що розраховують щільність пікселів, і спеціальне ПЗ для проєктування систем спостереження, що відображають області ідентифікації, розпізнавання, детектування й спостереження на плані з урахуванням перешкод, що обмежують зону огляду камери.

Таке ПЗ також дозволяє відобразити результати тривимірного моделювання системи й показати замовникові, що побачить кожна камера, коли система відеоспостереження буде встановлена.

Візуальне відображення різними квітами областей ідентифікації, розпізнавання й детектування людей на плані дає можливість проєктувальникові не тільки полегшити вибір оптимальних місць розташування камер, але також і спростити процес узгодження проєкту із замовником.

Проєктувальник також може завантажити моделі тестових людей у який-небудь пакет 3D-моделювання з наступним зменшенням дозволу зображень до розв'язної здатності камери перед демонстрацією операторові.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Python – динамічна інтерпретована об'єктно-орієнтована скриптова мова програмування із строгою динамічною типізацією. Офіційний сайт мови програмування Python <https://www.python.org/>. Python – багатоцільова мова програмування, яка дозволяє писати код, що добре читається. Відносний лаконізм мови Python дозволяє створити програму, яка буде набагато коротше свого аналога, написаного на іншій мові. Python – багатоплатформова мова програмування. Це означає, що програми на Python можна запускати в різних операційних системах без будь-яких змін.

Ще однією перевагою Python є його стандартна бібліотека, яка встановлюється разом з Python і містить готові інструменти для роботи з

					ВКРМ-123.24.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

операційною системою, веб-сторінками, базами даних, різними форматами даних, для побудови графічного інтерфейсу програм тощо. Програми, написані на мові програмування Python, можуть бути як невеликими скриптами, так і складними системами. Python абсолютно безкоштовний.

Швидкість виконання коду Python

Один з можливих недоліків Python – швидкість виконання коду. Python не є компільованою мовою. Код на Python спочатку компілюється у внутрішній байт-код, який потім виконується інтерпретатором Python. У більшості випадків при використанні Python виходять програми повільніші в порівнянні з такими мовами, як C.

Втім, сучасні комп'ютери мають таку обчислювальну потужність, що для більшості застосунків швидкість розробки важливіша швидкості виконання, а програми на Python зазвичай пишуться набагато швидше.

Окрім того, Python легко розширюється модулями, написаними на C або C++. Такі модулі можуть використовуватися для виконання частин програми, що створюють інтенсивне навантаження на процесор.

Використання Python

Python використовується для різних цілей: для створення ігор і веб-застосунків, розробки внутрішніх інструментів для різноманітних проектів. Мова також широко застосовується в науковій області для досліджень і розв'язування прикладних завдань.

Застосування мови програмування Python

1. BitTorrent – протокол для обміну даними.
2. Ubuntu Software Center – вільне програмне забезпечення для пошуку, установки і видалення пакунків в системі Ubuntu Linux.
3. Blender – програма для створення тривимірної комп'ютерної графіки, що включає засоби моделювання, анімації, вимальовування, пост-обробки відео, а також створення відеоігор.

					ВКРМ-123.24.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

4. GIMP – растровий графічний редактор, із підтримкою векторної графіки.
5. World of Tanks.
6. Вільна енциклопедія Вікіпедія.
7. Пошукова система Google.
8. DropBox – файловий хостинг, що включає персональне хмарне сховище, синхронізацію файлів і програму-клієнт.
9. YouTube – популярне відеосховище.
10. ...

Версії Python

Мови програмування з часом змінюються – розробники додають в них нові можливості, а також виправляють помилки. Так з’являються різні версії мови. Наприклад, код написаний на Python 2 у більшості випадків не буде працювати у версії Python 3 без внесення додаткових змін.

Процесор є найважливішим компонентом в комп’ютері. Одна з основних функцій процесора – це обробка даних згідно комп’ютерної програми, яка є списком інструкцій, шляхом виконання арифметичних і логічних операцій над фрагментами даних.

Кожна інструкція в програмі – це команда, яка «повідомляє» процесору, яку операцію він повинен виконати. Процесор комп’ютера може розуміти лише ті інструкції, які написані на машинній мові. Машинна мова – це штучна мова, створена для передачі команд комп’ютеру. За допомогою машинної мови створюються ефективні програми, оскільки розробник отримує доступ до всіх можливостей процесора. Машинна мова – мова низького рівня.

Інструкція машинної мови існує для кожної операції, яку процесор здатний виконати – є інструкція для додавання чисел, є інструкція для віднімання чисел і т.д. Увесь набір інструкцій, який центральний процесор може виконати, відомий як набір інструкцій процесора.

					ВКРМ-123.24.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

Наприклад, у вас є певна програма, яка зберігається на диску вашого комп'ютера. Для виконання програми, ви здійснюєте подвійний клік на значку програми. Це змушує програму копіюватися з диска в оперативну пам'ять, після чого процесор комп'ютера виконує копію програми, яка знаходиться в оперативній пам'яті.

Коли процесор виконує інструкції програми, він бере участь у процесі, який є відомим як цикл `fetch – decode – execute` (отримати – декодувати – виконати). Цей цикл виконується для кожної інструкції у програмі і складається з трьох кроків:

Отримати

Програма – це послідовність інструкцій на машинній мові. Першим кроком циклу є завантаження (отримання) наступної інструкції з пам'яті в процесор.

Декодувати

Інструкція машинної мови – це двійкове число, яке представляє команду, що повідомляє процесору виконати певну операцію. На цьому кроці процесор декодує інструкцію, яку було «витягнуто» з пам'яті, для визначення того, яка операція повинна виконуватись.

Виконати

Останній крок циклу – виконати операцію.

Хоча процесор комп'ютера розуміє тільки машинну мову, людині непрактично писати програми на машинній мові. Така програма може мати тисячі або навіть мільйони бінарних інструкцій, і написання такої програми буде дуже обтяжливим процесом.

З цієї причини була створена мова асемблера як альтернатива машинній мові. Замість використання двійкових чисел для написання інструкцій, мова асемблера використовує короткі слова, відомі як мнемокоди.

Незважаючи на те, що мова асемблера не вимагає двійкових інструкцій, як у випадку машинної мови, проте вона вимагає високих знань про процесор.

					ВКРМ-123.24.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

Використовуючи мову асемблера, навіть для найпростішої програми, необхідно написати велику кількість інструкцій.

Мова програмування високого рівня дозволяє створювати складні програми, не знаючи, як працює процесор, і не записуючи великої кількості інструкцій низького рівня. Крім того, більшість мов програмування високого рівня використовують слова, які легко зрозуміти.

Python – одна із популярних сучасних мов програмування високого рівня. Python – інтерпретована мова програмування. Python – це високорівнева інтерпретована мова програмування, на відміну від C++, яка є прикладом компільованої мови програмування. Назва Python відноситься як до мови програмування, так і до інтерпретатора – комп'ютерної програми, яка зчитує початковий код (написаний на Python) і виконує інструкції (команди).

Для перекладу мови високого рівня на машинну мову доступні два типи програм:

1. Компілятор.
2. Інтерпретатор.

Завантаження Python

Версії інтерпретатора Python для різних операційних систем доступні для безкоштовного завантаження за адресою <https://www.python.org/downloads>.

Середовище програмування для Python

Для написання програм використовують текстові редактори або інтегровані середовища розробки, які включають в себе різні інструменти для роботи з кодом: засіб для написання коду (текстовий редактор), інтерактивний інтерпретатор, відлагоджувач тощо.

Текстові редактори та інтегровані середовища програмування для Python:

– IDLE – стандартний редактор Python. Встановлюється разом з Python для користувачів Windows, окремим паунком для користувачів Linux.

					ВКРМ-123.24.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ док.ум.	Підпис	Дата		22

– Notepad++ – безкоштовний текстовий редактор початкового коду, який підтримує велику кількість мов, в тому числі і Python. Лише для користувачів Windows.

– Visual Studio Code – це легкий, але потужний редактор початкового коду, який розповсюджується безкоштовно і доступний у версіях для платформ Linux, Windows і macOS.

– PyScripter – інтегроване середовище розробки для мови програмування Python. Для користувачів Windows. Поширюється безкоштовно.

– Wing IDE 101 – вільне інтегроване середовище для Python, розроблене для навчання програмістів-початківців. Для користувачів Linux, Windows і macOS. Поширюється безкоштовно.

– Geany – вільний текстовий редактор з базовими елементами інтегрованого середовища розробки, доступний для операційних систем Linux, Windows і macOS.

– PyCharm – інтегроване середовище розробки для мови програмування Python. PyCharm є власницьким програмним забезпеченням. Наявна безкоштовна версія Community з усіченим набором можливостей. Для користувачів Linux, Windows і macOS.

– Thonny – IDE для вивчення програмування мовою Python. Для користувачів Linux, Windows і macOS.

– Mu – редактор коду Python для програмістів-початківців. Для користувачів Linux, Windows і macOS.

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускню кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи відеоаналітики на основі систем машинного зору.

					ВКРМ-123.24.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

В процесі розробки випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти необхідно виконати наступний обсяг роботи:

а) провести аналіз існуючих систем-аналогів для виявлення їх позитивних і негативних якостей. Результати аналізу врахувати в подальших розробках;

б) вибрати та обґрунтувати методику побудови системи контролю роботи технологічного обладнання на виробництві в автоматизованому режимі. Розробити функціональну та структурну схеми системи;

в) розробити програмне забезпечення системи, що дозволить реалізувати поставлену технічним завданням задачу. Побудувати блок-схеми алгоритмів програми та підпрограми;

г) організувати інтерфейс користувача з метою формування та виводу на екран ЕОМ повідомлень про некоректні дії користувача та нестандартні ситуації в роботі технологічного обладнання;

д) розробити рекомендації по організаційних та методичних заходах, які забезпечать впровадження системи в промислову експлуатацію та її подальшу успішну експлуатацію;

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

ж) розробити заходи по охороні праці при впровадженні та експлуатації системи, а також розробити заходи з цивільного захисту;

з) сформулювати висновки про виконаний обсяг робіт та одержані результати.

					ВКРМ-123.24.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Реалізація підтримки HEVC у різних поколіннях процесорів

Споживачі очікують від розроблювачів оптимізації мультимедіа додатків для всіх типів пристроїв, тому важливо знати про можливості сучасних поколінь процесорів Intel.

– Процесори Intel Core 4-го покоління (Haswell), 2-3,5 ГГц, 4 ядра: підтримують програмний декодер HEVC, здатний декодувати потоки HEVC 4K у реальному часі.

– Процесори Intel Core 5-го покоління (Broadwell) підтримують гібридне 8-бітне програмно-апаратне кодування HEVC.

– Процесори Intel Core 6-го покоління (Skylake) підтримують 8-бітне кодування й декодування HEVC з апаратним прискоренням.

Стиск HEVC у порівнянні зі стиском AVC (H.264) (значне зниження швидкості даних)

Алгоритм стиску HEVC використовується головним чином постачальниками рішень для обробки відео й постачальниками потокового відео, оскільки цей алгоритм дозволяє набагато краще стискати відеодані. В алгоритмі AVC використовувалися блоки 16x16 пікселів, тоді як в HEVC використовуються блоки аж до 64x64 пікселів. В H.264 підтримувалося 8 напрямків пророкування, тоді як в H.265 підтримується 35 напрямків пророкування між кадрами.

Крім того, алгоритм H.265 здатний ефективніше виявляти області кадру, що залишаються незмінними протягом декількох кадрів (наприклад, тло в сцені з діалогом). Такі області кодуються тільки в початковому кадрі, після цього кодування не повторюється до виявлення змін. Кодувальник HEVC також може змінювати розмір незмінного блоку, якщо змінюється лише частина незмінної області, або збільшувати розмір блоків, якщо в наступних кадрах незмінна

					ВКРМ-123.24.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

область стає крупніше. За рахунок цього підвищується ефективність і знижується навантаження на процесор.

Засоби розробки Intel, що підтримують HEVC

Корпорація Intel пропонує інструменти й ресурси, що допомагають розроблювачам додатків працювати із графічними процесорами. Існує два основних пакети Intel SDK, що допомагають оптимізувати програми для ГП, на додаток до Intel® Media Server Studio. Їхній вибір залежить від платформ, пристроїв і моделей роботи, для яких призначене розроблювальний додаток.

– Використовуйте Intel® Media SDK для оптимізації декодування відео й для кодування відеоматеріалів для мультимедіа додатків, що працюють на клієнтських і мобільних пристроях. У цьому пакеті задіяні апаратне прискорення й досягається найвища продуктивність безлічі самих популярних кодеків, об'єднаних в одному високорівневому API. За рахунок цього додатка будуть готові до майбутнього: код, розроблений для більше ранніх поколінь процесорів Intel, буде автоматично й найбільш оптимальним образом працювати на новому обладнанні без додаткової доробки й налаштування. Крім того, можна буде з легкістю додавати доступ до нових можливостей, які будуть підтримуватися перспективним устаткуванням.

– Використовуйте Intel® SDK для додатків OpenCL, щоб використовувати апаратне прискорення ГП загального призначення. OpenCL працює головним чином з виконуються блоками, що. Проте в процесори Intel додаються нові розширення, що дозволяють скористатися в додатках OpenCL перевагами апаратних блоків Intel з фіксованими функціями.

– Використовуйте пакет Intel® Media Server Studio Professional Edition, до складу якого входить Intel Media SDK, для швидкого перекодування мультимедіа й для оптимізації серверних додатків для обробки мультимедіа, інфраструктури зв'язку (обробка відео й відеоконференції, цифрові системи відеоспостереження), хмарних відеосистем і центрів обробки даних, а також для убудованих систем. Цей пакет включає потужні засоби оптимізації й аналізу продуктивності обробки

					ВКРМ-123.24.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

відео, а також засобу, що прискорюють перехід на HEVC, що дозволяє прискорити розробку й знизити ресурсоемність додатків.

В обох SDK для доступу до апаратних можливостей Intel використовується той самий драйвер. В Windows* всі бібліотеки, необхідні для запуску Media SDK і додатків OpenCL, автоматично встановлюються разом із графічним драйвером, тому немає необхідності впаковувати додаткові бібліотеки часу виконання разом з додатками.

Підведемо підсумки. Відео використовується повсюдно, обсяги відеоданих постійно ростуть. Обробка таких даних має найважливіше значення, оскільки багато споживачів відмовляються від спеціалізованих пристроїв, таких як DVD-програвачі, на користь потокової передачі відеоданих для фільмів і ігор. Формат кодування відео HEVC (H.265), апаратна підтримка якого вперше реалізована в процесорах Intel 6-го покоління, забезпечує найбільш ефективний стиск і дає можливість працювати з відео надвисокої чіткості, що особливо важливо у зв'язку з появою на ринку все більшого числа пристроїв, що підтримують дозвіл UHD. Щоб оптимізувати код і скористатися можливостями декодування й кодування HEVC, можна використовувати Intel Media SDK і Intel SDK для додатків OpenCL або завантажити безкоштовну ознайомлювальну версію Intel Media Server Studio.

Платформи для AI Video Analytics

Висока складність комп'ютерного зору з машинним навчанням викликає потребу в новій інфраструктурі та обчислювальних методах, таких як Edge AI. Такі розподілені концепції Edge Computing покращують надійність, масштабованість і ефективність аналізу відео за допомогою машинного навчання. Таким чином, були представлені платформи комп'ютерного бачення, які дозволяють підприємствам розробляти та постачати спеціальні програми відеоаналітики, які інтегруються з існуючими камерами та VMS.

Архітектура, керована моделями, абстрагує камери та моделі штучного інтелекту всіх типів для розробки високопродуктивних додатків із будівельними

					ВКРМ-123.24.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

блоками. Програми зору можна розгортати в масштабі на крайніх пристроях і обробляти велику кількість потоків камери в режимі реального часу.

Відеоаналітика ШІ в різних галузях

Найбільші додатки на ринку відеоаналітики включають безпеку: виявлення інцидентів, керування вторгненнями, підрахунок людей, моніторинг дорожнього руху, автоматичне розпізнавання номерних знаків (ANPR), розпізнавання облич, AR та оцінка руху еґо. Крім того, відеоаналітика була корисною для таких галузей, як виробництво, безпека, роздрібна торгівля, охорона здоров'я та готельний бізнес тощо.

AI Video Analytics у безпеці

Відеоаналітика працює над наданням рішень для спостереження та безпеки, створюючи загальні засоби ідентифікації та виявлення різних об'єктів у відеопотоках. Така технологія корисна для відстеження людей або об'єктів, які цікавлять відео, або ідентифікації та виявлення зловмисників. Використання відеоаналітики AI для цих цілей дозволяє позначати певні об'єкти та подавати сигнали про підозрілу поведінку.

Виявлення вертикального руху

Конкретним прикладом відеоаналітики для безпеки може бути система виявлення підйому на паркан. Персонал служби безпеки, як правило, навчений знати, що люди, які ходять поза огорожею, вважаються регулярними, але підніматися на вершину огорожі або боротися з нею – нерегулярно.

Програмне забезпечення відеоаналітики, навчене розпізнавати тонкі відмінності в напрямку руху між звичайною та нерегулярною поведінкою, пов'язаною з огорожею, можна пов'язати з відео в реальному часі з камер спостереження.

Якби хтось почав лазити по паркану, програмне забезпечення розпізнало б вертикальний рух як ненормальне явище та створило б певний сигнал тривоги. Для порівняння, якби хтось проходив поряд із парканом, він створював би

					ВКРМ-123.24.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

горизонтальний рух, який не класифікується системою виявлення як підозріла діяльність.

Існує кілька програм відеоаналітики в різних варіантах. Наприклад, виявлення людини з відстеженням об'єктів можна використовувати для виявлення людини, яка піднімається на паркан у зоні огляду. У цій програмі можливості відеоаналітики базуються на інтегрованих алгоритмах виявлення об'єктів, які працюють безпосередньо на пристрої, а не на зовнішньому сервері, щоб виконувати виявлення в реальному часі (граничні обчислення).

Різноманітність правил можна запускати одночасно та надсилати сповіщення безпосередньо з камери за допомогою текстового повідомлення, електронної пошти або до системи керування відео.

Класифікація відеооб'єктів III

Класифікація об'єктів відеоканалу передбачає виявлення небезпечних об'єктів у прямому ефірі з камери чи заданому відео. Невеликі відмінності між об'єктами, які іноді навіть важко побачити охоронцям перед камерою, можуть бути виявлені програмами відеоаналітики, навченими знаходити незначні відмінності, які можуть відрізнити небезпечний об'єкт від безпечного.

Наприклад, рентгенівська перевірка безпеки може використовувати програми відеоаналітики, навчені класифікувати об'єкти на подачах багажу в режимі реального часу під час реєстрації безпеки, щоб ідентифікувати конкретні об'єкти, що представляють інтерес, наприклад гострі інструменти чи зброю. Така технологія вже впроваджена в усьому світі, оскільки її точність зростає. Управління транспортної безпеки (TSA) запровадило сканери комп'ютерної томографії (КТ) із найсучаснішою 3-D технологією на контрольно-пропускних пунктах аеропортів США (перегляньте більше додатків відеоаналітики в авіації).

Відеотехнологія штучного інтелекту зараз удосконалюється, щоб підвищити точність виявлення об'єктів у відеокадрах у широкому діапазоні реальних додатків комп'ютерного бачення.

					ВКРМ-123.24.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

Відстеження поведінки відео AI

Подібно до виявлення руху, розглянутого у прикладі огорожі, інші типи поведінки також є відповідними підставами для класифікації відеоаналітики. Наприклад, відстеження поведінки включає поведінку людини по відношенню як до себе, так і до більших об'єктів, наприклад транспортних засобів, і до того, що це означає для безпеки загальної території. Нижче наведено два менших приклади відстеження поведінки, реалізованого у відеоаналітиці.

– **Виявлення тиняння:** у розумних містах відеоаналітика навчена помічати, коли люди чи транспортні засоби залишаються у визначеній зоні довше, ніж дозволяє визначений користувачем час. Для безпеки території, залежно від уподобань виконавця програми, може бути активована сигналізація. Така поведінка ефективна для сповіщення в режимі реального часу про підозрілу поведінку в аптечних відділах, банкоматах, наркодиспансерах та інших місцях.

– **Виявлення зупиненого транспортного засобу:** ця частина відеоаналітики корисна для запобігання простою або зупинці транспортних засобів у недозволеному місці протягом тривалого часу. Виявляються транспортні засоби, які зупинилися поблизу чутливої зони довше, ніж це дозволяє користувач. Така поведінка ідеально підходить для того, щоб зупинити транспортні засоби, які перешкоджають завантажувально-приймальним докам, забезпечити дотримання правил паркування та зменшити час очікування транспортних засобів біля паркувальних служб або воріт паркування. Зупинені транспортні засоби на дорозі, що рухається, також можуть вказувати на незареєстровані аварії або проблеми з транспортними засобами, і така технологія може попередити відповідні органи про такі випадки.

– **Саботаж камери:** розширене виявлення втрати відео може розпізнавати, коли пряме відеопотік було зламано або підроблено. Наприклад, якщо вандал фарбує або закриває об'єктів або тягнеться, щоб відсунути фіксовану камеру від наміченої сцени, спрацьовує тривога.

					ВКРМ-123.24.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

AI Video Analytics в роздрібній торгівлі

Індустрія роздрібної торгівлі може запровадити аналітику AI для відеопотоків у багатьох ситуаціях. Ці компоненти управління роздрібною торгівлею допомагають оптимізувати операції та створити кращий досвід клієнтів без збільшення людської відповідальності чи додаткових операційних витрат, пов'язаних із дорогим обладнанням. Ознайомтеся з нашою великою статтею про візуальний штучний інтелект у роздрібній торгівлі.

Інтелектуальне керування чергами

Відеоаналітика надає інформацію про кращу політику для кас і навіть може налаштувати магазини на можливості без каси. Це дозволяє магазинам здійснювати самоконтроль і дотримуватись кодексу честі, не боячись крадіжки чи інших нечесних порушень. Керування чергами також може надати інформацію про те, що працює, а що ні, щоб керувати розміром черг у магазинах. Під час пандемії, наприклад, керування чергами може бути важливим для запобігання поширенню.

Підрахунок людей

Підрахунок людей можна проводити за допомогою відеоаналітики. Роздрібна торгівля передбачає багато експериментів із дисплеями та маркетинговими стратегіями. Спостерігати або мати доступ до того, скільки клієнтів приходить і коли, корисно для магазинів, щоб знати, що працює з точки зору маркетингу та огляду продукту.

Крім того, помічати, скільки клієнтів проводять тривалий час біля вітрин, корисно для магазину, оскільки це покращує взаємодію з клієнтами та покращує бізнес магазину. З точки зору підрахунку людей, відеоаналітика надає оперативну інформацію та інформацію про бренд, а також розкриває безліч інших аспектів відносин із клієнтами.

AI Video Analytics в охороні здоров'я

Заклади охорони здоров'я завжди надавали пріоритет сучасним технологіям для оптимізації витрат і забезпечення безпеки своєї практики,

					ВКРМ-123.24.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

оскільки охорона здоров'я як галузь контролюється суворим державним і корпоративним законодавством. Реалізації відеоаналітики AI можуть бути корисними в охороні здоров'я для психічного здоров'я, точності діагностики та спостереження за літніми чи молодими пацієнтами в лікарнях. Ознайомтеся з іншими відеопрограмами ШІ для охорони здоров'я.

Моніторинг пацієнтів на дому

Технологія спостереження робить спостереження за літніми пацієнтами в будинках престарілих можливим і зручним для опікунів. Падіння є основною причиною травм і смерті людей похилого віку, тому моніторинг вдома корисний для виявлення незвичних положень або періодів, коли людина знаходиться на підлозі або є недієздатною.

Персональні медичні пристрої можуть ефективно виявляти падіння, але для ефективності їх потрібно носити постійно. Відеоаналітика для виявлення падінь штучним інтелектом забезпечує більш вільне використання рук і може бути модифікована, щоб не просто виявляти падіння. Наприклад, така система може також визначити, чи літня людина прийняла певні ліки, коли мала.

Аналіз психічного здоров'я

Поєднання розширеної відеоаналітики та машинного навчання з аналізом обличчя та досвідом клініцистів-людей може покращити здатність медичного працівника робити правильний висновок про стан психічного здоров'я пацієнта. Помітний підхід включає аналіз емоцій обличчя за допомогою ШІ. Відеоаналітику можна навчити точно виявляти відмінності в нормальній і ненормальній поведінці обличчя чи фізичній поведінці.

Медичні працівники часто записують ці невербальні комунікації як частину свого прогнозу, але досить суб'єктивно і лише якщо вони їх помічають. Відеоаналітика в програмах для психічного здоров'я гарантує, що тонкі натяки в поведінці пацієнта не залишаться непоміченими.

					ВКРМ-123.24.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

Біотехнологія

Ранній скринінг хвороботворних мікроорганізмів харчового походження є ключем до забезпечення безпеки харчових продуктів. В даний час досліджуються біосенсори, спрямовані на виявлення сальмонели за допомогою обробки відео на смартфоні та флуоресцентного маркування. Відеоаналітика також може аналізувати живі корми бактерій і ідентифікувати одні бактерії від інших, що робить її корисною для визначення відмінностей у складі бактерій.

AI Video Analytics в розумних містах

Аналіз відео в режимі реального часу з алгоритмами глибокого навчання має відомі приклади використання в розумних містах. Прочитайте нашу статтю про сучасний список найкращих і найцінніших програм комп'ютерного зору в розумних містах.

Кілька компаній, які займаються відеоаналітикою, намагаються розробити більш інтегровані рішення, пов'язані з містами. На viso.ai надайте технології бачення містам і постачальникам державних послуг по всьому світу, від Гренландії до Швейцарії, Сполучених Штатів і Австралії. Різноманітні розподілені камери різних типів можна інтегрувати, щоб забезпечувати міським операторам постійний зворотний зв'язок для прийняття обґрунтованих рішень.

Міські агенції можуть посилити залучення громадян і оптимізувати роботу за допомогою аналізу даних у режимі реального часу та співпраці між відомствами. З економічної точки зору сприяння розумним містам стимулює нові потоки доходів і економічний розвиток шляхом підвищення активності клієнтів і їх поведінки.

Відеоаналітика корисна для міст, які керують натовпом людей і є частиною моделі розумного міста. Автоматичне розпізнавання номерних знаків і моніторинг дорожнього руху є двома прикладами відеоаналітики, яка використовується в містах. Ці програми оптимізують громіздкі процеси, які вимагають достатнього втручання людини.

					ВКРМ-123.24.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		33

Розпізнавання номерних знаків транспортних засобів

Автоматичне розпізнавання номерних знаків (ANPR) складається з точних систем, здатних зчитувати номерні знаки автомобіля без втручання людини. Використання високошвидкісного захоплення зображення з підтримкою освітлення дає можливість системам відеоаналітики виявляти та зчитувати номерні знаки майже в реальному часі.

Таким чином, символи номерних знаків розпізнаються за допомогою оптичного розпізнавання символів (OCR), перетворюючи зображення на рядки цифрового тексту. Це дає змогу відеоаналітичним системам виявляти та записувати номерні знаки. Сучасні програми ANPR створюють набори метаданих для кожного виявленого номерного знака для повторного використання органами в інших системах. ANPR корисний для запису автомобілів, що проїжджають на червоне світло, дорожніх аварій тощо.

Інтелектуальний моніторинг дорожнього руху

Відеоаналітика може надати інформацію, корисну для аналізу дорожнього руху та моніторингу заторів. На додаток до виявлення небезпечних аварій і ситуацій, моніторинг дорожнього руху дає кількісну інформацію про кількість транспортних засобів у певних областях у певний час і моделі руху.

У разі аварії ці аналітичні системи, що включають аналіз дорожнього руху, пізніше надають поліції допомогу для збору доказів у разі судового розгляду.

Підрахунок транспортних засобів

Цей аспект відеоаналітики передбачає розрізнення автомобілів, вантажівок, автобусів і таксі для створення корисної статистики, яка використовується для отримання інформації про трафік. Мережеві камери можуть фіксувати концентрацію автомобілів, що швидко рухаються, в одному районі порівняно з іншим, що може бути корисним для міста, щоб знати, які впровадження контролю руху є ефективними. Підрахунок транспортних засобів також дає зрозуміти, коли в майбутньому потрібно проводити ремонт доріг.

					ВКРМ-123.24.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

3.2 Розробка структурної схеми

Нове покоління IP-камер з розв'язною здатністю HD і вище, високою частотою зміни кадрів і швидкісних інтерфейсів висуває серйозні вимоги до систем передачі й обробки даних. А для рішення завдань машинного зору, таких як комп'ютерна графіка (у кіноіндустрії), відеоспостереження, медичні дослідження або операції й інші, необхідно працювати з незжатими даними, одержуваними безпосередньо від оптичного сенсора. Як правило, для обробки такого обсягу інформації необхідні значні обчислювальні ресурси, можливі й інші підходи.

У деяких випадках проблеми передачі й прийому більших обсягів даних вирішуються шляхом локалізації обробки відео й балансування навантаження, а зберігання більших обсягів різномірних даних здійснюється за допомогою спеціалізованих гібридних сховищ, що підтримують роботу із зображеннями й відео, структурованими й метаданими.

При проектуванні систем безпеки доводиться враховувати значний перелік погроз і діючі нормативні документи. Відеомоніторинг припускає як роботу з архівом, так і оперативне відеоспостереження.

При проектуванні таких систем необхідно погодити з усіма користувачами об'єкта план розміщення відеокамер і зони огляду, скласти графік робіт, передбачити технічне обслуговування систем, провести аналіз умов їхнього функціонування протягом тривалого строку експлуатації, забезпечити відстеження строків впровадження інженерної інфраструктури, а також виділити місце для установки активного встаткування. Найчастіше на великих транспортних об'єктах згодом монтуються конструкції, що загороджують огляд, і ймовірність такої події варто враховувати при проектуванні будь-якої відеосистеми.

Звичайно за допомогою системи відеорозпізнавання вдається оперативно розпізнати скупчення людей, а також ідентифікувати й зафіксувати факти їхнього

					ВКРМ-123.24.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		35

неадекватного руху (бійки, падіння, хаотичний рух, різке прискорення, перетинання забороненої зони), виявити в контрольованій зоні залишені або зниклі предмети, виявити задимлення або відкритий вогонь.

При цьому важливо зрозуміти, які завдання потрібно вирішити замовникові, щоб допомогти йому визначити реальні цілі, оскільки можливості відеодетекторів звичайно переоцінюються. Найчастіше корисно проводити техніко-економічне обґрунтування впровадження кожної підсистеми. Неодмінна умова – пілотні випробування системи в умовах реального застосування.

При виборі системи відеоаналітики варто керуватися умовами її використання, видом застосовуваного обчислювача й типом самої відеоаналітики. Функції обчислювача, «мозку» системи, можуть бути передані камері, винесеним промисловим блокам або серверам. Кожний з варіантів має свої переваги й недоліки. Більше дешеві рішення, як правило, функціонально слабкіше.

У реальних умовах відеоаналітика нерідко забезпечує зовсім не той результат, що обіцяє виробник. Для її ефективного застосування дуже важливий ракурс зйомки, а також стабільність тла й висвітлення – поява тіней і засвітлення. Залежно від цих умов різні детектори й алгоритми поведуться по-різному.

Відеоаналітика звичайно ґрунтується на вирахуванні тла, детектуванні руху й наступної класифікації об'єкта (у цьому випадку важливо правильне застосування масштабної сітки) або на алгоритмах машинного зору з використанням навчених класифікаторів (формування ознак, що класифікують). Обидва варіанти мають свої недоліки. Перший більше розповсюджений – таке рішення дешевше. Алгоритми машинного зору працюють точніше, але для них потрібно більше обчислювальних ресурсів.

Цікаві перспективи відкриваються у випадку застосування камер машинного зору для класифікації й підрахунку відвідувачів у роздрібній торгівлі. Їх можна використовувати для оптимізації продажів на основі збирається статистики, що, – розпізнавання й підрахунку відвідувачів, їхньої класифікації й трекінгу, побудови «теплових карт».

					ВКРМ-123.24.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		36

Ще одна область застосування камер машинного зору – розпізнавання осіб (див. рис. 3.1).

Завдання розпізнавання осіб можна вирішити за допомогою обробки незжатого відео на локальній відеокарті, змонтованої безпосередньо на камері. У цьому випадку зображення не передається в мережу – на сервер відправляються тільки статистичні дані

Відеоідентифікація може виконуватися оператором або в автоматичному режимі, коли використовується біометричне розпізнавання. В обох випадках потрібні створення зон контролю, розміщення серверів і АРМ для оперативного реагування.

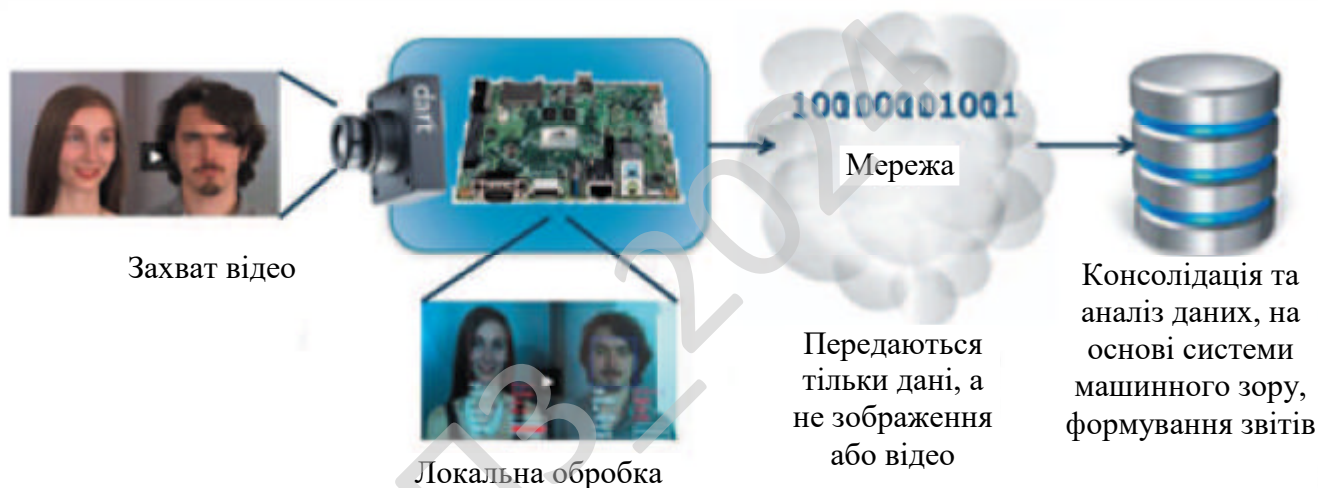


Рисунок 3.1 – Структурна схема системи

На результат відеоідентифікації впливає якість зображень у пошуковій базі даних і «картинок на вході» (у свою чергу, воно залежить від можливостей пристроїв і умов реєстрації), а також самі алгоритми (різні алгоритми висувають різні вимоги до характеристик використовуваних зображень). Нерідко заявлені виробником параметри отримані на основі бази даних з високоякісними зображеннями, і реальні результати роботи системи можуть сильно відрізнятись від очікуваних. Тому важливо, щоб база даних була наповнена якісними зображеннями.

перевищують 0,000011 і 0,016%. За вартістю такі системи зовсім незабаром будуть порівнянні із системами розпізнавання відбитка пальця.

В Україні ринок біометрії перебуває в зародковому стані (його обсяг не більше 50-60 млн доларів). Однак, згідно із прогнозами, у найближчій перспективі середньорічні темпи росту даного сегмента складуть 35%, і до 2020 року його оберт збільшиться до 500 млн доларів. Основні драйвери росту – держсектор, банківський сектор, електронна комерція, СКУД.

Камери з інтелектом

У децентралізованих системах відеоспостереження «інтелектом» оснащується сама камера. Зовнішнього сервера для керування не потрібно, оскільки камера підтримує необхідні функції налаштування й відеоаналітики. Економія на серверах, мережній інфраструктурі й програмних ліцензіях дозволяє знизити вартість рішення в цілому.

IP-камери для відеоспостереження на транспорті з розв'язною здатністю 6 Мпкс і кутом огляду 1800 виготовляються з армованого скловолокна й споживають усього 4 Вт. У числі убудованих аналітичних функцій – підрахунок об'єктів, фіксація перетинання заданого коридору, виявлення нестандартного поведіння, статистика по об'єктах.

Значна частина обробки відео виконується самою камерою, тому вимоги до каналів передачі даних знижуються. До того ж відео обсягом до 256 Гбайт можна зберігати на SD-карті. Для контролю швидко, що рухаються об'єктів, замість стандартного кодеку H.264 можуть знадобитися спеціальні кодеки. IP-камери з оглядом 1800 успішно застосовуються й у торгівлі.

Популярним рішенням стають «здвоєні» пристрої, у яких камера денного відеоспостереження доповнена тепловізором. Така камера може сполучати два зображення й оснащуватися убудованим датчиком руху. Вона здатна виявити порушника навіть у повній темряві. Інше можливе застосування – контроль температури об'єкта.

					ВКРМ-123.24.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		39

Ще одне швидко, що розвивається напрямок, – використання камер у дронах, такі дрони можуть застосовуватися як для захисту об'єктів, так і для нападу. Не виключено, що через кілька років нас очікує війна дронів.

Поряд з підвищенням «рівня інтелекту» IP-камер, актуальною тенденцією стало збільшення їхнього дозволу. У минулому році цілий ряд виробників систем відеоспостереження випустили IP-камери з розв'язною здатністю 4K (Ultra HD). Де можуть застосовуватися такі продукти? Один з додатків – відеоспостереження на стадіонах, коли потрібно контролювати більші площі й тисячі глядачів.

Основні труднощі при організації системи відеоспостереження на стадіоні пов'язані з більшою кількістю камер відеоспостереження, зберіганням величезного масиву даних, значними витратами на ПЗ.

Системи з розв'язною здатністю 4K можуть стати ефективним рішенням для великих об'єктів, забезпечити спостереження за значним скупченням людей і великих просторів. За допомогою відео-камери 4K можна вести широкий огляд і розрізнити навіть самі дрібні деталі. Використання всього однієї відеокамери означає зменшення витрат на установку, економію на обслуговуванні системи й ліцензуванні ПЗ.

Однак все більша деталізація зображення спричиняє ріст вимог до швидкості передачі даних (бітрейту). Наприклад, при стиску кодеком H.264 бітрейт камери з розв'язною здатністю 8 Мп@25 кадрах/с дорівнює 16 Мбіт/с. Рішення проблеми – ефективний стиск відео за допомогою кодеку H.265. Стандарт H.265, прийнятий ITU/ISO в 2013 році, забезпечує зворотну сумісність із H.264. Кодек H.265 на 50% ефективніше H.264, і при однаковому бітрейті якість відео з використанням H.265 вище, ніж для H.264. До того ж навантаження на канали передачі даних нижче; додатково можна заощадити на ємності систем зберігання.

Таким чином, спільне використання відеокамер 4K и стиск H.265 дозволить без значних витрат ефективно впровадити відеоспостереження надвисокої чіткості в існуючу систему безпеки. І цілком імовірно, що через два-

					ВКРМ-123.24.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

три роки деякі системи відеоспостереження вже будуть переведені на 4K з кодеком H.265.

Відеоспостереження й хмари

Для рішення завдань відеоаналітики можна використовувати й хмарні сервіси. Сервіс консолідує метадані, що збираються із установлених у замовника IP-камер Smartec Neuro і лічильників, а потім здійснює їхню обробку засобами відеоаналітики VCA Technology. З його допомогою можна генерувати різноманітні звіти (статистику й графіки відвідувань за зазначений період), будувати теплові карти й навіть переглядати «живе відео», передане з торговельної точки.

Послуги відеоспостереження по сервісній моделі відмінно підходять для малого й середнього бізнесу, до 15% малих і середніх підприємств готові скористатися послугами хмарного відеоспостереження, хоча в 85% таких клієнтів ті або інші рішення вже встановлені. І число потенційних замовників із сегмента СМБ росте. Причин декілька: часті переїзди з офісу в офіс у зв'язку зі зміною ставок оренди, небажання нести капітальні витрати на власне відеоспостереження, відсутність підготовлених фахівців для монтажу й експлуатації своєї системи відеоспостереження.

Останнім часом усе більше компаній звертають увагу не тільки на охоронну функцію відеоспостереження – вони починають сприймати відеоспостереження як додатковий інструмент для здійснення своєї діяльності й контролю за станом бізнесу. Поступово стали мінятися й потреби: з автономного програмно-апаратного комплексу відеоспостереження перетворюється в хмарну послугу, що забезпечує доступ до перегляду зображення з відеокамер з будь-якої точки миру, що надає можливості для рольового доступу до такого перегляду й до архівних записів, а крім того, що гарантує схоронність архівних записів.

Основні споживачі сервісу відеоспостереження – це малі підприємства зі сфери обслуговування: юридичні компанії, салони краси, будівельні організації, кафе й ресторани, магазини. Відеоспостереження в цьому випадку не стільки виконує охоронну функцію, скільки стає ефективним інструментом для бізнесу.

					ВКРМ-123.24.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41

Камери, звичайно, допомагають підсилити міри безпеки, але разом з тим забезпечують спостереження за торговельним залом і касовою зоною, приміщенням для прийому відвідувачів, зоною розвантаження товару, роботою персоналу – тобто дозволяють стежити за дотриманням трудової дисципліни й рівнем надаваного сервісу.

Установка камер над касами й у торговельних приміщеннях дозволяє виявляти позаштатні ситуації й здійснювати їхній розбір. На складі й у виробничих цехах можна стежити за переміщеннями товарів і зайнятістю робітників. За допомогою камер зручно спостерігати за всією територією підприємства, навіть якщо це кілька об'єктів, розташованих у різних місцях.

Таким чином, постачальникам послуг відеоспостереження вдається запропонувати малим підприємствам привабливі рішення для тих проблем, з якими сьогодні вони зіштовхуються найбільше часто. Однак перешкоди залишаються: це висока вартість зберігання відео й значне навантаження на канали передачі даних у випадку потокового відео HD-Якості. Трохи знизити вартість зберігання можна за допомогою дискових масивів спеціальної конфігурації, оптимізованих під відеоархіви.

Послуга оператора припускає не тільки обслуговування, але й установку встаткування. А наявність бібліотеки проектів, використання типових елементів для швидкого конструювання проекту й відпрацьована практика мережних обстежень спрощують розгортання системи.

Згодом у користувачів з'являються потреби в додаткових послугах, де потрібна складна відеоаналітика, – наприклад, для організації пасивної охорони території (віртуальний периметр, проникнення в контрольовану зону, повідомлення про рух у кадрі за розкладом) або при застосуванні відеоаналітики як маркетинговий інструмент, що дозволяє управляти продажами (лічильник відвідувачів, теплові й кінетичні карти, контроль порожніх прилавків, відкриття й закриття точок продажів).

У результаті розвитку подібних сервісів відеоспостереження й відеоаналітика стають сучасними й ефективними інструментами для бізнесу.

					ВКРМ-123.24.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

Установка камер сприяє підвищенню рівня безпеки в торговельному й іншому приміщеннях, а також дозволяє підсилити контроль за роботою співробітників і поліпшити обслуговування.

3.3 Розробка функціональної схеми

На рисунку 3.2 зображена функціональна схема системи. Нижче розглянемо її більш докладно. Цифрова система відеоспостереження, що розроблена:

- миттєвий пошук і перегляд відеозапису по камері, даті й часу;
- можливість інтеграції з іншими комп'ютерними системами безпеки;
- легка й недорога трансляція відеоархівів по каналах зв'язку (Інтернет та ін.);
- можливість відправлення тривожних повідомлень по електронній пошті й SMS;
- забезпечує високу якість відтвореного відеозапису;
- високу швидкість доступу до відеоархіву;
- можливість цифрового збільшення й масштабування будь-якого кадру;
- можливість експорту відеоінформації на сумісні зовнішні носії.

Цифрові системи відеоспостереження виконують наступні функції:

- Перегляд зображення по локальній мережі. Інтерфейс віддаленого робочого місця нічим не відрізняється від інтерфейсу на відеосервері. Завдяки чому доступні всі функції по керуванню системою, що й на сервері, включаючи перегляд відеоархіву й звуковий супровід. Існує відмінність як зображення відео на віддаленому робочому місці від зображення на відеосервері. Так, як по лініях зв'язку передається стисле зображення, якість відображення цілком залежить від величини стиску відеосигналу, установленої по кожній камері на відеосервері. Віддалених робочих місць може бути трохи: використовується так званий мережний режим. Такий режим може бути використаний на об'єктах з декількома

					ВКРМ-123.24.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		43

постами охорони, з наявністю постів начальника охорони, адміністратора системи безпеки; для забезпечення можливості спостереження за переміщенням співробітників начальницькому составу; а також для аналізу архіву відеозображення віддалено по локальній мережі не займаючи робоче місце співробітників оперативної служби.

– Керування поворотними пристроями відеокамер. Програмне забезпечення має можливість підключення поворотних пристроїв відеокамер і керування ними прямо з робочого місця.

– Обробка зовнішніх датчиків. Крім функцій відеоспостереження може виконувати контролювати різні датчики (охоронної й пожежної сигналізації), і управляти зовнішніми виконавчими пристроями (сиреною, голосовим оповіщувачем або іншим пристроєм).

– Джойстик керування системою відеоспостереження. Маніпулятор для керування системою заміняє мишу й клавіатуру. При експлуатації цифрових відеосерверів нерідко виникають проблемами втручання оператора в роботу системи: зміна налаштувань операційної системи, налаштувань програми відеоспостереження, установки ігор і сторонніх додатків. Рішенням даної проблеми є заміна стандартних пристроїв введення (клавіатури й миші) на спеціалізований маніпулятор. Даний маніпулятор дозволяє працювати тільки із програмним забезпеченням.

– Моніторинг. Можливе підключення до 32-х камер, відображення всіх на одному екрані. Можливість поділу камер на два монітори. Використання режиму overlay дозволяє виводити зображення на екран монітора з аналоговою якістю. Можливість цифрового збільшення в 2, 4, 8 і 16 разів. Функція голосового супроводу подій (при підключених аудіоколонках).

– Запис. Стиск зображення здійснюється за алгоритмом Motion-Wavelet (розмір кадру залежить від дозволу, кольоровості камери, ступеня стиску й деталізації зображення) дозволяє досягти значень бітрейта в межах від 2 до 80 Кб. Режими запису: по детекції руху, по команді, постійно, передзапис і дозапис по кожній камері.

					ВКРМ-123.24.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		44

– Детекція руху. Відстеження наявності об'єктів, що рухаються. Основні переваги: висока перешкодозахищеність, емпіричні налаштування за розміром й контрастністю зображення, конфігурування області детекції руху незалежно для кожної камери, візуальне виділення контурів об'єктів, що рухаються, перерозподіл ресурсів для тривожної камери.



Рисунок 3.2 – Функціональна схема системи

– Відтворення. Одна камера в один момент часу, перегляд відеоархіву одночасно із записом, індексація відеоархіву при запуску системи для швидкого пошуку, пошук у відеоархіві по даті й часу, відображення щільності запису за добу, покадрове програвання вперед та назад, збільшення швидкості програвання до 25 Fps, утиліта зовнішнього декодування й перегляду відеозаписів зроблених системою, можливість збереження шматків відеоархіву в avi, що б мати можливість перегляду на будь-якому комп'ютері.

– Убудований WEB server. Надає можливість контролювати й управляти відеокамерами, переглядати відеоархів, а також здійснювати постановку й зняття з охорони з віддаленого комп'ютера через web-браузер.

– Аудіореєстрація. Дозволяє вводити в систему аналоговий сигнал із зовнішніх джерел: мікрофонів, телефонних каналів і інших. Аудіореєстрація може вестися по незалежних каналах, а може й синхронно. У випадку налаштування аудіореєстрації разом з відеозображенням, архів від цих камер записується спільно зі звуком. Аудіореєстрація необхідна у випадках недостатчості інформації, що надають відеокамери. Більше твердий контроль за підлеглими й персоналом. Найбільш часті сфери й місця застосування аудіоконтролю – кімнати переговорів і нарад, місця для паління, спостереження за будинком під час відсутності хазяїна, спостереження за обслуговуючим персоналом, (покоївкою, гувернанткою й іншою прислугою), запис телефонних переговорів секретаря. Для реалізації необхідного ефекту від аудіоконтролю й поліпшення розбірливості записаного архіву рекомендується застосування якісного встаткування прослуховування.

Розглянувши усі блоки функціональної схеми перейдемо до розгляду діаграми взаємодії процесів, які відбуваються у системі.

3.4 Розробка діаграми процесів

Діаграма взаємодії процесів використовується для візуалізації процесів обробки даних (структурне проектування). Для розробника вважається звичним спочатку креслити діаграму взаємодії процесів даних рівня контексту, завдяки чому буде показано взаємодію системи.

Ця діаграма в подальшому підлягає уточненню шляхом деталізації процесів та потоків даних з метою показати систему що розробляється. Діаграма процесів розробленої системи зображена на рисунку 3.3. При детальному її розгляді можна побачити як саме проходить взаємодія у розробленій системі.

Використовується модель проектування, графічне представлення «потоків» даних в інформаційній системі.



Рисунок 3.3 – Діаграма взаємодії процесів

Діаграми потоків даних містять чотири типи елементів:

- Процеси які являють собою трансформацію даних в рамках описуваної системи.
- Сховища даних (репозиторії).
- Зовнішні по відношенню до системи сутності.
- Потоки даних між елементами трьох попередніх типів.

Таким чином, розглянувши опис системи, структурну, функціональну схеми системи, та діаграму взаємодії процесів перейдемо до опису блок-схем основної програми, та підпрограм, які використовуються, для реалізації системи.

4 РЕАЛІЗАЦІЯ ПРОЕКТУ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ПРАВИЛЬНІСТЬ ПРОЕКТНИХ РІШЕНЬ

4.1 Блок-схеми та опис алгоритмів функціонування системи

Блок-схеми є основою ПЗ. Тому від точності і детальності проробки блок-схеми залежить результат всієї програми.

При виборі початкової точки відліку при побудові схем було враховано, що виходячи з вибору мови програмування і інших технічних засобів, програма буде об'єктно-орієнтована що вимагає оптимізації, також те, що при розробці програми слід надати особливу увагу модулю відеоаналітики на основі систем машинного зору.

Функціональні блоки на схемі позначають прямокутниками, всередині яких надписують їх найменування відповідно до функцій, що виконуються. Зв'язки між функціональними блоками (внутрішні впливи) позначаються лініями зі стрілками, які вказують напрям впливів.

Функціональні блоки можуть виконуватися в укрупненому і розгорненому вигляді. У першому випадку на схемі зображають найважливіші блоки системи і зв'язки між ними.

На рисунку 4.1 зображена основна блок-схема програми, на рисунку 4.2 зображено роботу підпрограми.

З яких видно що робота основної програми складається з початкових етапів ініціалізації ПЗ, перевірки наявності ресурсів системи, блоку початку основного циклу з чеканням запиту від користувача в якому відбувається виклик підпрограми та останньої стадії – перевірки поточного стану та поверненням на початок схеми чи з завершенням роботи розробленого ПЗ.

					ВКРМ-123.24.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

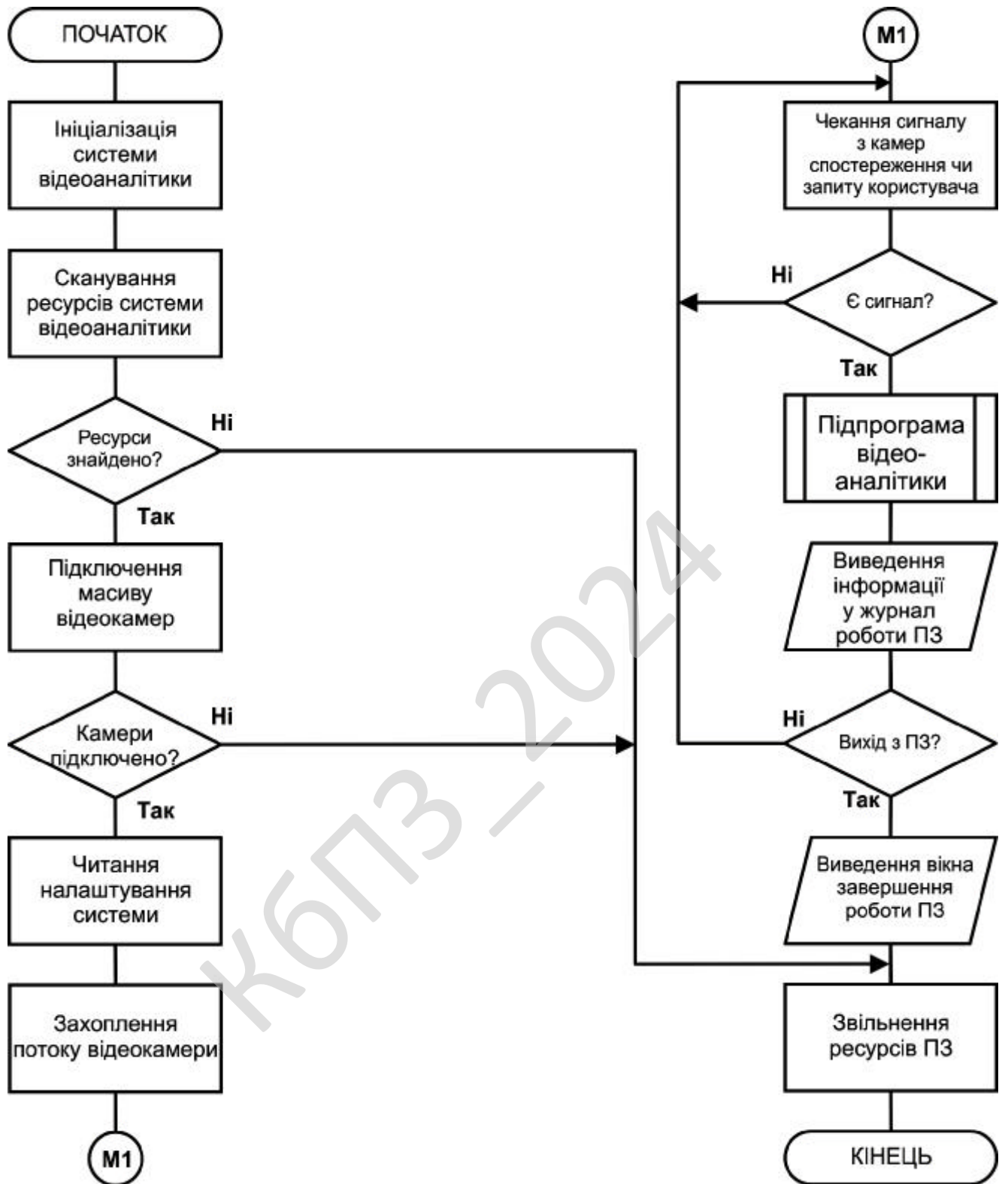


Рисунок 4.1 – Блок-схема основної програми

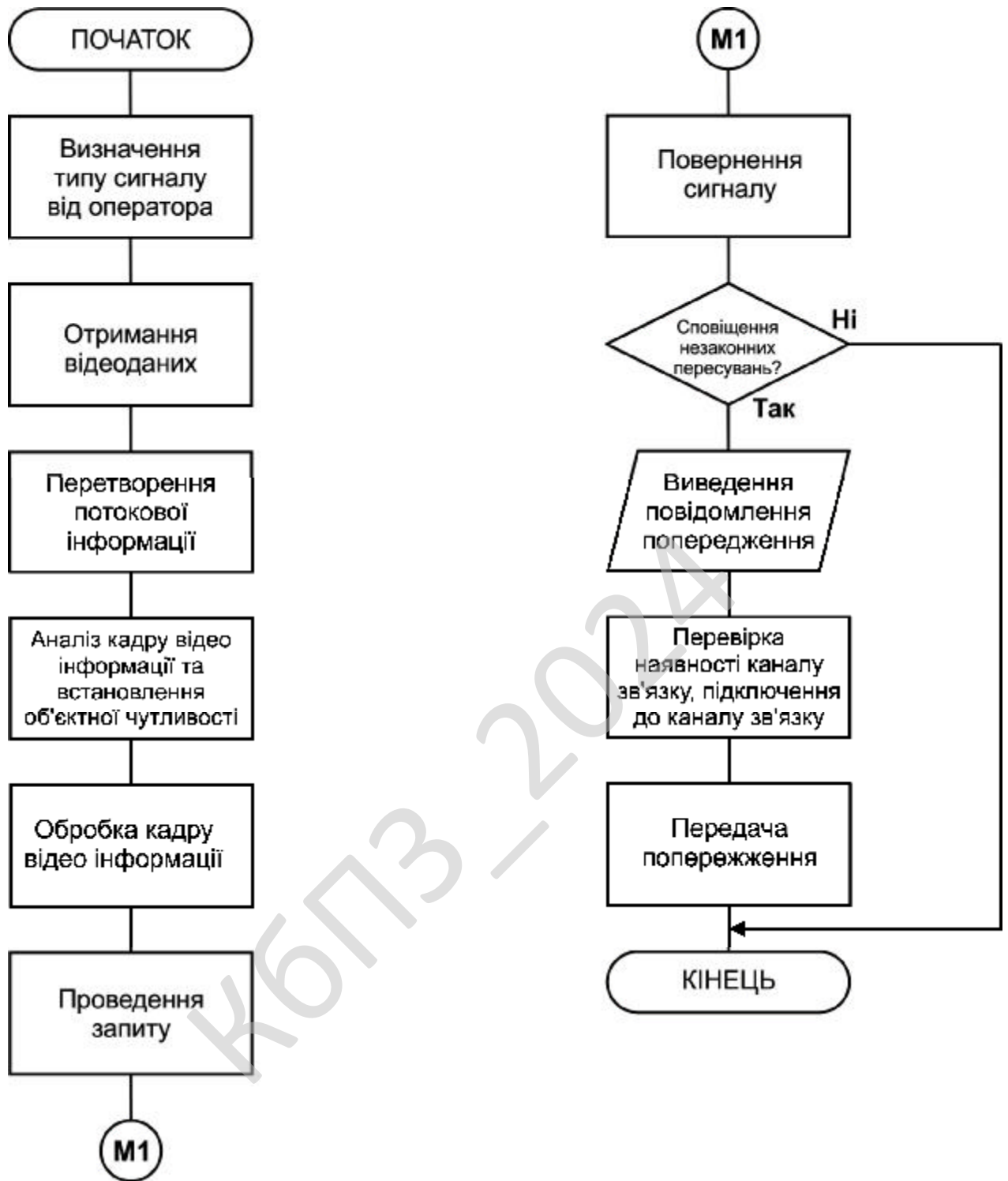


Рисунок 4.2 – Блок-схема роботи підпрограми

При роботі підпрограми виконується основний функціонал системи з циклічними послідовностями, перевіркою поточного стану та поверненням в основну програму прапорів стану виконання.


```

def detect_objects(frame):
    contours, _ = cv2.findContours(frame, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)
    return contours

# Основний модуль обробки відеопотоку
def process_video(source=0):
    cap = video_capture(source)

    if cap is None:
        return

    ret, prev_frame = cap.read()
    if not ret:
        print("Не вдалося отримати перший кадр")
        return

    prev_frame = preprocess_frame(prev_frame)

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

        current_frame = preprocess_frame(frame)
        motion_mask = motion_detection(prev_frame, current_frame)
        contours = detect_objects(motion_mask)

        for contour in contours:
            if cv2.contourArea(contour) > 500:
x, y, w, h = cv2.boundingRect(contour)
cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)

        cv2.imshow("Motion Detection", frame)
        prev_frame = current_frame

        if cv2.waitKey(30) & 0xFF == ord('q'):
            break

    cap.release()
    cv2.destroyAllWindows()

# Виклик основного модуля для обробки відео з веб-камери

```

					ВКРМ-123.24.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		52

- Система відстеження помилок.
- Діаграми Ганта та календар.
- Ведення новин проекту, документів та управління файлами.
- Сповіщення про зміни за допомогою RSS-потоків та електронної пошти.
- Власна Wiki для кожного проекту.
- Форуми для кожного проекту.
- Облік часових витрат.
- Налаштування власних (custom) полів для задач, затрат часу, проектів та користувачів.
- Легка інтеграція із системами керування версіями (SVN, CVS, Git, Mercurial, Vazaar и Darcs).
- Створення записів про помилки на основі отриманих листів
- Підтримка LDAP автентифікації;
- Можливість самореєстрації нових користувачів;
- Багатомовний інтерфейс (у тому числі українська мова);
- Підтримка СКБД: MySQL, PostgreSQL, SQLite.

Діаграма Ганта (*Gantt chart*, також стрічкова діаграма, графік Ганта) – це популярний тип діаграм, який використовується для ілюстрації плану, графіка робіт за будь-яким проектом. Є одним з методів планування та управління проектами.

Діаграма Ганта являє собою відрізки (графічні плашки), розміщені на горизонтальній шкалі часу. Кожен відрізок відповідає окремому завданню або підзадачі. Завдання і підзадачі, складові плану, розміщуються по вертикалі. Початок, кінець і довжина відрізка на шкалі часу відповідають початку, кінцю і тривалості завдання. На деяких діаграмах Ганта також показується залежність між завданнями.

Діаграма може використовуватися для представлення поточного стану виконання робіт: частина прямокутника, що відповідає завданню,

					ВКРМ-123.24.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54

заштриховується, відзначаючи відсоток виконання завдання; показується вертикальна лінія, що відповідає моменту «сьогодні».

Часто діаграма Ганта використовується спільно з таблицею зі списком робіт, рядки якої відповідають окремо взятій задачі, зображеній на діаграмі, а стовпці містять додаткову інформацію про задачу.

Система відстеження помилок Багтрекер – прикладна програма для допомоги розробникам програмного забезпечення (програмістам, тестувальникам тощо) враховувати і контролювати помилки, знайдені у програмах, питання щодо функціональності, рішення та оновлення, побажання користувачів, а також стежити за процесом їх виконання.

Кожному, хто розробляв програмні продукти, добре знайоме співвідношення «20/80» – останні 20 % роботи тривають 80 % часу.

Як це не парадоксально, але нічого дивного в цій пропорції немає, адже саме на завершальній стадії починається тестування проекту, коли виявляються помилки, і що більший проект, то більше буде знайдено помилок.

Водночас досить часто виявляється, що більшість цих помилок були відомі та могли бути виправлені з меншими витратами на попередніх стадіях роботи, але не були вчасно описані, а потім загубилися серед інших важливих завдань.

Отже, система відстеження помилок у найпростішому варіанті – це процес, що включає в себе виявлення помилки, її опис, виправлення і перевірку цього виправлення, тобто процес «стеження» за багом протягом всього як його життєвого циклу, так і життєвого циклу розробки в цілому.

Сукупність інформації про дефект. Головний компонент такої системи – база даних, що містить відомості про виявлені дефекти. Ці відомості можуть включати в себе:

- номер (ідентифікатор) дефекту;
- хто повідомив про дефект;
- дата і час виявлення дефекту;
- версія продукту, в якій виявлено дефект;

					ВКРМ-123.24.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		55

- серйозність (критичність) дефекту та пріоритет рішення;
- опис кроків для відтворення дефекту (неправильної поведінки програми);
- відповідальний за усунення дефекту;
- обговорення можливих рішень та їх наслідків;
- поточний стан виправлення дефекту;
- версії продукту, в якій дефект виправлений.

Крім того, розвинені системи надають можливість прикріплювати файли, які допомагають описати проблему, наприклад, дамп пам'яті або скріншот.

Використання. Основна перевага систем відстеження помилок полягає в забезпеченні чітких централізованих оглядів, запитів на розробку (включаючи помилки і виправлення) та їх стан. У корпоративному середовищі, системи відстеження помилок можуть бути використані для генерації звітів по продуктивності програмістів виправлення помилок. Однак, це може іноді приводити до неточних результатів, тому що різні помилки можуть мати різні ступені пріоритету та серйозності, що пов'язано з складністю їх фіксації.

Життєвий цикл дефекту. Як правило, система відстеження помилок використовує той чи інший варіант «життєвого циклу» помилки, стадія якого визначається поточним станом помилки.

Типовий життєвий цикл дефекту:

1. Новий – дефект зареєстрований тестувальником.
2. Призначений – призначений відповідальний за виправлення дефекту.
3. Дозволений – дефект переходить назад у сферу відповідальності тестувальника.

Як правило, супроводжується резолюцією, наприклад:

- Виправлено (виправлення включені у версію таку-то).
- Дубль (повторює дефект, що вже знаходиться в роботі).
- Не виправлено (працює відповідно до специфікації, має занадто низький пріоритет, виправлення відкладено до наступної версії тощо).
- «В мене все працює» (запит додаткової інформації про умови, в яких дефект проявляється).

					ВКРМ-123.24.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

4. Далі тестувальник проводить перевірку виправлення, залежно від чого дефект або знову переходить у стан «Призначений» (якщо він описаний як виправлений, але не виправлений), або у стан «Закрито».

5. Відкрито повторно – дефект знайдено знову в іншій версії.

Система може надавати адміністраторові можливість налаштування користувачі, які можуть переглядати і редагувати помилки залежно від їх стану, переводити їх в інший стан або видаляти.

У корпоративному середовищі, система відстеження помилок може використовуватися для отримання звітів, що показують продуктивність програмістів при виправленні помилок. Однак, часто такий підхід не дає достатньо точних результатів через те, що різні помилки мають різну ступінь серйозності та складності. При цьому серйозність проблеми прямо не стосується складності її усунення.

Незважаючи на те що я працював над ПЗ один в реалізації програми я використовував підходи пришвидшення розробки на основі методологій Agile.

Гнучка розробка програмного забезпечення (Agile software development, agile-методи) – клас методологій розробки програмного забезпечення, що базується на ітеративній розробці, в якій вимоги та розв'язки еволюціонують через співпрацю між самоорганізовуваними багатофункціональними командами.

Гнучка розробка – найкращий засіб для підвищення продуктивності розробників програмного забезпечення.

Більшість гнучких методологій націлені на мінімізацію ризиків, шляхом зведення розробки до серії коротких циклів, що мають назву ітерацій, які зазвичай тривають один-два тижні. Кожна ітерація сама по собі виглядає як програмний проект в мініатюрі, і включає всі завдання, необхідні для видачі мінімального приросту за функціональністю: планування, аналіз вимог, проектування, кодування, тестування і документування. Хоча окрема ітерація, як правило, недостатня для випуску нової версії продукту, мається на увазі те, що гнучкий

					ВКРМ-123.24.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		57

- вітання змін вимог навіть наприкінці розробки (це може підвищити конкурентоспроможність отриманого продукту);
- часта поставка робочого програмного забезпечення (кожен місяць або тиждень або ще частіше);
- тісне, щоденне спілкування замовника з розробниками впродовж всього проекту;
- проектом займаються мотивовані особистості, які забезпечені потрібними умовами роботи, підтримкою і довірою;
- рекомендований метод передачі інформації – особиста розмова (віч-на-віч);
- робоче програмне забезпечення – найкращий вимірник прогресу;
- спонсори, розробники та користувачі повинні мати можливість підтримувати постійний темп на невизначений термін;
- постійну увагу поліпшенню технічної майстерності та зручному дизайну;
- простота – мистецтво не робити зайвої роботи;
- найкращі технічні вимоги, дизайн та архітектура виходять у самоорганізованої команди;
- постійна адаптація до мінливих обставин.

Маніфест та Принципи гнучкої розробки містять високорівневі ідеї щодо того, як потрібно вибудовувати процес розробки програмного забезпечення, щоб успішно завершувати проекти й створювати команди, в яких приємно та цікаво працювати.

Документи визначають, що потрібно для цього зробити, але не говорять, як це зробити. По-іншому й не могло бути, оскільки Маніфест та Принципи народилися внаслідок консенсусу представників різних (хоча й споріднених) напрямів, які могли знайти спільну основу лише на рівні базових цінностей та принципів.

					ВКРМ-123.24.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		59

Критика. Багато керівників проектів, що працюють у традиційних методологіях на кшталт «водоспаду», критикують agile-методи.

Один з повторюваних пунктів критики: при agile-підході часто нехтують створенням «дорожньої карти» розвитку продукту, так само як і управлінням вимогами, в процесі якого і формується така «карта». Гнучкий підхід до управління вимогами не має на увазі далекосяжних планів (по суті, управління вимогами просто не існує в даній методології), а має на увазі можливість замовника раптом і несподівано наприкінці кожної ітерації виставляти нові вимоги, що часто суперечать архітектурі вже створеного і поставленого продукту. Таке іноді призводить до катастрофічних «авралів» з масовим рефакторингом і переробками практично на кожній черговій ітерації.

Крім того вважається, що робота в agile мотивує розробників вирішувати всі прибулі завдання найпростішим і найшвидшим можливим способом, при цьому часто не звертаючи уваги на коректність коду з точки зору вимог базової платформи (підхід «працює, та й добре», при цьому не враховується, що може перестати працювати при найменшій зміні або ж породити важкі до відтворення дефекти після реального розгортання у клієнта). Це призводить до зниження якості продукту і накопиченню дефектів.

Методології. Існують методології, які дотримуються цінностей і принципів заявлених в Agile Manifesto, деякі з них:

1. Agile Modeling – набір понять, принципів і прийомів (практик), що дозволяють швидко і просто виконувати моделювання і документування в проектах розробки програмного забезпечення. Не включає в себе детальну інструкцію з проектування, не містить описів, як будувати діаграми на UML.

Основна мета – ефективне моделювання і документування; але не охоплює програмування та тестування, не включає питання управління проектом, розгортання і супроводу системи. Однак включає в себе перевірку моделі кодом.

2. Agile Unified Process (AUP) спрощена версія IBM Rational Unified Process (RUP), розроблена Скоттом Амблером, яка описує просте і зрозуміле наближення (модель) для створення програмного забезпечення для бізнес-додатків.

					ВКРМ-123.24.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		60

3 Agile Data Method – група ітеративних методів розробки програмного забезпечення, в яких вимоги та рішення досягаються в рамках співпраці різних крос-функціональних команд.

4. DSDM заснований на концепції швидкої розробки додатків (Rapid Application Development, RAD). Являє собою ітеративний і інкрементний підхід, який надає особливого значення тривалій участі в процесі користувача/споживача.

5. Essential Unified Process (EssUP).

6. Екстремальне програмування (Extreme programming, XP).

7. Feature driven development (FDD) – функціонально-орієнтована розробка. Використовуване в FDD поняття функції або властивості (feature) Системи досить близько до поняття прецеденту використання, використовуваному в RUP, істотна відмінність – це додаткове обмеження: «кожна функція повинна допускати реалізацію не більше, ніж за два тижні». Тобто якщо сценарій використання досить малий, його можна вважати функцією. Якщо ж великий, то його треба розбити на декілька відносно незалежних функцій.

8. Getting Real – ітераційний підхід без функціональних специфікацій, що використовується для веб-додатків. У даному методі спершу розробляється інтерфейс програми, а потім її функціональна частина.

9. OpenUP – це ітераційно-інкрементний метод розробки програмного забезпечення. Позиціюється, як легкий і гнучкий варіант RUP. OpenUP ділить життєвий цикл проекту на чотири фази: початкова фаза, фази уточнення, конструювання та передачі. Життєвий цикл проекту забезпечує надання зацікавленим особам та членам колективу точок ознайомлення і прийняття рішень впродовж усього проекту. Це дозволяє ефективно контролювати ситуацію і вчасно приймати рішення про задовільність результатів. План проекту визначає життєвий цикл, а кінцевим результатом є остаточний додаток.

10. Scrum встановлює правила керування процесом розробки та дозволяє використовувати вже існуючі практики кодування, коректуючи вимоги або вносячи тактичні зміни. Використання цієї методології дає можливість виявляти і

усувати відхилення від бажаного результату на більш ранніх етапах розробки програмного продукту.

11. Бережлива розробка програмного забезпечення (lean software development). Використовує підходи з концепції бережливого виробництва.

4.2 Захист розробленого програмного забезпечення

Для захисту розробленого програмного забезпечення запропоновано використовувати алгоритм SHACAL-1, який заснований на перетвореннях алгоритму SHA-1. SHACAL-1 шифрує 160-бітний блок даних з використанням 512-бітного ключа шифрування. Допускається використання більше коротких ключів шифрування (не коротше 128 біт), які перед виконанням розширення ключа (процедура розширення ключа також успадкована від SHA-1 і буде описана нижче) повинні бути доповнені нульовими бітами для досягнення 512-бітного розміру. В алгоритмі SHACAL-1 передбачено 80 раундів шифрування. Шифруєме повідомлення представляється у вигляді п'яти 32-бітних субблоків A , B , C , D і E , над якими в кожному раунді виконуються наступні дії:

$$A_{i+1} = K_i + (A_i \lll 5) + f_i(B_i, C_i, D_i) + E_i + M_i,$$

$$B_{i+1} = A_i,$$

$$C_{i+1} = B_i \lll 30,$$

$$D_{i+1} = C_i,$$

$$E_{i+1} = D_i,$$

де i – номер раунду ($i = 0 \dots 79$),

K_i – фрагмент розширеного ключа для i -го раунду,

f_i – функція для i -го раунду (див. нижче),

\lll – операція побітового циклічного зрушення вліво,

M_i – константи, що модифікують, певні в такий спосіб:

Раунди	Значення константи
0...19	5A827999
20...39	6ED9EBA1
40...59	8F1BBCDC

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Розглянемо розроблене ПЗ відеоаналітики на основі систем машинного зору яке зображено на рисунку 5.1. Обчислення виконується через консольний інструмент з подальшою передачею результатів до інтерфейсу. З рисунку можна побачити що інтерфейс головного вікна розподілено на наступні функціональні розділи:

- Навігаційне меню: Файл; Фільтри; Дані; Камери; Налаштування; Вікна; Довідка.
- Блоку обрання камери.
- Розділу виведення результату роботи системи – відеопотік.
- Навігаційного меню яке викликається натисканням правої клавіші маніпулятора миші та функціональних кнопок ПЗ.

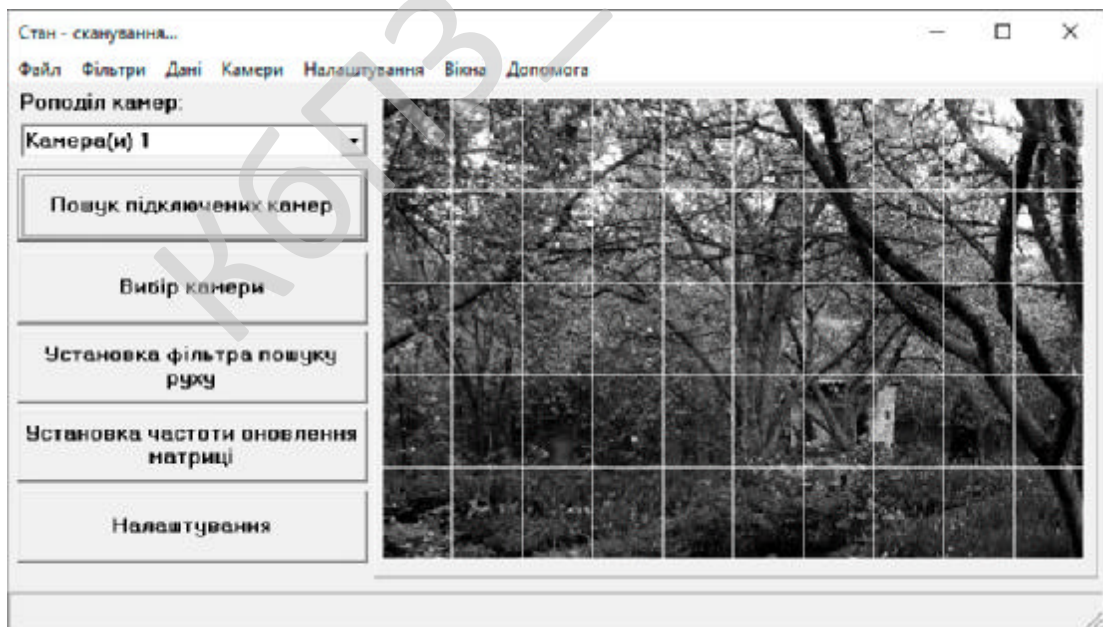


Рисунок 5.1 – Головне вікно ПЗ

Розроблена програма має дуже простий і інтуїтивно зрозумілий інтерфейс з користувачем. Кожен, хто в достатньому обсязі володіє операційним середовищем Windows без особливих складностей освоїть і цю програму, оскільки її інтерфейс інтуїтивно зрозумілий.

Якщо програма не видала ніяких помилок, і працює, то можна використовувати, інакше слід слідувати інструкціям, які пропонує програма.

На рисунку 5.2 зображено авторські дані розробленого програмного забезпечення.

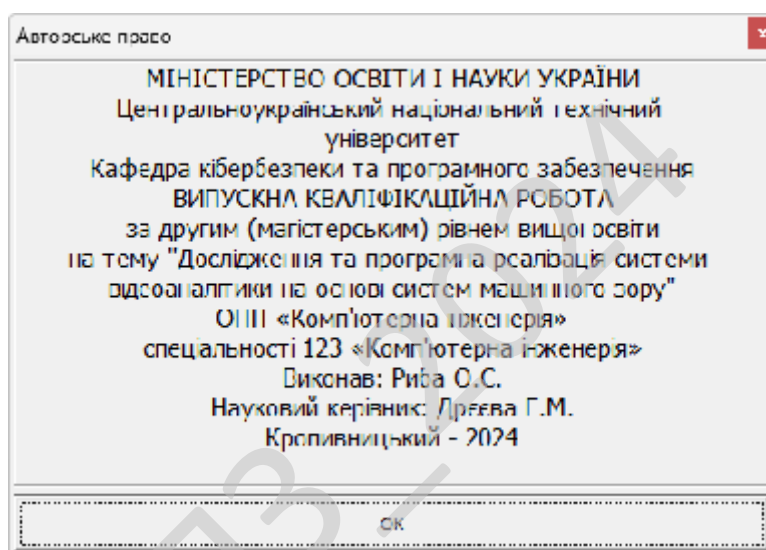


Рисунок 5.2 – Авторське право

Розглянемо процес впровадження програмного забезпечення, це процес налаштування програмного забезпечення під певні умови використання, а також навчання користувачів роботі з програмним продуктом. Впровадження програмного забезпечення це усі дії, що роблять розроблену програмну систему готовою до використання. Даний процес є частинною життєвого циклу програмного забезпечення.

Загалом процес розгортання складається з кількох взаємопов'язаних дій із можливими переходами між ними. Ця активність може відбуватися як з боку виробника так і з боку споживача. Оскільки кожна програмна система є

					ВКРМ-123.24.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		65

унікальною, то усі процеси та процедури під час розгортання важко передбачити. Тому, "розгортання" можна трактувати як загальний процес відповідно до певних вимог та характеристик. Розгортання може здійснюватись програмістом і в процесі розробки програмного забезпечення.

До діяльностей пов'язаних із розгортанням програмного забезпечення відносять:

- Випуск.
- Встановлення та активація.
- Деактивація.
- Адаптація.
- Обновлення.
- Вмонтування.
- Відстежування версій.
- Видалення.
- Вилучення з обігу.

При впровадженні програмного забезпечення потрібно урахувати наступні дії:

– Виділення критичних, з точки зору загального результату, процедур в діяльності організації. Коли набір таких процедур визначений, необхідно в першу чергу використовувати ІТ рішення для автоматизації операцій усередині саме цих процедур. Таким чином, розроблене ІТ рішення автоматично стає життєво важливим і затребуваним для організації, а також буде забезпечена публічність процесу впровадження.

– Розширення нормативної бази організації шляхом включення до неї регламентів, що описують порядок виконання процедур автоматизованих процесів. В іншому випадку є небезпека виникнення неузгодженості між автоматизованими процедурами та іншими процесами організації.

– Виконання робіт з загальної стандартизації існуючої діяльності організації, коли виділяються кращі практики виконання процедур і включаються

					ВКРМ-123.24.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		66

в IT рішення за принципом найбільшої корисності для більшості учасників. Відсоток таких процедур щодо загального обсягу автоматизації може бути невеликий, але це надає процесу побудови рішення вагу в організації за рахунок збільшення його необхідності.

Під час роботи над програмою було проведено тестування програмного забезпечення, тобто технічне дослідження, призначене для виявлення інформації про якість продукту відносно контексту, в якому воно має використовуватись.

Тестування включає як процес пошуку помилок або інших дефектів, так і випробування програмних складових з метою їх оцінки.

Проводилась оцінка:

- відповідності поставленим вимогам;
- правильна відповідь для усіх можливих вхідних даних;
- виконання функцій за прийнятний час;
- практичність;
- сумісність з ОС та стороннім ПЗ.

Оскільки число можливих тестів для програмних компонент практично нескінченне, тому стратегія тестування полягала в тому, щоб провести всі можливі тести з урахуванням наявного часу та ресурсів.

Як результат ПЗ тестувалось стандартним виконанням програми з метою виявлення помилок або інших дефектів.

Обрано умови розповсюдження – commercial software.

Програмне забезпечення, створене комерційною організацією з метою отримання прибутку від його використання іншими, наприклад, шляхом продажу копій.

Найважливішою особливістю комерційних програмних продуктів є підтримка великих компаній, прямо зацікавлених у поширенні програм. Багато організацій надають виключно платну підтримку своїх продуктів, такий підхід, як правило, використовують організації надають відкриті вихідні коди. Для продуктів, що розповсюджуються на комерційній основі діють зазвичай

					ВКРМ-123.24.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		67

безкоштовні служби підтримки, покликані збільшити рівень довіри у клієнтів і потенційних покупців.

Далеко не завжди, але як правило терміни критично важливих змін в комерційних продуктах значно менше, ніж у некомерційних проектів. Це пов'язано з тим, що над комерційним продуктом працюють цілі групи розробників і ця робота є їх основним заняттям. Розробникам-початківцям як правило доводиться шукати додаткові способи заробітку, і це збільшує час, що витрачається на доповнення і зміни програм. Так як основним рушійним фактором створення комерційного ПЗ є одержання прибутку, то комерційні програмні продукти першими заповнюють вільні ніші та пропонують варіанти вирішення завдань відразу по мірі виявлення вакууму в будь-якому секторі ринку.

Окремий вид комерційних програм, коли їх розробка оплачується безпосередньо замовником. Такі програми найчастіше позбавлені всіх переваг комерційних продуктів, оскільки мають обмежений бюджет, але більш адаптовані до вимог замовника, ніж аналоги.

КБПЗ-2024

					VKPM-123.24.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		68

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи відеоаналітики на основі систем машинного зору.

Метою розробки є дослідження та програмна реалізація системи відеоаналітики на основі систем машинного зору.

Об'єктом дослідження є процес відеоаналітики на основі систем машинного зору.

Предметом дослідження є методи відеоаналітики на основі систем машинного зору.

Методи дослідження базуються на методах теорії розпізнання образів, методах математичної статистики, методах розробки програмного забезпечення.

Наукова новизна отриманих результатів. У процесі рішення завдань, обумовлених цілями дослідження, отримані наступні результати:

- Удосконалено метод відеоаналітики на основі систем машинного зору.
- Розроблено вітчизняний продукт відеоаналітики на основі систем машинного зору, який має більш широкі можливості, на відміну від існуючих аналогів.

					VKPM-123.24.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		69

7 МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ ІТ-ПРОЄКТУ

7.1 Визначення цільової аудиторії кінцевого готового продукту

Результати дослідження та програмної реалізації системи відеоаналітики на основі машинного зору можуть зацікавити різні галузі та організації, де необхідний автоматичний моніторинг, аналітика або підвищена ефективність обробки відеоданих (рисунок 7.1).

Інноваційні рішення у відеоаналітиці на основі машинного зору можуть допомогти різним секторам підвищити продуктивність, безпеку, оптимізувати процеси та знижувати витрати на моніторинг та обробку відеоданих.

7.2 Оцінка привабливості шляхом застосування методів експертних оцінок

Для оцінки привабливості проєкту програмної реалізації системи відеоаналітики на основі систем машинного зору можна використати метод експертних оцінок, де група експертів оцінює різні критерії проєкту. Цей метод дозволяє отримати зважену оцінку від спеціалістів, які мають досвід у цій галузі.

Для системи відеоаналітики обираємо наступні критерії (рис. 7.2).

Кожному критерію призначається вага: інноваційність – 0.2, попит на ринку – 0.25, технічна здійсненність – 0.15, масштабованість – 0.1, економічна ефективність – 0.2, ризики – 0.1.

Кожен експерт оцінює кожен критерій за шкалою від 1 до 10, де 1 – найгірший показник, 10 – найкращий. Результати зводимо в таблицю 7.1.

					ВКРМ-123.24.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

1. безпека та правопорядок	охоронні агентства та правоохоронні органи для моніторингу в громадських місцях, на об'єктах критичної інфраструктури чи приватних територіях, система може виявляти підряду поведінку, несанкціонований доступ та інші порушення. системи контролю доступу; використання для розпізнавання обличчя і контролю доступу в будівлі чи закриті об'єкти.
2. транспорт та логістика	автомобільні дороги та інфраструктура: відеоналітика дозволяє контролювати потоки транспорту, виявляти порушення правил дорожнього руху, прогнозувати затери. авіація та порти: виявлення аномальної поведінки, моніторинг багажу, а також автоматизація процесів контролю безпеки.
3. роздрібна торгівля та маркетинг	роздрібні магазини та торговельні мережі: аналіз поведінки покупців, контроль черг, оптимізація викладки товарів та підвищення якості обслуговування. маркетингові агенції: використання відеоналітики для вивчення інтересів клієнтів, зон активності у магазинах чи торговельних центрах, що допомагає краще розробляти рекламні кампанії.
4. виробництво та промисловість	автоматизація виробництва: для контролю за процесами на лінії виробництва, виявлення браку або помилок у виробництві, що підвищує якість продукції та знижує витрати. системи охорони праці: контроль за дотриманням техніки безпеки та виявлення порушень у реальному часі для мінімізації ризиків.
5. агробізнес	моніторинг тваринництва та рослинництва: використання відеоналітики для контролю за станом посівів, моніторингу здоров'я тварин, відстеження їхньої поведінки та умов проживання
6. міські системи та "розумне місто"	інфраструктура "розумного міста": виявлення небезпек, контроль графіку, підвищення безпеки в громадських місцях, а також контроль за станом інфраструктури (наприклад, вуличного освітлення чи стану доріг). житлово-комунальні послуги: відеоналітика може допомагати у моніторингу прибудинкових територій, оптимізації процесів утримання території, а також виявлення порушень правил утримання об'єктів.
7. освіта та медицина	навчальні заклади: безпека учнів і студентів, аналіз відвідуваності, моніторинг зон для підвищення комфорту. медичні установи: спостереження за пацієнтами для запобігання нещасним випадкам, моніторинг медперсоналу для забезпечення контролю інфекційної безпеки.
8. фінансові установи	банки та кредитні організації: моніторинг приміщень для забезпечення безпеки, а також аналіз поведінки клієнтів у відділеннях для підвищення якості обслуговування.

Рисунок 7.1 – Цільова аудиторія

Підсумкова оцінка для кожного критерію розраховується як добуток середньої оцінки на вагу. Сумуючи результати для всіх критеріїв, отримуємо загальну оцінку привабливості (таблиця 7.2).

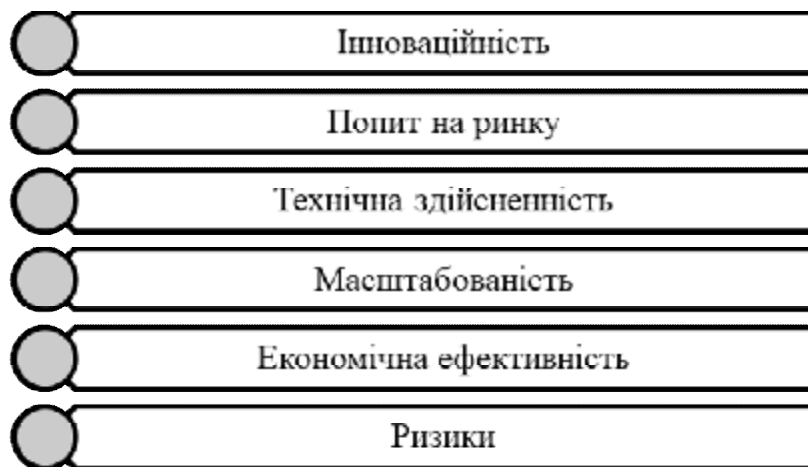


Рисунок 7.2 – Критерії оцінювання

Таблиця 7.1 – Зведені результати експертних оцінок

Критерій	Експерт 1	Експерт 2	Експерт 3	Середня оцінка
Інноваційність	8	9	7	8.0
Попит на ринку	7	8	8	7.67
Технічна здійсненність	9	8	8	8.33
Масштабованість	6	7	6	6.33
Економічна ефективність	8	9	8	8.33
Ризики	5	6	6	5.67

Таблиця 7.2 – Загальна оцінка привабливості по критеріям

Критерій	Середня оцінка	Вага	Результат
Інноваційність	8.0	0.2	1.6
Попит на ринку	7.67	0.25	1.92
Технічна здійсненність	8.33	0.15	1.25
Масштабованість	6.33	0.1	0.63
Економічна ефективність	8.33	0.2	1.67
Ризики	5.67	0.1	0.57

Підсумкова оцінка привабливості = 1.6 + 1.92 + 1.25 + 0.63 + 1.67 + 0.57 = 7.64 Загальна оцінка 7.64 вказує на високу привабливість проєкту. Однак, результати можуть вказати на окремі аспекти для вдосконалення (наприклад, масштабованість та ризику).

7.3 Вибір методу оцінки вартості ПЗ

Вибір методу оцінки вартості для програмної реалізації системи відеоаналітики на основі машинного зору залежить від ряду факторів, таких як тип проєкту, доступність даних, цілі оцінки та стадія розробки. Для системи відеоаналітики на основі машинного зору оптимальним може бути поєднання кількох методів (рис.7.3).

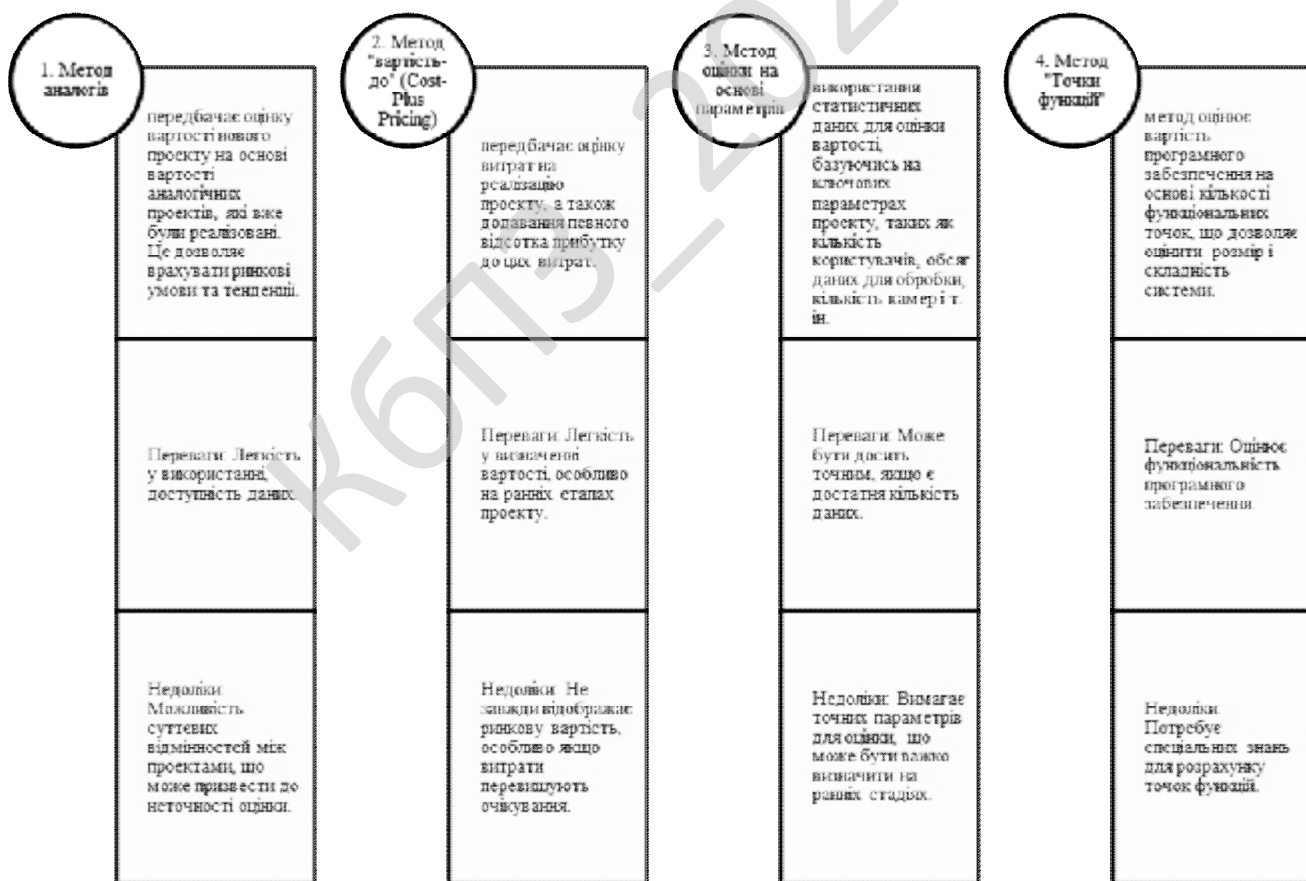


Рисунок 7.3 – Методи оцінювання проєкту

Продовження таблиці 7.3

3. Очікувані вигоди

Зниження втрат: Завдяки моніторингу та аналітиці виявлено та зменшено кількість випадків крадіжок на 30%. Якщо річні втрати від крадіжок становили \$100,000, то зниження складе \$30,000.

Підвищення ефективності обслуговування: Оптимізація розташування товарів та контролю черг призвела до зростання середнього чека на 10%. Якщо річний обсяг продажів становить \$1,000,000, то додатковий дохід складе \$100,000.

Зменшення витрат на персонал: Завдяки автоматизації контролю витрати на охорону зменшилися на \$15,000 на рік.

Загальні очікувані вигоди за перший рік: \$30,000 (зниження втрат) + \$100,000 (зростання продажів) + \$15,000 (економія на охороні) = \$145,000

4. Економічна ефективність

Чистий прибуток від впровадження: \$145,000 (вигоди) - \$87,000 (витрати) = \$58,000

Рентабельність інвестицій (ROI): $ROI = \frac{\text{Чистий прибуток}}{\text{Витрати}} \times 100\%$
 $ROI = \frac{58,000}{87,000} \times 100\% \approx 66.67$

Впровадження системи відеоаналітики на основі машинного зору в роздрібній торгівлі може призвести до значного економічного ефекту, зокрема підвищення рентабельності, зниження витрат і покращення обслуговування клієнтів. У цьому прикладі, завдяки системі, магазин може отримати чистий прибуток у \$58,000 за перший рік при ROI близько 66.67%. Цей підрахунок демонструє, як технологічні рішення можуть сприяти підвищенню економічної ефективності підприємств.

7.5 Пропозиція алгоритму просування проєкту розробки ПЗ

Алгоритм просування проєкту програмної реалізації системи відеоаналітики на основі систем машинного зору повинен включати кілька етапів, які забезпечать ефективне впровадження та досягнення максимального комерційного успіху (рис. 7.4).

Цей алгоритм допоможе вам систематично та ефективно просувати проєкт програмної реалізації системи відеоаналітики на основі машинного зору, з урахуванням стратегічних, маркетингових та технологічних аспектів, що є важливими для забезпечення успіху проєкту.

7.6 Оптимізація каналів збуту та шляхів реалізації ПЗ

Оптимізація каналів збуту та шляхів реалізації проєкту програмної реалізації системи відеоаналітики на основі систем машинного зору є ключовими для максимізації дохідності та успіху проєкту. Ось кілька стратегій, які можна застосувати для ефективної реалізації:

- визначення цільових сегментів ринку;
- інтеграція з існуючими системами;
- мультिकанальність збуту;
- підвищення цінності продукту через сервіс;
- залучення та утримання клієнтів;
- розширення каналу збуту через онлайн-маркетинг;
- підвищення ефективності через аналітику;
- пошук інвестицій та фінансування.

					ВКРМ-123.24.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		76

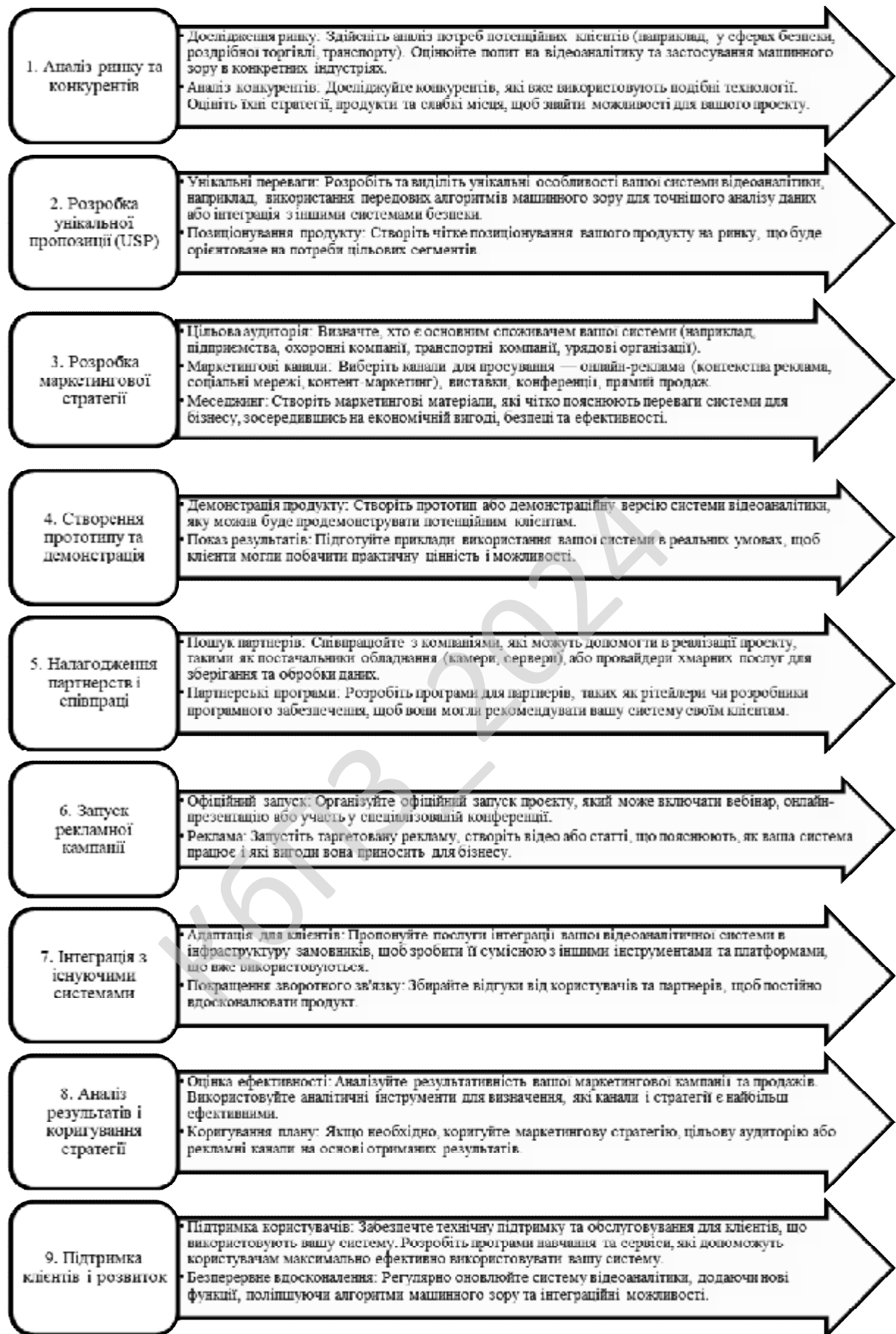


Рисунок 7.5 – Алгоритм для просування проєкту

Оптимізація каналів збуту та шляхів реалізації проекту системи відеоаналітики на основі машинного зору потребує комбінування традиційних і новітніх стратегій. Це включає в себе виявлення правильних цільових груп, інтеграцію з іншими системами, партнерство з дистриб'юторами та впровадження сучасних маркетингових підходів. Окрім цього, важливим є забезпечення постійного сервісу і підтримки для збільшення лояльності та утримання клієнтів.

7.7 Визначення ключових факторів успіху конкретного проекту

Ключові фактори успіху проекту програмної реалізації системи відеоаналітики на основі систем машинного зору включають кілька критичних аспектів, які сприяють успішному впровадженню та досягненню бізнес-цілей (рисунок 7.6).

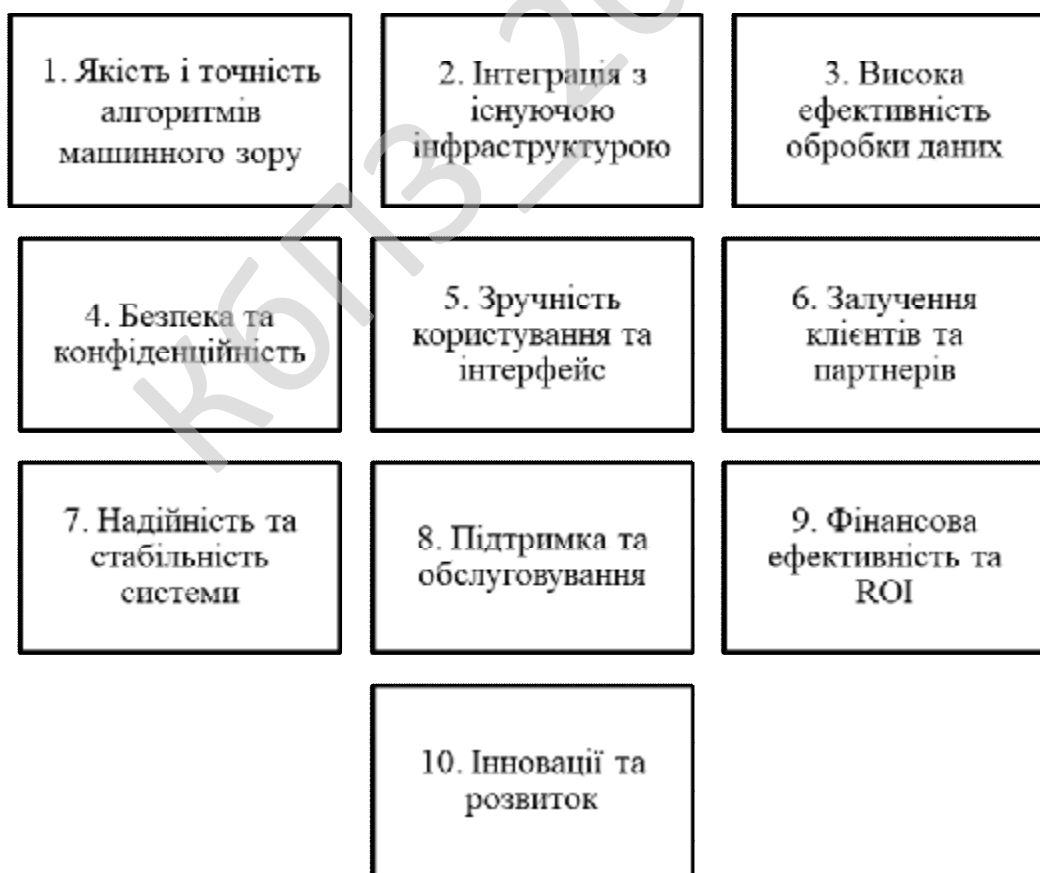


Рисунок 7.6 – Ключові фактори успіху проекту

Ключові фактори успіху для проєкту програмної реалізації системи відеоаналітики на основі машинного зору включають високоякісні алгоритми, інтеграцію з існуючими інфраструктурами, надійність та безпеку системи, підтримку клієнтів, ефективність фінансових витрат, а також здатність адаптуватися до нових технологій і потреб ринку. Важливо також забезпечити постійне вдосконалення та підтримку системи для досягнення максимального комерційного та технологічного успіху.

КБПЗ_2024

					ВКРМ-123.24.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		79

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

Згідно закону України “Про охорону праці” [3] кожна компанія впроваджує заходи з охорони праці. Реалізується трудові відносини з вживанням необхідних засобів з охорони праці та розробки відповідних документів:

- Інструкцій з охорони праці по кожній професії і загальні.
- Положення про охорону праці.
- Накази з охорони праці.
- Журнали реєстрації та інструктажу.

Роботодавець створює відділ який працює відповідно до типового положення, яку затверджується центральним органом виконавчої влади і забезпечує виконання вимог державної політики у сфері охорони праці.

За недотриманням вимог, керівники ІТ компаній можуть бути притягнуті до відповідальності, яка виглядає у виді накладання штрафу. Якщо в результаті порушення умов охорони праці є постраждалі працівники то керівні особи ІТ компаній притягуються до кримінальної відповідальності.

Законом України “Про охорону праці” [3] регламентуються загальні положення державної політики в галузі охорони праці, а конкретизуються ці положення нормативно-правовими актами про охорону праці, зокрема Наказом Міністерства соціальної політики України 14.02.2018 № 207, який зареєстровано в Міністерстві юстиції України 25 квітня 2018 р. за №508/31960 «Про затвердження Вимог щодо безпеки та захисту здоров’я працівників під час роботи з екранними пристроями» [5], яким затверджено нормативно-правовий акт з охорони праці НПАОП 0.00-7.15-18, «Правила охорони праці під час експлуатації електронно-обчислювальних машин», та «Державні санітарні правила і норми

					ВКРМ-123.24.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		80

роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98.

Програмісти у процесі роботи мають негативний вплив на органи зору, а також мають значну розумову напругою і нервово-емоційне навантаження. Руки (суглоби пальців та м'язи рук) при роботі з клавіатурою мають теж істотне навантаженням. До шкідливих факторів, які впливають на робітників галузі інформаційних технологій (ІТ) спеціалісти відносять високочастотні електромагнітні коливання (випромінювання) роботи апаратної частини ЕОМ та виділення шкідливих газів.

Ці шкідливі фактори можуть привести до професійних захворювань.

Розглянемо шкідливі чинники роботи програмістів керуючись наступними нормативно-правовими актами: «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98 [5], та «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями» НПАОП 0.00-7.15-18.

Умови праці програміста включають наступні фактори:

- параметри повітряного середовища в приміщенні;
- вентиляція приміщення;
- освітлення приміщення;
- параметри повітряного середовища в приміщенні, тощо.

Щоб запропонувати заходи щодо зменшення негативного впливу комп'ютера на організм людини визначимо фактори, які можуть викликати професійне захворювання і впливають на працездатність програміста.

8.2 Аналіз умов праці на робочому місці ІТ-фахівця

На робочому місці ІТ-фахівця (або програміста) виникають небезпечні та шкідливі для безпечної життєдіяльності фактори:

- підвищений рівень шуму;

					ВКРМ-123.24.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		81

- несприятливі мікрокліматичні умови;
- недостатній рівень освітленості;
- шкідливі речовини;
- підвищений рівень електромагнітних випромінювань радіочастот;
- висока напруга електричної мережі;
- статична електрика та інші.

Робота програміста супроводжується також підвищеним ступенем напруженості трудового процесу. При систематичному впливі виробничих факторів, які не відповідають нормативним показникам, зростає рівень професійно зумовленої захворюваності працюючих та можуть виникнути професійні захворювання органів зору, руху, нервової системи. Таким чином, вивчення умов праці на робочому місці програміста є необхідною умовою запобігання негативних наслідків впливу небезпечних та шкідливих факторів. Робоче місце, добре пристосоване до трудової діяльності інженера, правильно і доцільно організоване, щодо простору, форми, розміру забезпечує йому зручне положення при роботі і високу продуктивність праці при найменшому фізичному і психічному напруженні.

Нормування параметрів проводиться в залежності від періоду року та категорії важкості виконуваних робіт. Для постійних робочих місць, якими є робочі місця ІТ-фахівців, встановлені оптимальні параметри мікроклімату, а за неможливості їх дотримання використовують допустимі параметри. Робота ІТ-фахівця за важкістю відноситься до Іа (роботи, що виконуються сидячи і не потребують фізичного напруження) та Іб (роботи, що виконуються сидячи, стоячи або пов'язані з ходінням та супроводжуються деяким фізичним напруженням) категорій. В таблиці 8.1. наведені оптимальні параметри мікроклімату в приміщеннях.

Виміряні за допомогою приладів температура та вологість у приміщеннях праці ІТ-фахівців повинні відповідати зазначеним у таблиці для теплого періоду року. Слід зазначити, що для нормалізації параметрів мікроклімату слід

					ВКРМ-123.24.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		82

При розробці оптимальних умов праці програміста необхідно враховувати освітленість. Рациональне освітлення робочого місця є одним з найважливіших факторів, що впливають на ефективність трудової діяльності людини, що попереджають травматизм і професійні захворювання. Правильно організоване освітлення створює сприятливі умови праці, підвищує працездатність і продуктивність праці. Освітлення на робочому місці програміста повинно бути таким, щоб працівник міг без напруги зору виконувати свою роботу. Стомлюваність органів зору залежить від ряду причин: недостатність освітленості; надмірна освітленість; неправильний напрям світла. Недостатність освітлення приводить до напруги зору, ослабляє увагу, приводить до настання передчасної стомленості. Надмірно яскраве освітлення викликає засліплення, роздратування і різь в очах. Неправильний напрямок світла на робочому місці може створювати різкі тіні, відблиски, дезорієнтувати працюючого. Всі ці причини можуть призвести до нещасного випадку або профзахворювань. [4]

8.3 Пропозиції щодо підвищення працездатності ІТ-фахівців

Поява та впровадження нових інформаційно-комунікаційних технологій зумовлює необхідність подальшого вдосконалення охорони праці фахівців ІТ-індустрії. Все це потребує розробки нових нормативно-правових актів з регламентації праці та відпочинку фахівців ІТ-індустрії і стандартів підприємств, центрів комп'ютерної техніки, центрів інформаційних технологій, сучасних комп'ютерних класів. Для підвищення розумової працездатності то зорової роботи повинна здійснюватися ергономічна оптимізація в рамках системи «оператор-термінал», яка сприятиме результативній фізичній та інтелектуальній працездатності і відновленню психосоматичного здоров'я фахівців ІТ-індустрії.

Всі наведені заходи щодо вдосконалення охорони праці фахівців ІТ-індустрії повинні контролюватися службою охорони праці та комісією з охорони праці підприємства. Особливе значення у соціальному захисті цієї категорії

					ВКРМ-123.24.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		84

працівників належить прийняття комплексного договору, який може забезпечити фахівців додатковими пільгами та компенсаціями.

Пропозиції щодо підвищення працездатності ІТ-фахівців, розіб'ємо на наступні групи:

1. Середовище і розпорядок праці. Для мінімізації негативних ефектів, що пов'язані з перевтомленням ІТ-фахівців, потрібно чітко прописати і реалізувати графік періодів праці-відпочинку, щоб фахівець міг можливість переключити увагу, дати можливість відпочити очам, мозку, елементарно, встати розім'яти ноги. Також потрібно зробити максимально комфортними умови мікроклімату у офісному приміщенні, де працюють ІТ-фахівці. Мається на увазі встановлення і експлуатація, коли виникає необхідність, кондиціонерів, опалення, та системи вентиляції, задля попередження перегрівання, переохолодження ІТ-фахівців, і подальшої неможливості ними виконувати свої функції. Також, за можливості, нами пропонується введення практики віддаленої праці ІТ-фахівцями, якщо роботодавець не може забезпечити оптимальні і безпечні умови в офісному приміщенні, або якщо фахівця вони не влаштовують із певних причин.

2. Фізичні і психоемоційні чинники. Першим і найважливішим чинником, що впливає на працездатність ІТ-фахівців є робоче місце, і саме тому, роботодавець має забезпечити максимальний його комфорт і безпеку. Гарантією цих факторів може слугувати сертифікація меблів, що використовуються на підприємстві ІТ-галузі. Тому нами пропонується закупівля тільки меблів, які пошли сертифікацію на відповідність. Під психоемоційними чинниками ми розуміємо гарне самопочуття фахівців, позитивний настрій, гарний психологічний клімат у колективі, тощо. Задля того, щоб психоемоційні чинники мали максимально позитивний ефект, керівництву слід поводити заходи, які сприятимуть укріпленню і покращенню міжособистісних стосунків у колективі, таких як психологічні тренінги, тимбілдінг, спортивні змагання і естафети. Також, сюди можна віднести розробку і впровадження системи мотивації працівників, як фінансової, так моральної і адміністративної.

					ВКРМ-123.24.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		85

Загальний опір розтіканню електричного струму заземлювача [11]:

$$R = (R_0 \cdot R_{\Pi}) / (R_0 \cdot \eta_{\Pi} + N \cdot R_{\Pi} \cdot K_{ев}) = \\ = (14,9 \cdot 20,1) / (14,9 \cdot 0,55 + 7 \cdot 20,1 \cdot 0,53) = 3,56 \text{ Ом.}$$

де $\eta_{\Pi} = 0,55$ – табличне значення коефіцієнта екранування з'єднуючої полоси [11].

Умова $R \leq R_{3Н}$ виконується ($3,56 \leq 4$).

Остаточна кількість вертикальних електродів дорівнює 7.

За потреби можна зменшити кількість електродів заземлювача, зменшивши загальний опір розтіканню електричного струму заземлювача методом зменшення питомого опору ґрунта, домішуючи у ґрунт безпосередньо навколи електродів заземлювача розчини солей NaCl, CaCl, сажу, соду, шлак або спеціальні суміші.

8.5 Висновки до розділу

Дотримання всіх необхідних умов праці не лише сприяє збереженню здоров'я працівників, а також підвищує ефективність виробництва в цілому.

З цих міркувань було здійснено аналіз приміщення, призначеного для праці програмістів, проведено розгляд небезпечних та шкідливих факторів, що негативно впливають на програмістів під час роботи.

Тільки повна усвідомленість працівника про можливі небезпеки, що можуть підстерігати його на робочому місці та дотримання вимог нормативних актів з питань охорони праці та відповідних рекомендацій фахівців, дозволять значною мірою знизити негативний вплив шкідливих та небезпечних факторів при роботі з комп'ютером на організм людини.

Виконано розрахунок захисного штучного заземлення, як одного з ключових факторів безпеки програміста.

					ВКРМ-123.24.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		88

9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи відеоаналітики на основі систем машинного зору.

В межах України в недостатній мірі представлені вітчизняні розробки в цій області.

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти наведені теоретичне узагальнення й рішення наукового завдання дослідження методів відеоаналітики на основі систем машинного зору.

Рішення даного завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем відеоаналітики на основі систем машинного зору.
- Досліджена система відеоаналітики на основі систем машинного зору.
- На основі отриманих результатів досліджень створена програмна реалізація системи відеоаналітики на основі систем машинного зору.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання відеоаналітики на основі систем машинного зору.

Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

Розроблене програмне забезпечення має простий, дружній та зручний інтерфейс користувача, що забезпечує легкість у освоєнні роботи програмного продукту, зручність у використанні, і не потребує особливих спеціальних знань.

					ВКРМ-123.24.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		89

При створенні програмного забезпечення було використано об'єктно-орієнтований підхід, що відповідає сучасним тенденціям у галузі розробки комерційних програмних систем.

Програма реалізована на мові високого рівня Python. Дана мова програмування дозволяє найбільш ефективно обробляти дані. Це дозволило мінімізувати строк розробки програмного забезпечення, і, як слід, зменшити витрати на його розробку. Запропоноване програмне забезпечення ділиться на загальне програмне забезпечення, що поставляється із засобами обчислювальної техніки й спеціальне програмне забезпечення, що спеціально розроблене для даної конкретної системи й включає програми, що реалізують її функції.

Програма призначена для виконання під управлінням багатозадачної операційної системи Windows 10/11.

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм SHACAL-1.

В цілому створене програмне забезпечення підтверджує правильність використаних проектних рішень та повністю відповідає вимогам технічного завдання. Створене програмне забезпечення має потенційну можливість для подальшого вдосконалення і застосування у різних галузях.

Проведено маркетингове та економічне обґрунтування ІТ-проєкту, що дозволило визначити ключові фактори успіху даного проєкту.

					ВКРМ-123.24.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		90

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Риба О.С. Дослідження та програмна реалізація системи відеоаналітики на основі систем машинного зору // Збірник праць молодих науковців ЦНТУ. – Вип. 14. – Кропивницький: ЦНТУ, 2024.
2. Peter Shirley, Steve Marschner. Fundamentals of Computer Graphics. 2009
3. Михайло Пічугін, Іван Канкін, Володимир Воротніков Комп'ютерна графіка. Навчальний посібник / Центр навчальної літератури 346 с. 2019р.
4. Маценко В.Г. Комп'ютерна графіка: Навчальний посібник. – Чернівці: Рута, 2009 – 343 с.
5. Інженерна комп'ютерна графіка: підручник / В.В. Проців [та ін.] / М-во освіти і науки України, Нац. гірн. унт-т. – Дніпро: НГУ, 2017. – 247 с.
6. Проців В.В. Прикладна комп'ютерна графіка [Текст]: Навч. посібник / В.В. Проців, К.А. Зіборов, К.М. Бас, Г.К. Ванжа; М-во освіти і наук, Нац. гірн. унт. – Д.: НГУ, 2016. – 187 с.
7. Kopf, Johannes and Lischinski, Dani. Depixelizing Pixel Art (англ.) // ACM Trans. Graph. – 2011. – Vol. 30, no. 4. – P. 99:1--99:8.
8. Giachetti, Andrea and Asuni, Nicola. Real-Time Artifact-Free Image Upscaling (англ.) // Trans. Img. Proc.. – 2011. – Vol. 20, no. 10. – P. 2760—2768.
9. Kuznetsov, O., Frontoni, E., Kryvinska, N., Chevardin, V., Smirnov, O. «Wireless Network Encryption Stream Ciphers, Computational Modeling, and Security Analysis». *Computational Modeling and Simulation of Advanced Wireless Communication Systems*, 2024, pp. 379–402.
10. Kuznetsov, O., Frontoni, E., Kryvinska, N., Smirnov, O., Imoize, G.L. «Computational Modeling of Enhanced Spread Spectrum Codes for Asynchronous Wireless Communication». *Computational Modeling and Simulation of Advanced Wireless Communication Systems*, 2024, pp. 403–447

					ВКРМ-123.24.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		91

11. Al-Mudhafar Aqeel, A.M., Smirnova, T., Buravchenko, K., Smirnov, O. «The method of assessing and improving the user experience of subscribers in software-configured networks based on the use of machine learning». *Advanced Information Systems*, 2023, 7(2), pp. 49-56.

12. Smirnov, O., Sydorenko, V., Aleksander, M., Zhyharevych, O., Yanchev, S. «Simulation of the cloud IoT-based monitoring system for critical infrastructures». *CEUR Workshop Proceedings*, Volume 3530, 2023, pp. 256-265.

13. Smirnov, O., Odarchenko, R., Smirnova, T., Bondar, S., Volosheniuk, D. «Optimal Structure Construction of Private 5G Network for the Needs of Enterprises». *Lecture Notes on Data Engineering and Communications Technologies*, 2023, 178, pp. 208–223.

14. Smirnov, O., Neskrodieva, T., Fedorov, E., Rudakov, K., Neskrodieva, A. «Method Detection Audit Data Anomalies on Basis Restricted Cauchy Machine» *CEUR Workshop Proceedings*, Volume 3187, 2022,

15. Smirnov O., Smirnova T., Anas M. Al-Oraiqat, Drieiev O., Polishchuk L., Sheroz Khan, Yassin M. Y. Hasan, Aladdein M. Amro, Hazim S. AlRawashdeh «Method for Determining Treated Metal Surface Quality Using Computer Vision Technology». *Sensors (Basel, Switzerland)* Volume 22, Issue 16, 6223, 2022.

16. Smirnov O., Kuznetsov A., Kryvinska N., Kiian A., Kuznetsova K. «Full Non-Binary Constant-Weight Codes». *SN Computer Science*, Vol 2, 337, 2021. <https://doi.org/10.1007/s42979-021-00739-w>

17. Smirnov O., Kuznetsov A., Zhora V., Onikiychuk A., Pieshkova O. «Hiding Messages in Audio Files Using Direct Spread Spectrum». 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2021, Cracow, Poland, 22-25 September 2021. P. 414-418.

18. Smirnov O., Kuznetsov A., Lokotkova I., Kuznetsova T., Florov S., Lebid O. «Using Orthogonal Signals to Hide Information in Images». 4 IEEE

International Conference on Advanced Information and Communication Technologies (AICT) – 2021, Lviv, Ukraine, September 21-25, 2021. P. 255-260.

19. Smirnov, O., Kuznetsov, A., Potii, O., Poluyanenko, N., Stelnyk, I., Mialkovsky, D. «Combining and filtering functions in the framework of nonlinear-feedback shift register». International Journal of Computing; 2020, Volume 19, Issue 2 – Research Institute for Intelligent Computer Systems – 2020. – P. 247-256.

20. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». CEUR Workshop Proceedings. Volume 2740, 2020, Pages 102-114.

21. Smirnov O., Alimseitova Zh., Adranova A., Akhmetov B., Lakhno V., Zhilkishbayeva G. «Models and algorithms for ensuring functional stability and cybersecurity of virtual cloud resources». Journal of theoretical and applied information technology Vol.98. No 21, 2020, P. 3334-3346.

22. Smirnov O., Kuznetsov A., Kovalchuk D., Kuznetsova T. «New technique for data hiding in cover images using adaptively generated pseudorandom sequences». CEUR Workshop Proceedings Volume 2654, 2020, Pages 1-14.

23. Smirnov O., Kuznetsov A., Onikiychuk A., Makushenko T., Anisimova O., Arischenko A. «Adaptive pseudo-random sequence generation for spread spectrum image steganography». 2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT), Ukraine, Kyiv, May 14-18. 2020. P. 161-165.

24. Smirnov O., Kuznetsov A., Kiian A., Cherep A., Kanabekova M., Chepurko I. «Testing of code-based pseudorandom number generators for post-quantum application». 2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT), Ukraine, Kyiv, May 14-18. 2020. P. 172-177.

25. Smirnov O., Kuznetsov A., Pushkar'ov A., Serhiienko R., Babenko V., Kuznetsova T., «Representation of Cascade Codes in the Frequency Domain». In: Radivilova T., Ageyev D., Kryvinska N. (eds) Data-Centric Business and

Applications. Lecture Notes on Data Engineering and Communications Technologies, vol 48. Springer, Cham. 2021. pp 557-587.

26. Smirnov, O., Drieieva, H., Drieiev, O., Polishchuk, Y., Brzhanov, R., Aleksander, M. «Method of fractal traffic generation by a model of generator on the graph». CEUR Workshop Proceedings Volume 2616, 2020, Pages 366-379.

27. Smirnov, O., Drieieva, H., Drieiev, O., Simakhin, V., Bondar, S., Odarchenko, R. «Managing multifractal properties of the binary sequence generated with the Markov chains», CEUR Workshop Proceedings Volume 2608, 2020, Pages 633-645.

28. Smirnov, O., Kuznetsov, A., Gorbacheva, L., Babenko, V., «Hiding data in images using a pseudo-random sequence», CEUR Workshop Proceedings Volume 2608, 2020, Pages 646-660.

29. Zhurakovskiy, B., Tsopa, N., Batrak, Y., Odarchenko, R., Smirnova, T «Comparative analysis of modern formats of lossy audio compression». Workshop Proceedings, 2020, 2654, стр. 315-327.

30. Smirnov O. Kuznetsov A., Zaichenko Yu., Pastukhov M., Oleshko O., Kuznetsova K., «Formation of Discrete Signals with Special Correlation Properties». International Conference on Information and Telecommunication Technologies and Radio Electronics, UkrMiCo 2019; Odessa; Ukraine; 9-13 September 2019. P.22-28.

31. Smirnov, O., Kuznetsov, A., Kolovanova, I., Kuznetsova, T., «Noise immunity of the algebraic geometric codes». International Journal of Computing; 2019, Volume 18, Issue 4 – Research Institute for Intelligent Computer Systems – 2019. – P. 393-407.

32. Smirnov, O., Kuznetsov, A., Reshetniak, O., Ivko, N., Katkova, T., Kuznetsova, T., «Generators of Pseudorandom Sequence with Multilevel Function of Correlation». 2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T), Kyiv, Ukraine, 8 – 11 October 2019 . P.517-522.

33. Smirnov, O., Krasnobayev, V., Yanko, A., Kuznetsova, T. «Methods of nulling numbers in the system of residual classes». CEUR Workshop Proceedings, Vol 2588, P. 90-106, 2019.

34. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Averchev, A., Pastukhov, M., Kuznetsova, K., «Formation of Pseudorandom Sequences with Special Correlation Properties», 2019 3rd International Conference on Advanced Information and Communications Technologies, AICT -2019/ Lviv, Ukraine, 2-6 July, 2019, P. 395-399.

35. Smirnov, O., Kuznetsov, A., Kavun, S., Babenko, B., Nakisko, O., Kuznetsova, K., «Malware Correlation Monitoring in Computer Networks of Promising Smart Grids», 2019 IEEE 6th International Conference On Energy Smart Systems (2019 IEEE ESS), Kyiv, Ukraine April 17-19, 2019 P. 347-352.

36. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Pastukhov, M., Kuznetsova, K., Prokopovych-Tkachenko, D., «Discrete Signals with Special Correlation Properties», CEUR Workshop Proceedings Volume 2353, CEUR Workshop Proceedings 2019, Pages 618-629.

37. Smirnov A.A., Kuznetsov A.A., Danilenko D.A., Berezovsky A., «The statistical analysis of a network traffic for the intrusion detection and prevention systems», Telecommunications and Radio Engineering. – Volume 74, Issue 1. – Begel House Inc. – 2015. – P. 61-78.

38. Smirnov O., Kuznetsov A., Kovalchuk D., Kuznetsova T. «New Technique for Hiding Data in Cover Images Using Adaptively Generated Pseudorandom Sequences». CEUR Workshop Proceedings Volume 2732, 2020, Pages 214-227.

39. Смірнова Т.В., Коноплицька-Слободенюк О.К., Буравченко К.О., Смірнов С.А., Кравчук О.В., Козірова Н.Л., Смірнов О.А. «Дослідження технологій забезпечення кібербезпеки хмарних сервісів IaaS, PaaS та SaaS». *Кібербезпека: освіта, наука, техніка*. 2024. №4(24), С. 6-27.

40. Батрак О., Смірнова Т., Гнатюк В., Одарченко Р., Смірнов О. «Дослідження показників ефективності функціонування та перспектив розвитку систем IP-телефонії». *Підводні технології*, 2024, № 13, с. 28-35.

41. Аль-Мудхафар Акіл Абдулхуссейн М., Смірнова Т.В., Буравченко К.О., Смірнов О.А. «Метод оцінки та підвищення користувальницького досвіду абонентів в програмно-конфігурованих мережах на основі використання машинного навчання». *Сучасні інформаційні системи*, 2023, том 7, № 2, С. 49-56.

42. Т.В. Смірнова, О.М. Дреєв, О.А. Смірнов «Хмарна інформаційна система оцінювання шорсткості з використанням дискретного частотного аналізу макروفотografій». IV міжнародна науково-практична конференція «Інформаційна безпека та комп'ютерні технології», м. Кропивницький. 15-16 квітня 2021р. – Кропивницький: ЦНТУ. – 2021. – С. 30.

43. О.А. Смірнов, П.С. Усік, «Дослідження перспектив використання технологічних рішень в мережах 5G» у Кібербезпека та інформаційні технології: монографія. – Х. : ТОВ «ДІСА ПЛЮС», 2020.С. 122-135.

44. О.А.Смірнов, Т.В.Смірнова, Л.І. Поліщук, К.О. Буравченко, А.О.Макевнін, «Дослідження хмарних технологій як сервісів», Кібербезпека: освіта, наука, техніка. № 3(7). С. 43-62. 2020.

45. Смірнов О.А., Дреєва Г.М., Дреєв О.М., Смірнова Т.В. «Фрактальний аналіз генератора самоподібного трафіку на основі ланцюга Маркова». *Центральноукраїнський науковий вісник. Технічні науки*. № 2(33). с. 161-172, 2019.

46. О. Смірнов, Є. Деменко, О. Онікійчук, А. Арищенко, Л. Горбачова, «Формування псевдовипадкових послідовностей для приховування даних в зображеннях» *Комп'ютерні науки та кібербезпека*. № 4. С. 30-37. 2019.

47. Смірнов О.А., Коноплицька-Слободенюк О.К., Смірнов С.А., Буравченко К.О., Смірнова Т.В. Поліщук Л.І. *Проектування комп'ютерних систем та мереж*. Навчальний посібник – Кропивницький: вид. Лисенко В.Ф. 2019. – 264 с.

					ВКРМ-123.24.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		96

48. Smirnov, O., Kuznetsov, A., Kuznetsova., K. Synthesis of Discrete Signals with Improved Correlation Properties. Монографія: In.: ISCI'2019: Information Security in Critical Infrastructures. Collective monograph. Edited by Ivan D. Gorbenko and Alexandr A. Kuznetsov, ASC Academic Publishing, USA, 2019, pp. 281-299. – ISBN: 978-0-9989826-8-7 (Hardback), ISBN: 978-0-9989826-9-4 (Ebook).

49. Смірнов О.А., Дреєва Г.М. Метод генерування фрактального трафіку за допомогою моделі генератора на графі. Монографія: Інформаційна безпека та інформаційні технології : монографія / за заг. ред. В. С. Пономаренка. – Х. : Вид. Рожко С.Г. 2019. С. 123-139

50. Дреєва Г.М., Смірнов О.А., Дреєв О.М. Метод генерування фрактальноподібної числової послідовності на основі скінченного автомату для моделювання трафіку у мережі. Центральноукраїнський науковий вісник. Технічні науки. № 1(32). с. 173-183, 2019.

51. Смірнов О.А., Кавун С.В., Коваленко О.В., Дреєв О.М. Мережні інформаційні технології. Навчальний посібник – Кіровоград: РВЛ КНТУ, 2016. – 159 с.

52. Смірнов О.А., Смірнов С.А. Дідик А.К., Дреєв О.М. Моделі системи нейромережових експертів безпечної маршрутизації у хмарних антивірусних системах. Збірник наукових праць "Системи обробки інформації". – Випуск 3 (140). – Х.: ХУПС – 2016. – С. 36-39.

53. Смірнов О.А., Кавун С.В., Коваленко О.В., Доренський О.П., Дреєв О.М., Вялкова В.І. Комп'ютерні мережі. Навчальний посібник – Кіровоград: РВЛ КНТУ, 2016. – 233 с.

54. Смірнов О.А., Дреєв О.М. Порівняння бітових щільностей при використанні різних методів кодування інформації. Збірник наукових праць "Системи обробки інформації". – Випуск 2 (118). т.2. – Х.: ХУПС – 2014. – С. 64-67

					ВКРМ-123.24.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		97

Додаток А
(обов'язковий)

Технічне завдання

Зміст

1 Найменування та область застосування.....	2
2 Підстава для розробки.....	2
3 Мета та призначення розробки.....	2
4 Джерела розробки.....	2
5 Технічні вимоги.....	2
5.1 Вміст проекту.....	2
5.2 Показники призначення.....	3
5.3 Вимоги до функціональних характеристик.....	3
5.4 Вимоги до архітектури.....	3
5.5 Вимоги до надійності.....	3
5.6 Умови експлуатації.....	4
5.7 Вимоги до складу та параметрів технічних засобів.....	4
5.8 Вимоги до інформаційної і програмної сумісності.....	4
5.8.1 Обладнання.....	4
5.8.2 Мова програмування.....	4
5.8.3 Вхідні дані.....	5
5.8.4 Вихідні дані.....	5
6 Вимоги до програмної документації.....	5
7 Економічні вимоги.....	5
8 Вимоги щодо охорони праці.....	5
9 Перелік документів, що розробляються.....	6
10 Етапи розробки.....	6
11 Порядок контролю та приймання.....	6

					ВКРМ-123.24.0036.00.00.ТЗ			
Вим.	Арк.	№ документа	Підпис	Дата				
Розробив	Риба О.С.				<i>Дослідження та програмна реалізація системи відеоаналітики на основі систем машинного зору</i>	Літ.	Аркуш	Аркушів
Перевірів	Дресва Г.М.					М	1	6
Н. Контр.	Коваленко А.С.				ЦНТУ КІ-23М			
Затв.	Смірнов О.А.							

1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи відеоаналітики на основі систем машинного зору.

2 Підстава для розробки

Підставою для розробки служить завдання на випускню кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, видане на кафедрі кібербезпеки та програмного забезпечення (нак. № 19-13 від 07.08.2024 року).

3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є дослідження та програмна реалізація системи відеоаналітики на основі систем машинного зору.

4 Джерела розробки

Джерелом цієї випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є стосовна до теми література і існуючі аналоги.

5 Технічні вимоги

5.1 Склад продукції

Складниками розробки є:

- вибір і обґрунтування методів реалізації проекту;
- розробка програмної частин системи, а також розробка взаємодії системи з ОС та з користувачем;

					ВКРМ-123.24.0036.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

- техніко-економічне обґрунтування доцільності прийнятого до розробки програмного забезпечення;
- аналіз умов праці;
- розробка програми, що реалізує спроектовані алгоритми роботи системи.

5.2 Показники призначення

Система повинна забезпечувати:

- програмну реалізацію системи відеоаналітики на основі систем машинного зору;
- цілісність даних у процесі роботи та при зберіганні;
- простий, інтуїтивно зрозумілий інтерфейс.

5.3 Вимоги до функціональних характеристик

Розроблене програмне забезпечення не повинно мати обмежень на версію драйверів та операційної системи.

5.4 Вимоги до архітектури

Компонент, що розробляється повинен використовувати системні засоби та апаратні засоби, що на даному етапі розвитку обчислювальної техніки найбільше поширені.

5.5 Вимоги до надійності

Програмні модулі написані по всім правилам, які стосуються стандартних викликів процедур, функцій, методів і форм, визначених технічною документацією на середовище розробки.

					ВКРМ-123.24.0036.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		3

5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

- температура повітря: 19-20 град. по Цельсію;
- відносна вологість повітря до 80%;
- атмосферний тиск 107 кПа.

5.7 Вимоги до складу та параметрів технічних засобів

Програмне забезпечення повинно бути реалізоване на ПЕОМ, працювати в ОС Windows 10/11 і з сумісними з цією платформою пристроями і прикладним програмним забезпеченням.

5.8 Вимоги до інформаційної і програмної сумісності

Переносність програмного забезпечення повинна бути забезпечена за рахунок його реалізації стандартного інтерфейсу взаємодії з ОС, що працюють під управлінням ОС Windows 10/11.

5.8.1 Обладнання

Комп'ютер Intel® Celeron/8 Mb/1.2 Gb/SVGA 14" 1Mb або сумісні з ним.

5.8.2 Мова програмування

Середовище Python.

					ВКРМ-123.24.0036.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

5.8.3 Вхідні дані

Опис алгоритму роботи запропонованої системи.

5.8.4 Вихідні дані

Робоча програма.

6 Вимоги до програмної документації

Програмна продукція повинна бути представлена у виді опису структури даних, схем та опису алгоритму, а також текстів вихідних модулів програмного забезпечення згідно ЄСПД .

7 Економічні вимоги

7.1 Для ПЗ необхідно виробити функціонально-вартісний аналіз варіантів розробки.

7.2 Виконати маркетингове та економічне обґрунтування ІТ-проєкту з урахуванням цін на 3 вересня 2024 року.

8 Вимоги щодо охорони праці

В частині охорони праці випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти повинні бути розглянуті пропозиції щодо підвищення працездатності ІТ-фахівців.

					ВКРМ-123.24.0036.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		5

9 Перелік документів, що розробляються

- Наукова новизна – 1 аркуш.
- Структурна схема системи – 1 аркуш.
- Функціональна схема системи – 1 аркуш.
- Діаграма процесів – 1 аркуш.
- Блок-схема алгоритму роботи програми – 2 аркуша.
- Показники економічної ефективності – 1 аркуш.
- Пояснювальна записка – 97 аркушів.

10 Етапи розробки

10.1 Збір і обробка інформації по темі випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти. Постановка задачі на виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти (складання ТЗ).

10.2 Проведення досліджень або експериментальних робіт для уточнення основних положень випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.

10.3 Розробка функціональних схем, блок схем алгоритмів роботи програмного забезпечення.

10.4 Побудова схем взаємодії даних.

10.5 Створення прототипу ПЗ.

10.6 Віднаходження ПЗ, аналіз отриманих результатів.

10.7 Робота над питанням охорони праці і техніки безпеки.

10.8 Маркетингове та економічне обґрунтування ІТ-проєкту.

10.9 Оформлення пояснювальної записки і виконання робіт по графічній частині.

11 Порядок контролю та приймання

11.1 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на попередній захист 02.12.2024 р.

11.2 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на захист 18.12.2024 р.

					ВКРМ-123.24.0036.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		6

Додаток Б
(обов'язковий)

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет

ЗАТВЕРДЖУЮ

Керівник випускної кваліфікаційної роботи за
другим (магістерським) рівнем вищої освіти

_____ Дреєва Г.М.

*Дослідження та програмна реалізація
системи відеоаналітики на основі систем машинного зору*

Лістинг програми

Код документу 12

Носій: CD/DVD-диск / USB-флеш-накопичувач

Загальна кількість аркушів: 19

Літера: РП

Кропивницький – 2024 року

Основна програма

```

import cv2
import numpy as np
import time
import os

# Модуль для захоплення відео
def video_capture(source=0):
    cap = cv2.VideoCapture(source)
    if not cap.isOpened():
        print("Не вдалося відкрити джерело відео")
        return None
    return cap

# Модуль для збереження відео
def video_writer(output_file, frame_width, frame_height, fps=20):
    fourcc = cv2.VideoWriter_fourcc(*'XVID')
    out = cv2.VideoWriter(output_file, fourcc, fps, (frame_width, frame_height))
    return out

# Модуль для попередньої обробки кадру
def preprocess_frame(frame):
    gray_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    blur_frame = cv2.GaussianBlur(gray_frame, (5, 5), 0)
    return blur_frame

# Модуль для обчислення різниці між кадрами для виявлення руху
def frame_difference(prev_frame, current_frame):
    diff_frame = cv2.absdiff(prev_frame, current_frame)
    _, thresh_frame = cv2.threshold(diff_frame, 25, 255, cv2.THRESH_BINARY)
    return thresh_frame

# Модуль для виявлення об'єктів на основі контурів
def detect_contours(thresh_frame):
    contours, _ = cv2.findContours(thresh_frame, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)
    return contours

# Модуль для виявлення та відстеження об'єктів
def track_objects(frame, contours):
    for contour in contours:
        if cv2.contourArea(contour) > 500:
            x, y, w, h = cv2.boundingRect(contour)
            cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)
    return frame

# Модуль для вимірювання FPS
def calculate_fps(start_time, frame_count):
    end_time = time.time()
    elapsed_time = end_time - start_time
    fps = frame_count / elapsed_time if elapsed_time > 0 else 0
    return fps

# Модуль для автоматичного збереження відео при виявленні руху
def auto_save_video(frame, motion_detected, video_writer):
    if motion_detected:
        video_writer.write(frame)

# Модуль для обробки відео з використанням методів машинного зору
def process_video(source=0, output_file="output.avi"):
    cap = video_capture(source)
    if cap is None:
        return

    ret, prev_frame = cap.read()

```

```

if not ret:
    print("Не вдалося отримати перший кадр")
    return

prev_frame = preprocess_frame(prev_frame)
frame_width = int(cap.get(3))
frame_height = int(cap.get(4))

out = video_writer(output_file, frame_width, frame_height)

frame_count = 0
start_time = time.time()

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

    current_frame = preprocess_frame(frame)
    thresh_frame = frame_difference(prev_frame, current_frame)
    contours = detect_contours(thresh_frame)

    frame_with_objects = track_objects(frame, contours)

    auto_save_video(frame, len(contours) > 0, out)

    fps = calculate_fps(start_time, frame_count)
    cv2.putText(frame_with_objects, f"FPS: {int(fps)}", (10, 30),
cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 2)

    cv2.imshow("Video Stream", frame_with_objects)

    frame_count += 1
    prev_frame = current_frame

    if cv2.waitKey(30) & 0xFF == ord('q'):
        break

cap.release()
out.release()
cv2.destroyAllWindows()

# Модуль для розпізнавання облич у відеопотоці
def face_detection(frame, face_cascade):
    gray_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces = face_cascade.detectMultiScale(gray_frame, scaleFactor=1.1,
minNeighbors=5, minSize=(30, 30), flags=cv2.CASCADE_SCALE_IMAGE)

    for (x, y, w, h) in faces:
        cv2.rectangle(frame, (x, y), (x + w, y + h), (255, 0, 0), 2)

    return frame

# Модуль для інтеграції розпізнавання облич у основну систему
def process_video_with_face_detection(source=0,
face_cascade_path="haarcascade_frontalface_default.xml"):
    face_cascade = cv2.CascadeClassifier(face_cascade_path)

    cap = video_capture(source)
    if cap is None:
        return

    ret, prev_frame = cap.read()
    if not ret:
        print("Не вдалося отримати перший кадр")
        return

    prev_frame = preprocess_frame(prev_frame)
    frame_width = int(cap.get(3))

```

```

frame_height = int(cap.get(4))

frame_count = 0
start_time = time.time()

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

    current_frame = preprocess_frame(frame)
    thresh_frame = frame_difference(prev_frame, current_frame)
    contours = detect_contours(thresh_frame)

    frame_with_objects = track_objects(frame, contours)
    frame_with_faces = face_detection(frame_with_objects, face_cascade)

    fps = calculate_fps(start_time, frame_count)
    cv2.putText(frame_with_faces, f"FPS: {int(fps)}", (10, 30),
cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 2)

    cv2.imshow("Video Stream with Face Detection", frame_with_faces)

    frame_count += 1
    prev_frame = current_frame

    if cv2.waitKey(30) & 0xFF == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()

# Модуль для збереження обробленого відео з обличчями
def save_video_with_face_detection(source=0,
output_file="output_with_faces.avi",
face_cascade_path="haarcascade_frontalface_default.xml"):
    face_cascade = cv2.CascadeClassifier(face_cascade_path)

    cap = video_capture(source)
    if cap is None:
        return

    ret, prev_frame = cap.read()
    if not ret:
        print("Не вдалося отримати перший кадр")
        return

    prev_frame = preprocess_frame(prev_frame)
    frame_width = int(cap.get(3))
    frame_height = int(cap.get(4))

    out = video_writer(output_file, frame_width, frame_height)

    frame_count = 0
    start_time = time.time()

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

        current_frame = preprocess_frame(frame)
        thresh_frame = frame_difference(prev_frame, current_frame)
        contours = detect_contours(thresh_frame)

        frame_with_objects = track_objects(frame, contours)
        frame_with_faces = face_detection(frame_with_objects, face_cascade)

        out.write(frame_with_faces)

```

```
    fps = calculate_fps(start_time, frame_count)
    cv2.putText(frame_with_faces, f"FPS: {int(fps)}", (10, 30),
cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 2)

    cv2.imshow("Saved Video with Face Detection", frame_with_faces)

    frame_count += 1
    prev_frame = current_frame

    if cv2.waitKey(30) & 0xFF == ord('q'):
        break

    cap.release()
    out.release()
    cv2.destroyAllWindows()

# Виклик основної системи з виявленням облич
save_video_with_face_detection()
```

КБПЗ_2024

Розпізнавання номерних знаків (файл license_plate_recognition.py)

```
import cv2
import numpy as np
import pytesseract

# Модуль для виявлення номерних знаків
def detect_license_plate(frame,
plate_cascade_path="haarcascade_russian_plate_number.xml"):
    plate_cascade = cv2.CascadeClassifier(plate_cascade_path)
    gray_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    plates = plate_cascade.detectMultiScale(gray_frame, scaleFactor=1.1,
minNeighbors=5, minSize=(30, 30))

    for (x, y, w, h) in plates:
        cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)
        roi = frame[y:y+h, x:x+w]
        plate_text = pytesseract.image_to_string(roi, config='--psm 8')
        cv2.putText(frame, plate_text, (x, y - 10), cv2.FONT_HERSHEY_SIMPLEX, 1,
(0, 255, 0), 2)

    return frame

# Модуль інтеграції з основною системою
def process_video_with_license_plate_recognition(source=0,
plate_cascade_path="haarcascade_russian_plate_number.xml"):
    cap = video_capture(source)

    if cap is None:
        return

    ret, prev_frame = cap.read()
    if not ret:
        print("Не вдалося отримати перший кадр")
        return

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

        frame_with_plates = detect_license_plate(frame, plate_cascade_path)
        cv2.imshow("License Plate Recognition", frame_with_plates)

        if cv2.waitKey(30) & 0xFF == ord('q'):
            break

    cap.release()
    cv2.destroyAllWindows()
```

Інтеграція з нейронними мережами для класифікації об'єктів
(файл object_classification.py)

```
import cv2
import numpy as np
from tensorflow.keras.models import load_model

# Завантаження попередньо навченої нейронної мережі
model = load_model("object_classification_model.h5")

# Модуль для класифікації об'єктів
def classify_object(frame, model):
    resized_frame = cv2.resize(frame, (224, 224))
    normalized_frame = resized_frame / 255.0
    input_data = np.expand_dims(normalized_frame, axis=0)

    prediction = model.predict(input_data)
    class_index = np.argmax(prediction)

    return class_index

# Модуль інтеграції класифікації об'єктів з основною системою
def process_video_with_object_classification(source=0):
    cap = video_capture(source)

    if cap is None:
        return

    ret, prev_frame = cap.read()
    if not ret:
        print("Не вдалося отримати перший кадр")
        return

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

        class_index = classify_object(frame, model)
        cv2.putText(frame, f"Class: {class_index}", (10, 30),
                    cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 2)
        cv2.imshow("Object Classification", frame)

        if cv2.waitKey(30) & 0xFF == ord('q'):
            break

    cap.release()
    cv2.destroyAllWindows()
```

Виявлення аномалій (файл anomaly_detection.py)

```
import cv2
import numpy as np
from sklearn.ensemble import IsolationForest

# Модуль для виявлення аномалій у відеопотоці
def detect_anomalies(frame, anomaly_model):
    frame_features = extract_frame_features(frame)
    is_anomaly = anomaly_model.predict([frame_features])
    return is_anomaly[0] == -1

# Модуль для інтеграції виявлення аномалій з основною системою
def process_video_with_anomaly_detection(source=0,
anomaly_model=IsolationForest()):
    cap = video_capture(source)

    if cap is None:
        return

    ret, prev_frame = cap.read()
    if not ret:
        print("Не вдалося отримати перший кадр")
        return

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

        if detect_anomalies(frame, anomaly_model):
            cv2.putText(frame, "Anomaly Detected!", (10, 30),
cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255), 2)

            cv2.imshow("Anomaly Detection", frame)

            if cv2.waitKey(30) & 0xFF == ord('q'):
                break

    cap.release()
    cv2.destroyAllWindows()
```

Збереження в базу даних (файл database_saving.py)

```
import sqlite3

# Модуль для збереження подій у базу даних
def save_event_to_database(event_type, timestamp):
    conn = sqlite3.connect('events.db')
    cursor = conn.cursor()

    cursor.execute('''CREATE TABLE IF NOT EXISTS events (id INTEGER PRIMARY KEY,
event_type TEXT, timestamp TEXT)''')
    cursor.execute('''INSERT INTO events (event_type, timestamp) VALUES (?,
?)''', (event_type, timestamp))

    conn.commit()
    conn.close()

# Модуль інтеграції з основною системою
def process_video_with_event_logging(source=0):
    cap = video_capture(source)

    if cap is None:
        return

    ret, prev_frame = cap.read()
    if not ret:
        print("Не вдалося отримати перший кадр")
        return

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

        motion_detected = detect_motion(frame, prev_frame)

        if motion_detected:
            save_event_to_database("Motion Detected", time.ctime())

        cv2.imshow("Event Logging", frame)

        if cv2.waitKey(30) & 0xFF == ord('q'):
            break

    cap.release()
    cv2.destroyAllWindows()
```

Веб-інтерфейс для віддаленого моніторингу (файл web_interface.py)

```
from flask import Flask, Response
import cv2

app = Flask(__name__)

# Модуль для віддаленого моніторингу через веб-інтерфейс
def generate_frames():
    cap = video_capture(0)

    if cap is None:
        return

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

        ret, buffer = cv2.imencode('.jpg', frame)
        frame = buffer.tobytes()

        yield (b'--frame\r\n'
              b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n')

    cap.release()

@app.route('/video_feed')
def video_feed():
    return Response(generate_frames(), mimetype='multipart/x-mixed-replace;
boundary=frame')

if __name__ == '__main__':
    app.run(debug=True)
```

Система сповіщень (файл notification_system.py)

```
import smtplib
from email.mime.text import MIMEText

# Модуль для сповіщення через email
def send_email_notification(subject, message, to_email):
    from_email = "your_email@gmail.com"
    password = "your_password"

    msg = MIMEText(message)
    msg['Subject'] = subject
    msg['From'] = from_email
    msg['To'] = to_email

    with smtplib.SMTP_SSL("smtp.gmail.com", 465) as server:
        server.login(from_email, password)
        server.sendmail(from_email, to_email, msg.as_string())

# Інтеграція сповіщень у систему відеоаналітики
def process_video_with_notifications(source=0):
    cap = video_capture(source)

    if cap is None:
        return

    ret, prev_frame = cap.read()
    if not ret:
        print("Не вдалося отримати перший кадр")
        return

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

        motion_detected = detect_motion(frame, prev_frame)

        if motion_detected:
            send_email_notification("Motion Detected", "Motion detected in the
            video stream", "recipient@example.com")

        cv2.imshow("Notification System", frame)

        if cv2.waitKey(30) & 0xFF == ord('q'):
            break

    cap.release()
    cv2.destroyAllWindows()
```

Підтримка декількох потоків відео (файл multi_camera_support.py)

```
import cv2

# Модуль для обробки декількох камер
def process_multiple_cameras(sources):
    caps = [video_capture(src) for src in sources]

    while all([cap.isOpened() for cap in caps]):
        for i, cap in enumerate(caps):
            ret, frame = cap.read()
            if not ret:
                continue

            cv2.imshow(f"Camera {i+1}", frame)

            if cv2.waitKey(30) & 0xFF == ord('q'):
                break

    for cap in caps:
        cap.release()

    cv2.destroyAllWindows()
```

КБПЗ_2024

Оптимізація для мобільних пристроїв (файл mobile_optimization.py)

```
import cv2
import tensorflow as tf

# Модуль для мобільної оптимізації
def optimize_for_mobile(model_path):
    converter = tf.lite.TFLiteConverter.from_saved_model(model_path)
    tflite_model = converter.convert()

    with open("model_optimized.tflite", "wb") as f:
        f.write(tflite_model)

# Інтеграція оптимізованої моделі у відеоаналітику
def process_video_with_mobile_optimized_model(source=0,
model_path="model_optimized.tflite"):
    cap = video_capture(source)

    if cap is None:
        return

    interpreter = tf.lite.Interpreter(model_path=model_path)
    interpreter.allocate_tensors()

    input_details = interpreter.get_input_details()
    output_details = interpreter.get_output_details()

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

        resized_frame = cv2.resize(frame, (224, 224))
        input_data = np.expand_dims(resized_frame, axis=0)

        interpreter.set_tensor(input_details[0]['index'], input_data)
        interpreter.invoke()

        output_data = interpreter.get_tensor(output_details[0]['index'])
        class_index = np.argmax(output_data)

        cv2.putText(frame, f"Class: {class_index}", (10, 30),
cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 2)
        cv2.imshow("Mobile Optimized Model", frame)

        if cv2.waitKey(30) & 0xFF == ord('q'):
            break

    cap.release()
    cv2.destroyAllWindows()
```

Виявлення змін у статичному середовищі (файл environment_change_detection.py)

```
import cv2
import numpy as np

# Модуль для виявлення змін у статичному середовищі
def detect_environment_change(static_background, current_frame, threshold=30):
    diff_frame = cv2.absdiff(static_background, current_frame)
    _, thresh_frame = cv2.threshold(diff_frame, threshold, 255,
cv2.THRESH_BINARY)
    return thresh_frame

# Інтеграція з основною системою для виявлення змін
def process_video_with_environment_change_detection(source=0):
    cap = video_capture(source)

    if cap is None:
        return

    ret, static_background = cap.read()
    if not ret:
        print("Не вдалося отримати перший кадр")
        return

    static_background = preprocess_frame(static_background)

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

        current_frame = preprocess_frame(frame)
        change_detected = detect_environment_change(static_background,
current_frame)

        cv2.imshow("Environment Change Detection", change_detected)

        if cv2.waitKey(30) & 0xFF == ord('q'):
            break

    cap.release()
    cv2.destroyAllWindows()
```

Підтримка розпізнавання облич у масках (файл `masked_face_detection.py`)

```
import cv2

# Модуль для виявлення облич у масках
def detect_masked_faces(frame, face_cascade, mask_cascade):
    gray_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    faces = face_cascade.detectMultiScale(gray_frame, scaleFactor=1.1,
minNeighbors=5, minSize=(30, 30))
    masks = mask_cascade.detectMultiScale(gray_frame, scaleFactor=1.1,
minNeighbors=5, minSize=(30, 30))

    for (x, y, w, h) in faces:
        cv2.rectangle(frame, (x, y), (x + w, y + h), (255, 0, 0), 2)

    for (x, y, w, h) in masks:
        cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)
        cv2.putText(frame, "Mask Detected", (x, y - 10),
cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 2)

    return frame

# Інтеграція з основною системою для виявлення облич у масках
def process_video_with_masked_face_detection(source=0,
face_cascade_path="haarcascade_frontalface_default.xml",
mask_cascade_path="haarcascade_mask.xml"):
    face_cascade = cv2.CascadeClassifier(face_cascade_path)
    mask_cascade = cv2.CascadeClassifier(mask_cascade_path)

    cap = video_capture(source)

    if cap is None:
        return

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

        frame_with_faces = detect_masked_faces(frame, face_cascade,
mask_cascade)
        cv2.imshow("Masked Face Detection", frame_with_faces)

        if cv2.waitKey(30) & 0xFF == ord('q'):
            break

    cap.release()
    cv2.destroyAllWindows()
```

**Вбудований модуль навчання для адаптації системи до нових умов
(файл adaptive_learning.py)**

```
import cv2
import numpy as np
from sklearn.svm import SVC
from sklearn.preprocessing import StandardScaler
from joblib import dump, load

# Модуль для навчання моделі на нових даних
def train_model(new_data, labels, model_path="adaptive_model.joblib"):
    scaler = StandardScaler()
    scaled_data = scaler.fit_transform(new_data)

    model = SVC()
    model.fit(scaled_data, labels)

    dump((scaler, model), model_path)

# Модуль для інтеграції навчання у відеоаналітику
def process_video_with_adaptive_learning(source=0,
model_path="adaptive_model.joblib"):
    scaler, model = load(model_path)

    cap = video_capture(source)
    if cap is None:
        return

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

        frame_features = extract_frame_features(frame)
        scaled_features = scaler.transform([frame_features])
        prediction = model.predict(scaled_features)

        cv2.putText(frame, f"Prediction: {prediction[0]}", (10, 30),
cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 2)
        cv2.imshow("Adaptive Learning", frame)

        if cv2.waitKey(30) & 0xFF == ord('q'):
            break

    cap.release()
    cv2.destroyAllWindows()

# Модуль для збирання даних для навчання
def collect_training_data(source=0, label=None, num_samples=100):
    cap = video_capture(source)
    if cap is None:
        return
```

```
collected_data = []
labels = []

for _ in range(num_samples):
    ret, frame = cap.read()
    if not ret:
        break

    frame_features = extract_frame_features(frame)
    collected_data.append(frame_features)
    labels.append(label)

    cv2.imshow("Collecting Training Data", frame)

    if cv2.waitKey(30) & 0xFF == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()

return np.array(collected_data), np.array(labels)
```

K6П3_2024

**Інтеграція з системами розпізнавання емоцій обличчя (файл
emotion_recognition.py)**

```
import cv2
from deepface import DeepFace

# Модуль для розпізнавання емоцій на основі обличчя
def recognize_emotions(frame):
    try:
        emotion_analysis = DeepFace.analyze(frame, actions=['emotion'],
enforce_detection=False)
        dominant_emotion = emotion_analysis['dominant_emotion']
        return dominant_emotion
    except Exception as e:
        print(f"Error: {str(e)}")
        return "No face detected"

# Інтеграція з основною системою для розпізнавання емоцій
def process_video_with_emotion_recognition(source=0):
    cap = video_capture(source)

    if cap is None:
        return

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

        dominant_emotion = recognize_emotions(frame)
        cv2.putText(frame, f"Emotion: {dominant_emotion}", (10, 30),
cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 2)
        cv2.imshow("Emotion Recognition", frame)

        if cv2.waitKey(30) & 0xFF == ord('q'):
            break

    cap.release()
    cv2.destroyAllWindows()
```

Оптимізація коду для роботи з відео високої роздільної здатності (файл
high_resolution_optimization.py)

```
import cv2

# Модуль для оптимізації роботи з відео високої роздільної здатності
def process_high_resolution_video(frame, scale_percent=50):
    width = int(frame.shape[1] * scale_percent / 100)
    height = int(frame.shape[0] * scale_percent / 100)
    resized_frame = cv2.resize(frame, (width, height))
    return resized_frame

# Інтеграція з основною системою для обробки відео високої роздільної здатності
def process_video_with_high_resolution_optimization(source=0, scale_percent=50):
    cap = video_capture(source)

    if cap is None:
        return

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

        optimized_frame = process_high_resolution_video(frame, scale_percent)
        cv2.imshow("High Resolution Optimization", optimized_frame)

        if cv2.waitKey(30) & 0xFF == ord('q'):
            break

    cap.release()
    cv2.destroyAllWindows()
```