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
“____” _____ 2025 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за другим (магістерським) рівнем вищої освіти
на тему
“Дослідження та програмна реалізація системи комплексних інтелектуальних рішень для відеонагляду”

Виконав здобувач вищої освіти
ІІ курсу, групи КН-24М
ОПП «Комп’ютерні науки»
спеціальності 122 «Комп’ютерні науки»
_____ Олексієнко С.В.
«____» _____ 2025 р.

Керівник проекту
кандидат технічних наук
_____ Лисенко І.А.
«____» _____ 2025 р.
Рецензент _____
_____

м. Кропивницький

АНОТАЦІЯ

Олексієнко С.В. Дослідження та програмна реалізація системи комплексних інтелектуальних рішень для відеонагляду. 122 Комп'ютерні науки. Центральноукраїнський національний технічний університет. Кропивницький. 2025.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи комплексних інтелектуальних рішень для відеонагляду.

Метою розробки є дослідження та програмна реалізація системи комплексних інтелектуальних рішень для відеонагляду.

Об'єктом дослідження є процес комплексних інтелектуальних рішень для відеонагляду.

Предметом дослідження є методи комплексних інтелектуальних рішень для відеонагляду.

Методи дослідження базуються на методах розпізнавання образів, методах великих даних, методах комп'ютерних мереж,, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація системи комплексних інтелектуальних рішень для відеонагляду.

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'ютерні науки, відеонагляд

ABSTRACT

Oleksienko S.V. Research and software implementation of a system of complex intelligent solutions for video surveillance. 122 Computer Science. Central Ukrainian National Technical University. Kropyvnytskyi. 2025.

In this final qualification work for the second (master's) level of higher education, software has been developed, which is intended for a system of complex intelligent solutions for video surveillance.

The purpose of the development is the research and software implementation of a system of complex intelligent solutions for video surveillance.

The object of the research is the process of complex intelligent solutions for video surveillance.

The subject of the research is the methods of complex intelligent solutions for video surveillance.

The research methods are based on pattern recognition methods, big data methods, computer network methods, mathematical statistics methods, software development methods.

The result of the work is a software implementation of a system of complex intelligent solutions for video surveillance.

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 user-friendly user interface has been developed. Instructions for working with the software are provided.

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

The program was developed in the Python environment.

Keywords: computer science, video surveillance

ЗМІСТ

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
IP	–	Internet Protocol. Протокол міжмережної взаємодії
PIN-код	–	персональний ідентифікаційний код
TCP	–	Transmission Control Protocol. Протокол керування передачею (даних) Основний транспортний протокол у стеці протоколів TCP/IP. Встановлює зв'язок між двома ПК й організує обмін даними в дуплексному режимі
TCP/IP	–	Transmission Control Protocol/Internet Protocol. Стек протоколів, що забезпечують організацію зв'язку між комп'ютерами в мережі Інтернет

ВСТУП

Актуальність теми. У кожній галузі платформи відеоспостереження стали важливими інструментами для забезпечення безпеки, моніторингу активів та підвищення операційної ефективності. Нещодавні технологічні досягнення збільшили попит на інтелектуальні, масштабовані та хмарні рішення для спостереження. Багато організацій зараз переходять на платформи на базі штучного інтелекту, які пропонують розширені можливості, такі як розпізнавання облич, виявлення об'єктів та складна аналітика даних. Новіші системи покращують виявлення загроз, прискорюють час реагування та мінімізують людські помилки завдяки автоматизації. Вибір нової платформи відеоспостереження являє собою значну інвестицію, яка має наслідки для організаційної безпеки та бізнес-аналітики. Індустрія відеоспостереження постійно розвивається, а штучний інтелект та хмарні обчислення відіграють вирішальну роль у сучасних рішеннях безпеки. Незалежно від того, чи шукаєте ви повністю хмарну систему на базі штучного інтелекту, чи простіше локальне рішення, варіанти відеонагляду задовольнять будь-які потреби безпеки. Вибір правильної платформи залежить від ваших конкретних вимог, бюджету та рівня інтеграції, який ви шукаєте для своєї інфраструктури спостереження.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи комплексних інтелектуальних рішень для відеонагляду.

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

– Огляд існуючих систем комплексних інтелектуальних рішень для відеонагляду.

– Дослідження системи комплексних інтелектуальних рішень для відеонагляду.

– Програмна реалізація системи комплексних інтелектуальних рішень для відеонагляду.

Об'єктом дослідження є процес комплексних інтелектуальних рішень для відеонагляду.

Предметом дослідження є методи комплексних інтелектуальних рішень для відеонагляду.

Методи дослідження базуються на методах розпізнавання образів, методах великих даних, методах комп'ютерних мереж,, методах математичної статистики, методах розробки програмного забезпечення.

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

– Удосконалено метод комплексних інтелектуальних рішень для відеонагляду.

– Розроблено вітчизняний продукт комплексних інтелектуальних рішень для відеонагляду, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі комплексних інтелектуальних рішень для відеонагляду.

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.

Робота апробована на LVII Науково-технічній конференції здобувачів вищої освіти LV науково-технічної конференції «Наука в ЦНТУ: основні досягнення та перспективи розвитку» (2025 р.), основні положення випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти надруковані у даній роботі збірника праць молодих науковців ЦНТУ, випуск №15.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи комплексних інтелектуальних рішень для відеонагляду, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Швидкий розвиток технологій спостереження значною мірою покращив безпеку в громадських місцях. Найбільш перспективним з цих розробок є поєднання методів розпізнавання образів із системами відеоспостереження. У наступній даній роботі представлено кілька методів розпізнавання образів, що використовуються для підвищення ефективності та точності систем безпеки в громадських місцях. Зі зростанням кількості камер спостереження, що встановлюються по всьому світу, виникає потреба в складних аналітичних інструментах для аналізу величезних обсягів створюваних даних.

Традиційні системи моніторингу безпеки не дуже ефективні в обробці відеоданих у режимі реального часу, що призводить до неефективності. Використання розпізнавання образів, такого як машинне навчання, глибоке навчання та комп'ютерний зір, дозволяє виявляти підозрілу поведінку, незвичайні закономірності та потенційні загрози.

У даній роботі розглядаються деякі найважливіші методи, такі як розпізнавання облич, відстеження руху, виявлення аномалій та аналіз поведінки натовпу. Також обговорюється вплив цих технологій на результати безпеки в громадському просторі, а також наслідки для конфіденційності та етичних питань.

Крім того, у даній роботі досліджуються масштабованість, проблеми та майбутні напрямки розвитку цієї галузі. Нарешті, у ній наголошується на тому, як за допомогою систем розпізнавання образів можна досягти запобігання злочинам, швидшого реагування на надзвичайні ситуації та загального підвищення безпеки громадських просторів.

1.2 Область застосування

Для задоволення різних випадків використання в різних галузях промисловості можуть бути розроблені індивідуальні типи та комбінації інструментів відеоаналітики. Власники бізнесу, фахівці та команди безпеки можуть використовувати готові системи для вирішення загальних потреб безпеки та управління організацією або інвестувати у створення індивідуальних рішень для задоволення унікальних галузевих вимог.

Нижче наведено приклади реальних випадків використання відеоаналітики та ключових галузевих застосувань.

Охорона здоров'я

Рішення для відеоаналітики допомагають медичним працівникам створювати безпечні умови для пацієнтів та персоналу. Системи розпізнавання облич можуть допомогти персоналу ідентифікувати осіб, які могли спричиняти проблеми в минулому. Водночас лікарняні камери з функцією виявлення об'єктів можна налаштувати для попередження служб безпеки про контрабанду, а функції виявлення зброї можуть виявляти потенційну вогнепальну зброю.

Також можна створювати галузеві системи відеоаналітики. Наприклад, програми можна навчити виявляти тривожні моделі рухів, пов’язані з падіннями та медичними подіями, що дозволить персоналу налаштовувати системи миттєвого оповіщення для вразливих пацієнтів. Відеоаналітика, така як технології відстеження об’єктів, також може допомогти персоналу забезпечити прийом ліків згідно з призначенням лікаря.

Транспорт

Камери відеоаналітики, що охоплюють дороги та перехрестя, використовуються для покращення управління дорожнім рухом у деяких сучасних містах. Відеоаналітика на основі штучного інтелекту може сканувати номерні знаки, щоб визначити, скільки автомобілів перебуває на дорозі в будь-який момент часу, і ці дані можна використовувати для інформування про роботу інфраструктури, такої як світлофори та мережі громадського транспорту, допомагаючи зменшити затори.

Камери моніторингу дорожнього руху, оснащені відеоаналітикою, також можуть бути використані для підвищення безпеки дорожнього руху, оскільки системи штучного інтелекту здатні виявляти потенційно небезпечні інциденти в режимі реального часу. Системи можуть виявляти автомобілі, що зупиняються в небезпечних зонах, рухаються хаотично або в неправильному напрямку, а результати оперативно надсилають відповідним органам для забезпечення швидкого та належного реагування.

Роздрібна торгівля

Системи відеоаналітики безпеки допомагають роздрібним торговцям розробляти ефективні заходи проти крадіжок та стратегії запобігання втратам, щоб захистити людей і майно. Аналітику розпізнавання облич з можливостями машинного навчання можна навчити автоматично ідентифікувати відомих крадіїв у магазинах, а також виявляти закономірності в поведінці та підозрілі рухи, пов'язані з попередніми випадками крадіжок.

Відеоаналітику в роздрібній торгівлі також можна використовувати, щоб допомогти роздрібним торговцям краще зрозуміти звички клієнтів. Наприклад, системи можуть аналізувати, як клієнти переміщуються магазином, щоб допомогти менеджерам визначити найкращі місця для певних товарів. Іноді відеоаналітику можна використовувати для збору демографічних даних клієнтів, які можна використовувати для впливу на маркетингові зусилля.

Розумні міста

Існує кілька варіантів використання відеоаналітики в розумних містах, де спеціально розроблені системи покращують усе: від безпеки до ефективності інфраструктури. Камери відеоаналітики, оснащені функціями виявлення натовпу, відстеження об'єктів та моніторингу присутності, можуть бути використані для виявлення ризиків безпеки, зменшення заторів та впливу на патрулі правоохоронних органів.

Відеоаналітику ALPR також можна використовувати для розробки автоматизованих систем управління паркуванням, допомагаючи громадянам з транспортними засобами ефективніше пересуватися розумними містами. Дані відеоаналітики також можна використовувати в ширших системах, таких як

рішення для управління відходами та планування транспорту, щоб допомогти управлінським командам планувати ефективні маршрути та практичні операції.

Будівництво

У поєднанні із системою відеоаналітики камери безпеки на будівельному майданчику можуть контролювати поведінку працівників, виявляти потенційні небезпеки та забезпечувати дотримання правил безпеки в режимі реального часу. Це допомагає відстежувати використання обладнання, контролювати хід робіт на будівельному майданчику та запобігати крадіжкам або несанкціонованому доступу, розпізнаючи незвичайну діяльність. Розширені алгоритми також можуть аналізувати моделі робочих процесів для оптимізації продуктивності та зменшення часу простою.

Виробництво

Камери відеоаналітики, встановлені для охоплення завантажених виробничих ліній, можна навчити виявляти проблеми у виробничих операціях. Програмне забезпечення зі штучним інтелектом може виявляти аномалії в сировині та товарах, придатних для продажу, щоб допомогти підприємствам покращити операції з контролю якості, а також постійно спостерігати за машинами та обладнанням, щоб допомогти в плануванні простоїв та технічного обслуговування.

Ініціативи щодо безпеки персоналу також можна покращити за допомогою відеоаналітики. Рішення для виробничого відеоспостереження можна навчити виявляти анонімні та потенційно небезпечні події, такі як неправильне використання персоналом обладнання або неносіння захисного спорядження, що дозволить менеджерам оперативно реагувати на потенційні ризики для безпеки та надавати точну інформацію, щоб допомогти обмежити вплив проблем безпеки.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи комплексних інтелектуальних рішень для відеонагляду, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

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

1. Lumana

Lumana зарекомендувала себе як лідер галузі завдяки своїй хмарній системі відеоспостереження на базі штучного інтелекту. Lumana, що вирізняється інтуїтивно зрозумілим інтерфейсом користувача, безперебійною інтеграцією з Інтернетом речей та розширеною відеоаналітикою, забезпечує майже людське сприйняття, допомагаючи командам бачити критичні події, розуміти повний контекст будь-якої ситуації та реагувати з неперевершеною швидкістю та точністю. Її можливості виявлення загроз у режимі реального часу та автоматизованого реагування встановлюють новий стандарт у галузі відеоспостереження, усуваючи необхідність ручного моніторингу та пропонуючи цінні дані для бізнес-аналітики.

Основні характеристики:

– Автоматизоване реагування на інциденти: ініціює заздалегідь визначені протоколи безпеки при виявленні загроз, від блокування дверей до ввімкнення сигналізації та сповіщення служб безпеки.

– Розширене розпізнавання обличчя: ідентифікує людей з високою точністю навіть за складних умов освітлення.

– Виявлення поведінкових аномалій: використовує машинне навчання для виявлення незвичайної активності та автоматичного сповіщення персоналу служби безпеки до ескалації інцидентів.

– Віддалене керування кількома місцями розташування: Забезпечує централізоване керування системами безпеки на кількох об'єктах через єдину, безпечну хмарну панель керування.

Переваги:

– Аналітика на базі штучного інтелекту та сповіщення в режимі реального часу.

– Покращена організаційна видимість.

– Розширений пошук дозволяє аналізувати мільйони годин відеозаписів за лічені секунди.

Недоліки:

– Налаштування може бути більш технічним залежно від випадку використання.

– Розгортання користувацької аналітики штучного інтелекту може зайняти деякий час.

2. Milestone

Milestone пропонує програмне забезпечення для керування відео (VMS), розроблене з архітектурою відкритої платформи. XProtect VMS платформи створена для масштабування для підприємств будь-якого розміру, від малих підприємств до великих підприємств. Хоча впровадження вимагає технічних знань, користувачі загалом високо оцінюють її комплексний набір функцій та надійність.

Основні характеристики:

– Архітектура відкритої платформи: підтримує понад 10 000 камер та пристроїв від більш ніж 150 виробників для гнучкої інтеграції обладнання.

– Централізована система управління: керує необмеженою кількістю камер на кількох об'єктах з одного інтерфейсу.

– Інтерактивні карти та візуалізація: відображає розташування камер та події в режимі реального часу на динамічних картах об'єктів для кращої обізнаності.

– Налаштовуваний механізм правил: Забезпечує складні автоматизовані відповіді на основі складних умовних тригерів від інтегрованих систем безпеки.

Переваги:

– Масштабований та гнучкий.

– Потужні інтеграції зі сторонніми розробниками.

– Велика команда підтримки клієнтів.

Недоліки:

– Потрібні технічні знання для налаштування та використання.

– Може бути дорогим для малого бізнесу.

3. Genetec

Genetec є піонером у сфері уніфікованих рішень безпеки, що інтегрує відеоспостереження, контроль доступу та аналітику. Її платформа Security Center забезпечує комплексне управління безпекою для підприємств та державних установ. Завдяки надійним протоколам кібербезпеки та відеоаналітиці на базі штучного інтелекту, Genetec приваблює компанії, які шукають комплексне рішення безпеки.

Основні характеристики:

– Уніфікована платформа безпеки: інтегрує відео, контроль доступу та розпізнавання номерних знаків в єдину систему для комплексного управління об'єктами.

– Розширений захист конфіденційності: пропонує автоматизовані інструменти редагування, сертифіковані європейськими органами захисту даних, для забезпечення відповідності GDPR без шкоди для безпеки.

– Управління рівнями загроз: дозволяє організаціям визначати та активувати різні протоколи безпеки залежно від зростаючих загроз.

– Наскрізне шифрування: використовує шифрування військового рівня на всій платформі, від підключень камер до збережених відеоматеріалів та експортованих доказів.

Переваги:

– Єдина екосистема безпеки.

– Відеоаналітика на базі штучного інтелекту.

– Надійні протоколи кібербезпеки.

Недоліки:

– Висока вартість корпоративних рішень.

– Складність розгортання.

4. Verkada

Verkada запровадила гібридний хмарний підхід, пропонуючи потужну аналітику на основі штучного інтелекту та зручний інтерфейс. Модель розгортання plug-and-play платформи підходить для організацій будь-якого розміру. Видатні функції Verkada включають сповіщення та складне розпізнавання облич, хоча її закрита екосистема обмежує можливості інтеграції.

Основні характеристики:

– Аналітика людей і транспортних засобів: надає детальну інформацію про переміщення людей і транспортних засобів, включаючи точний підрахунок, виявлення натовпу та управління паркуванням.

– Інтеграція датчиків навколишнього середовища: Включає датчики якості повітря, температури, вологості та руху для комплексного моніторингу об'єктів, що виходить за рамки відеоспостереження.

– Командний центр: Централізує керування необмеженою кількістю камер із можливостями швидкого пошуку, які дають результати за лічені хвилини, а не за години.

– Безпечний віддалений доступ: дозволяє авторизованим користувачам переглядати відеозаписи з будь-якого пристрою, забезпечуючи безпеку завдяки двофакторній автентифікації та детальним журналам аудиту.

Переваги:

– Гібридне хмарне сховище для підвищеної безпеки.

– Інтегрована відеоаналітика.

– Легко розгортати та масштабувати.

Недоліки::

Обмежена інтеграція зі сторонніми розробниками

Обмежені можливості штучного інтелекту

5. Avigilon

Як компанія Motorola Solutions, Avigilon здобула репутацію завдяки сильним сторонам у сфері відеоаналітики та обладнання для спостереження. Її комплексне рішення включає камери з покращеним штучним інтелектом та пропозиції VMS. Avigilon перевершує інших у сфері зображення високої роздільної здатності, що робить її популярним вибором для застосувань, що вимагають високої чіткості відео.

Основні характеристики:

– Відеоаналітика: Виявлення об'єктів дозволяє швидко знаходити певних осіб або транспортні засоби по всьому об'єкту.

– Зображення високої чіткості: Забезпечує камери з роздільною здатністю до 30 МП, що забезпечують виняткову деталізацію для точної ідентифікації на значних відстанях.

– Виявлення аномалій: використовує штучний інтелект для виявлення критичних подій у мережі спостереження, спрямовуючи увагу оператора туди, де це найбільше потрібно.

– Платформа камер H5: оснащена вбудованими процесорами штучного інтелекту, які виконують аналітику на периферії, зменшуючи споживання пропускної здатності та вимоги до сервера.

Переваги:

– Камери високої роздільної здатності з можливостями штучного інтелекту.

– Відмінні функції судово-медичного відео пошуку.

– Сильна інтеграція апаратного та програмного забезпечення.

Недоліки:

– Потрібне власне обладнання для використання повного набору функцій.

– Високі початкові витрати.

6. Rhombus

Rhombus спеціалізується на хмарному відеоспостереженні, пропонуючи розумні камери з аналітикою в режимі реального часу та віддаленим моніторингом. Завдяки розпізнаванню облич, контролю доступу та датчикам навколишнього середовища, Rhombus може розширюватися відповідно до потреб організації.

Основні характеристики:

– Інтеграція датчиків: поєднує відеодані з датчиками навколишнього середовища для моніторингу руху, якості повітря, температури та присутності.

– Контроль доступу: пропонує системи контролю доступу для поєднання зчитування бейджів із відеоверифікацією для безпеки входу.

– Автоматизоване управління зайнятістю: моніторить зайнятість у режимі реального часу для забезпечення відповідності нормам місткості.

– Виявлення незвичайної активності: позначає відхилення від типових моделей поведінки для розслідування та перегляду.

Переваги:

– Повністю керовано хмарою.

– Автоматизація на базі штучного інтелекту.

– Відповідність вимогам безпеки.

Недоліки:

– Немає локальної опції.

– Потрібне потужне інтернет-з'єднання.

7. Axis

Компанія Axis зберігає свою позицію визнаного лідера ринку, відомого своїми преміальними IP-камерами та мережевими рішеннями. Її платформа відеоспостереження бездоганно інтегрується з різними системами безпеки.

Продукцію Axis часто обирають за її довговічність та чудову якість відео, що робить її популярною у великомасштабних операціях відеоспостереження, таких як моніторинг міст та транспортних вузлів.

Основні характеристики:

– Технологія Zipstream: Зменшує вимоги до пропускної здатності та сховища до 50%, зберігаючи при цьому важливі деталі відеоматеріалів.

– Аналітика на периферії: Вбудовані потужні процесори для виявлення на пристрої, що зменшують вимоги до центрального сервера.

– AXIS Camera Station: Спрощує щоденні операції безпеки завдяки оптимізованому інтерфейсу, розробленому для ефективного моніторингу та розслідування.

– Продуктивність за умов слабкого освітлення: За допомогою технології Lightfinder можна знімати кольорові зображення майже в повній темряві, балансуючи експозицію в складних умовах освітлення.

Переваги:

– Преміальне обладнання для камери.

– Широкий спектр інтеграцій.

– Надійний та масштабований.

Недоліки::

– Дорого, з високими початковими витратами.

– Крива навчання для нових користувачів.

8. Eagle Eye

Eagle Eye Networks – провідний постачальник хмарних послуг відеоспостереження, що пропонує кібербезпеку та аналітику на основі штучного інтелекту на масштабованій платформі. Eagle Eye особливо популярна завдяки своєму підходу, орієнтованому на мобільні пристрої, який забезпечує безперешкодний доступ до відеоматеріалів у реальному часі та записаних відео через мобільні пристрої.

Основні характеристики:

– Керування пропускною здатністю: Автоматично налаштовує якість відео на основі доступних мережевих ресурсів, забезпечуючи високу якість відео, коли це необхідно для розслідувань.

– Відстеження об'єктів між камерами: Відстежує людей або транспортні засоби під час їхнього переміщення по об'єкту через зображення з кількох камер для комплексного моніторингу.

– Гнучка архітектура сховища: поєднує хмарне сховище, локальний запис та гібридні підходи на основі конкретних вимог організації.

– Відкрита платформа API: забезпечує глибоку інтеграцію з бізнес-системами, платформами контролю доступу та користувацькими додатками за допомогою комплексних інструментів для розробників.

Переваги:

– 100% хмарне управління.

– Аналітика на основі штучного інтелекту та розпізнавання облич.

– Надійне шифрування та безпека даних.

Недоліки:

– Застаріле програмне забезпечення та складне у використанні.

– Потрібне стабільне інтернет-з'єднання.

9. DW Spectrum

Цифровий сторожовий пристрій (DW) Spectrum пропонує інтуїтивно зрозумілу, масштабовану систему керування відео з потужними аналітичними можливостями. Він відомий своєю надійністю в корпоративних розгортаннях і підтримує широкий спектр камер, що робить його гнучким варіантом для компаній, яким потрібне індивідуальне рішення VMS.

Основні характеристики:

– Клієнт для різних платформ: забезпечує узгоджену роботу з користувачами на різних платформах: Windows, Mac, Linux та мобільних пристроях, що забезпечує гнучкий доступ до системи.

– Інтелектуальне виявлення руху: мінімізує хибні тривоги завдяки вдосконаленій фільтрації, яка розрізняє відповідний рух та фактори навколишнього середовища.

– Віртуальний огляд камери: Забезпечує ефективний моніторинг великих територій за допомогою запрограмованих послідовностей перегляду на основі налаштовуваних шаблонів.

– Інтеграція контролю доступу: Зв'язується з провідними платформами контролю доступу для створення єдиного досвіду безпеки з корельованими подіями.

Переваги:

– Простий та інтуїтивно зрозумілий інтерфейс.

– Підтримує широкий спектр камер.

– Масштабована модель ліцензування.

Недоліки:

– Бракує деяких розширених функцій штучного інтелекту.

– Потрібне ручне налаштування для розширених параметрів.

10. Videoloft

Videoloft спеціалізується на хмарному відеоспостереженні для малого та середнього бізнесу. Платформа забезпечує просту хмарну інтеграцію з існуючою інфраструктурою відеоспостереження, пропонуючи економічний шлях оновлення для застарілих систем.

Основні характеристики:

– Технологія хмарного адаптера: підключається до існуючих відеореєстраторів та мережевих відеореєстраторів, що забезпечує хмарне сховище та віддалений доступ без заміни поточної інфраструктури камер.

– Варіанти багаторівневого зберігання: пропонує налаштовувані політики зберігання для кожної камери, що дозволяє організаціям визначати пріоритети зберігання на основі вимог безпеки.

– Мобільне керування: Забезпечує комплексне керування системою за допомогою інтуїтивно зрозумілих додатків для смартфонів із push-сповіщеннями про події руху.

– Адміністрування кількох сайтів: спрощує керування розподіленими локаціями завдяки єдиному інтерфейсу з детальним контролем дозволів.

Переваги:

– Доступне рішення для хмарного зберігання даних.

– Працює з існуючим обладнанням для відеоспостереження.

– Легко налаштувати та керувати.

Недоліки:

– Обмежена розширена аналітика.

– Залежить від підключення до Інтернету.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Як мова програмування обрана Python. Python – високорівнева мова програмування загального призначення з акцентом на продуктивність розроблювача й читаність коду. Синтаксис ядра Python мінімалістичний. У той же час стандартна бібліотека включає великий обсяг корисних функцій.

Python підтримує кілька парадигм програмування, у тому числі структурне, об'єктно-орієнтоване, функціональне, імперативне й аспектно-орієнтоване. Основні архітектурні риси – динамічна типізація, автоматичне керування пам'яттю, повна інтроспекція, механізм обробки виключень, підтримка багатопоточні обчислень і зручні високорівневі структури даних. Код у Python організовується у функції й класи, які можуть поєднуватися в модулі (які у свою чергу можуть бути об'єднані в пакети).

Еталонною реалізацією Python є інтерпретатор CPython, що підтримує більшість активно використовуваних платформ. Він поширюється вільно під

дуже ліберальною ліцензією, що дозволяє використовувати його без обмежень у будь-яких застосунках, включаючи пропрієтарні. Є реалізації інтерпретаторів для JVM (з можливістю компіляції), MSIL (з можливістю компіляції), LLVM і інших. Проект PyPy пропонує реалізацію Python на самому Python, що зменшує витрати на зміни мови й постановку експериментів над новими можливостями.

Python – мова програмування, що активно розвивається, нові версії (з додаванням/зміною язикових властивостей) виходять приблизно раз у два з половиною року. Внаслідок цього й деяких інших причин на Python відсутні ANSI, ISO або інші офіційні стандарти, їхня роль виконує CPython.

Python портований і працює майже на всіх відомих платформах – від КПК до мейнфреймів. Існують порти під Microsoft Windows, практично всі варіанти UNIX (включаючи FreeBSD і Linux), Plan 9, Mac OS і Mac OS X, iPhone OS 2.0 і вище, Palm OS, OS/2, Amiga, AS/400 і навіть OS/390, Symbian і Android.

При цьому, на відміну від багатьох портуємих систем, для всіх основних платформ Python має підтримку характерних для даної платформи технологій (наприклад, Microsoft COM/DCOM). Більше того, існує спеціальна версія Python для віртуальної машини Java – Jython, що дозволяє інтерпретаторові виконуватися на будь-якій системі, що підтримує Java, при цьому класи Java можуть безпосередньо використовуватися з Python й навіть бути написаними на Python. Також кілька проектів забезпечують інтеграцію із платформою Microsoft.NET, основні з яких – IronPython і Python.Net.

Python підтримує динамічну типізацію, тобто тип змінної визначається тільки під час виконання. Тому замість «присвоювання значення змінної» краще говорити про «зв'язування значення з деяким ім'ям». У Python є убудовані типи: бульові, рядки, Unicode-рядки, цілі числа довільної точності, числа із плаваючої коми, комплексні числа й деякі інші. З колекцій Python підтримує кортежі (*tuples*), списки, словники (асоціативні масиви) і, починаючи з версії 2.4, безлічі. Всі значення в Python є об'єктами, у тому числі функції, методи, модулі, класи.

Додати новий тип можна або написавши клас (class), або визначивши новий тип у модулі розширення (наприклад, написаному мовою С). Система класів підтримує спадкування (одиночне й множинне) і метапрограмування. Можливе спадкування від більшості убудованих типів і типів розширень.

Всі об'єкти діляться на посилальні й атомарні. До атомарного ставляться int, long, complex і деякі інші. При присвоюванні атомарних об'єктів копіюється їхнє значення, у той час як для посилальних копіюється тільки покажчик на об'єкт, таким чином, обидві змінні після присвоювання використовують те саме значення. Посилальні об'єкти бувають змінювані й незмінні. Наприклад, рядки й кортежі є незмінними, а списки, словники й багато інших об'єктів – змінюваними. Кортеж у Python є, по суті, незмінним списком. У багатьох випадках кортежі працюють швидше списків, тому якщо ви не плануєте змінювати послідовність, то краще використовувати саме їх.

Мова має чіткий і послідовний синтаксис, продуману модульність й масштабованість, завдяки чому вихідний код написаних на Python програм легко читаємий.

Python – стабільна й розповсюджена мова. Він використовується в багатьох проектах і в різних якостях: як основна мова програмування або для створення розширень і інтеграції застосунків. На Python реалізоване велика кількість проектів, також він активно використовується для створення прототипів майбутніх програм. Python використовується в багатьох великих компаніях.

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускну кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи комплексних інтелектуальних рішень для відеонагляду.

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) сформувати висновки про виконаний обсяг робіт та одержані результати.

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Більшість систем безпеки містять камери відеоспостереження, які допомагають службам безпеки виявляти потенційні загрози. Однак, чим більшою та складнішою стає установка, тим складніше може бути забезпечити належне спостереження за всіма активними камерами та потоками спостереження.

З появою інтелектуальних технологій, таких як штучний інтелект та машинне навчання, сучасні системи спостереження можна запрограмувати на автоматичне виявлення аномальних подій та сигналів безпеки, що допомагає командам зосередити свої зусилля на подіях, що розгортаються, та питаннях негайної важливості.

Це основна передумова відеоаналітики. Ця передова технологія може самостійно аналізувати відеоконтент та отримувати з нього корисну інформацію для покращення прийняття рішень та підвищення ефективності реагування систем безпеки. Щоб дізнатися більше про практичні можливості аналітики відеоспостереження, нижче наведено повний посібник із застосування, можливостей та варіантів використання відеоаналітики.

Відеоаналітика – це передова технологія, яка автоматично аналізує контент, записаний відеокамерами. Інтелектуальні алгоритми обробляють відеодані в режимі реального часу, щоб генерувати інформацію про те, що відбувається, у серії зображень. Відеоаналітика для безпеки зазвичай використовується для виявлення та отримання інформації про рух об'єктів, людей та транспортних засобів на записах відеоспостереження .

Системи відеоаналітики спостереження пропонують більш практичний та ефективний спосіб перегляду та спостереження за записами з камер спостереження. Контент, знятий кількома камерами протягом кількох днів, може

бути автоматично сортований за інтересами, допомагаючи співробітникам служби безпеки виявляти підозрілу активність та належним чином реагувати на неї в режимі реального часу та під час розслідувань.

Системи відеоаналітики обробляють відеопотоки за допомогою алгоритмів, розроблених для виявлення певних подразників. Захоплені зображення послідовно переглядаються спеціалізованими програмними інструментами, запрограмованими на перевірку певних подій або об'єктів, які можуть становити загрозу безпеці.

У простому сенсі, відеоаналітика шукає аномальні відмінності в послідовності зображень, а потім генерує дані про ці події за допомогою алгоритмів на основі правил. Наприклад, якщо відеокамера фіксує об'єкт, що рухається в її полі зору, відеоаналітика ставитиме запитання, щоб допомогти визначити об'єкт і вирішити, чи заслуговує його присутність на подальші дії.

Під поняттям відеоаналітики слід розуміти два основні типи систем:

– Традиційна відеоаналітика: Базові системи використовують алгоритми на основі правил для аналізу відеоконтенту. Якщо щось у серії зображень змінюється, програмне забезпечення задасть низку запитань типу « *якщо/тоді»*, щоб звузити коло можливої зміни. Однак традиційні системи аналітики не можуть зберігати інформацію або навчатися на основі раніше зареєстрованих інцидентів.

– Відеоаналітика на основі штучного інтелекту: Відеоаналітика на основі штучного інтелекту також використовує процес на основі правил для отримання інформації про зображення. Однак їхні алгоритми використовують інструменти штучного інтелекту та машинного навчання, щоб навчатися на основі ширших даних. Простіше кажучи, глибоке навчання у відеоаналітиці дозволяє системам вивчати закономірності з історичних подій для підвищення точності виявлення.

Поширені типи відеоаналітики в системах відеоспостереження

Відеоаналітика в режимі реального часу дозволяє службам безпеки виявляти закономірності, аномальні події та підозрілу активність, які в іншому

випадку могли б залишитися непоміченими. Камери відеоаналітики забезпечують постійне спостереження за ключовими зонами, а різні алгоритми відеоаналітики спеціально розроблені для пошуку певних подразників. Нижче наведено деякі поширені типи аналітики.

Автоматичне розпізнавання номерних знаків (ALPR)

Камери ALPR використовують спеціальний тип відеоаналітики, який називається оптичним розпізнаванням символів (OCR), для зчитування інформації про номерні знаки транспортних засобів, що проїжджають повз. Технологію ALPR можна використовувати для підтримки операцій управління паркуванням та контролю доступу транспортних засобів, а також для спостереження за під'їзними дорогами та паркувальними зонами, щоб виявити наявність підозрілих транспортних засобів.

Виявлення натовпу

Алгоритми відеоаналітики, що використовуються для виявлення натовпу, запрограмовані на ідентифікацію людей та вимірювання щільності натовпу в полі зору камери. Аналітика виявлення натовпу використовується для підвищення безпеки на живих заходах, попередження команд про потенційні вузькі місця та порушення, які можуть потребувати додаткової уваги, а також для відстеження рівня заповненості та виявлення незвичайної активності.

Розпізнавання обличчя

Розпізнавання облич використовується для ідентифікації присутності людських облич у відеоконтенті, а також для порівняння облич з тими, що зберігаються в базах даних. Цей тип відеоаналітики може контролювати доступ до безпечних місць та покращувати безпеку периметра, попереджаючи команди про присутність відомих правопорушників та сторонніх осіб, які перебувають навколо приватних володінь.

Підрахунок людей

Системи відстеження людей аналізують різні типи біометричних показників, щоб краще зрозуміти дії людей у цільових зонах. Завдяки

розпізнаванню облич, виявленню руху та поведінкових характеристик ці системи можуть ідентифікувати осіб та стежити за ними в приміщеннях, щоб покращити методи виявлення вторгнень та управління зайнятістю.

Відстеження об'єктів

Відеоаналітика для відстеження об'єктів контролює наявність та рух певних предметів у полі зору камери. Алгоритми можна налаштувати відповідно до різних випадків використання відеоаналітики. Наприклад, камери можуть бути запрограмовані для моніторингу посилок під час їхнього переміщення через пункти доставки та виконання замовлень або використовуватися для відстеження автомобілів для підтримки операцій з контролю дорожнього руху.

Виявлення руху

Алгоритми відеоаналітики, оптимізовані для виявлення руху, запрограмовані на безперервний пошук ознак руху в заданій області. Для цієї мети зазвичай використовується аналітика відеоспостереження з машинним навчанням. Системи можна навчити розуміти, як простори використовуються за нормальних умов, тому вони попереджають охоронців лише про рухи, які можуть становити проблему.

Виявлення предметів без нагляду

Системи відеоаналітики можна запрограмувати на моніторинг появи та руху статичних об'єктів у визначеному місці. Ці рішення добре впроваджуються в громадських місцях, таких як торгові центри, розважальні заклади та транспортні вузли, щоб допомогти співробітникам служби безпеки виявляти потенційні бомби та забезпечувати вільні аварійні виходи від перешкод.

Моніторинг зайнятості

Інструменти моніторингу заповнюваності підраховують кількість людей, які проходять через заздалегідь визначену зону протягом встановленого періоду часу. Цю технологію можна використовувати для забезпечення підтримки безпечного рівня заповнюваності та для збору даних про те, як використовуються приміщення. Наприклад, у роздрібній торгівлі її можна використовувати для

визначення того, коли послуги є найбільш популярними, та для вимірювання ефективності організаційних планів.

3.2 Розробка структурної схеми

Переваги технології відеоаналітики

Вибір розробки та впровадження індивідуальних рішень для відеоаналітики може надати бізнесу та фахівцям з безпеки кілька суттєвих переваг. Завдяки інтелектуальному програмному забезпеченню, яке використовується для ефективного управління та отримання аналітичних даних з величезних обсягів, команди людей можуть покращити виконання ключових завдань, автоматично виділяючи високоякісну та релевантну аналітику.

1. Підвищена ефективність

Типова система безпеки містить багато IP-камер та моніторів, розташованих для охоплення ключових зон. Навіть найменша зміна в сигналі безпеки може свідчити про загрозу, що розгортається, але командам може бути важко ефективно спостерігати за всіма камерами безперервно.

Системи відеоаналітики безпеки можна навчити автоматично виявляти аномальні події та попереджати операторів про них, надсилаючи інформацію персоналу диспетчерської відеоспостереження та персоналу на місці через SMS або електронну пошту. Це допомагає забезпечити доведення до відома відповідного персоналу всіх підозрілих дій, а високоякісні записи подій миттєво генеруються для покращення розслідувань.

2. Покращене прийняття рішень

Поряд з виявленням аномальних подій, які можуть вимагати подальшого розслідування, системи відеоаналітики можуть розпізнати тип інциденту, який може розгорнутися. Завдяки ключовим типам відеоаналітики, таким як інструменти відстеження об'єктів та розпізнавання облич, персонал може бути

попереджений про наявність вогнепальної зброї, сторонніх осіб або скупчення людей, що сприятиме обґрунтованому реагуванню на інциденти.

3. Зменшення кількості хибних тривог

Хоча традиційні системи безпеки можуть бути ефективними засобами стримування злочинів, оператори або центральні станції моніторингу повинні проводити додатковий аналіз, щоб зрозуміти причини активації. За оцінками, від 95% до 98% спрацьованих сигналів тривоги про крадіжку, паніку та пограбування є хибнопозитивними, що потенційно призводить до втрати часу та ресурсів, що може зробити організації вразливими до ширших ризиків.

Оскільки системи відеоаналітики безпеки розроблені для ідентифікації та розуміння конкретних подразників в унікальних умовах їхнього конкретного середовища, ризик хибних тривог можна суттєво зменшити. Це дозволяє персоналу швидко реагувати на ризики та скорочує час, витрачений на аналіз даних фізичної безпеки, щоб персонал міг зосередитися на важливих завданнях.

4. Потенційна економія коштів

Хоча початкові витрати на розробку систем відеоаналітики можуть бути високими, компанії, які впроваджують ці інструменти, можуть отримати довгострокову економію коштів. Перш за все, підвищена точність систем безпеки відеоаналітики може допомогти обмежити фінансовий вплив таких подій, як крадіжка та пошкодження майна, а додаткові переваги можна знайти в організаційних покращеннях.

Для аналізу, організації та реагування на дані безпеки знадобиться менше часу та ресурсів, що дозволить організаціям підвищити ефективність моніторингу відеоспостереження та процесів розслідування. Рішення для відеоаналітики також можуть бути впроваджені для спостереження за тим, як співробітники та гості взаємодіють з фізичними активами та інфраструктурою, допомагаючи підприємствам покращувати послуги та відповідати очікуванням клієнтів.

5. Постійні вдосконалення

Дані, зібрані та проаналізовані інструментами відеоаналітики, можна використовувати для постійного вдосконалення ширших аспектів діяльності бізнесу. Наприклад, відеоаналітика в поєднанні з камерами зі штучним інтелектом може допомогти фахівцям зрозуміти типи загроз, до яких їхні організації найбільш вразливі, впливаючи на майбутні рішення щодо вдосконалення систем безпеки та внутрішніх політик.

Подібні принципи можна застосовувати до інших застосувань відеоаналітики. Системи, розгорнуті для спостереження за виробничими завданнями, можуть надавати інформацію щодо покращення виробництва. Аналітика ALPR може допомогти командам покращити операції з управління паркуванням, а поведінкова аналітика може сприяти покращенню обслуговування клієнтів у роздрібній торгівлі та готельному бізнесі.

Міркування щодо впровадження систем відеоаналітики

Системи безпеки відеоаналітики та управління бізнесом можуть забезпечити вимірні переваги лише за умови адаптації до унікальних потреб організації. Розробляючи індивідуальні рішення для відеоаналітики, власники бізнесу повинні враховувати такі фактори:

– Обробка: Аналітичне програмне забезпечення може обробляти дані на центральному сервері або на найближчому пристрої. Старіші камери зазвичай потребують серверних систем, які часто страждають від затримки під час обробки аналітики. Для порівняння, новіші IP- камери можуть використовувати периферійну аналітику для обробки даних та отримання аналітики про сам пристрій у режимі реального часу.

– Точність виявлення: Відеоаналітика з використанням машинного навчання та функцій штучного інтелекту може бути використана для розробки високоточних систем. Програмне забезпечення можна навчити розуміти нормальні умови роботи цільових зон та попереджати персонал про аномальні події.

– Масштабованість: Потреби в безпеці можуть змінюватися з часом, тому вкрай важливо вибрати рішення, яке можна масштабувати за потреби. Подумайте, наскільки легко та економічно ефективно може бути додавання нового обладнання та програмних функцій відеоаналітики до запропонованих інсталяцій.

– Можливості інтеграції: Аналітичні дані, отримані за допомогою інструментів відеоаналітики, можна використовувати для оптимізації роботи ширших систем безпеки. Перевірте, чи підтримують потрібні рішення конфігурації відкритого API та чи сумісні вони з існуючими пристроями безпеки для бізнесу.

Майбутні тенденції у відеоаналітиці

Зростаючий інтерес до технологій на основі штучного інтелекту призвів до збільшення впровадження штучного інтелекту в різних бізнес-секторах, причому з 2022 року цілих 72 % організацій впроваджують штучний інтелект для виконання принаймні однієї бізнес-функції. Оскільки штучний інтелект та алгоритми глибокого навчання стають дедалі більш досконалими, все більше галузей готові отримати вигоду від інтелектуальних систем відеоаналітики.

Відеоаналітика на основі штучного інтелекту зараз є звичайним явищем у сучасних розумних містах та технологіях розумних будівель, а інструменти використовуються не лише для покращення безпеки, але й для підвищення енергоефективності, моніторингу навколишнього середовища та забезпечення належного обслуговування інфраструктури. Відеоаналітика, пов'язана з промисловим Інтернетом речей (IIoT), допомагає фахівцям виконувати ці завдання, дозволяючи автономно коригувати фізичну інфраструктуру у відповідь на високоякісні дані.

Інтерес до нових технологій штучного інтелекту, таких як відеоаналітика, поширюється також на малий бізнес та додатки, а дослідження, опубліковане у 2024 році, показало, що 98% малих підприємств зараз використовують інструменти на базі штучного інтелекту. Оскільки програмне забезпечення для

відеоаналітики стає більш доступним та зручним у використанні, його впровадження в усіх галузях промисловості та середовищах, ймовірно, продовжуватиме зростати.

Відеоаналітика в режимі реального часу надає численні суттєві переваги підприємствам у більшості основних секторів, дозволяючи фахівцям отримувати практичну інформацію про важливі процеси безпеки, інфраструктури та організації. Використовуючи унікальні системи відеоаналітики, команди можуть покращити реагування на безпеку, отримати інформацію про бізнес-операції та допомогти працівникам-людям виконувати завдання безпечно та ефективно, що може бути корисним для сучасних підприємств будь-якого розміру.

Мережевий відеореєстратор (NVR) – це спеціалізований комп'ютер, який записує відео з камер безпеки в цифровому форматі на жорсткий диск. Оскільки NVR не має можливості відеозахоплення, відео зазвичай обробляється та кодується з IP-камери спостереження або камери CCTV і передається на NVR для зберігання через мережу Ethernet або Wi-Fi. NVR зазвичай використовуються в системах IP-відеоспостереження.

Мережеві відеореєстратори замінили застарілі цифрові відеореєстратори (DVR). Переваги включають:

- Запис відео та аудіо.
- Краща якість зображення.
- Гнучкість системи.
- Краще покриття огляду.
- Дротове або бездротове.
- Потрібен 1 кабель для відео, аудіо та живлення.
- Розпізнавання облич, номерних знаків тощо завдяки кращій якості зображення.

Функції мережевого відеореєстратора (NVR)

Функції NVR можуть включати:

- Відеоаналітику.

– Параметри режиму запису.

– Мережеві комутатори з живленням через Ethernet (PoE).

– Керування PTZ-камерою.

– Дистанційне налаштування.

– Тригери запису.

– Схеми стиснення відео.

Порівняння цифрових відеореєстраторів та мережевих відеореєстраторів

Цифровий відеореєстратор (DVR) записує відеозаписи з камер спостереження на локальні пристрої зберігання даних, найчастіше на жорсткий диск. DVR може записувати відео з аналогових джерел на місці або захоплювати відео з цифрового джерела. DVR можна підключити до аналогових камер за допомогою коаксіальних кабелів, що дозволяє отримати до них віддалений доступ. DVR пропонують розширені функції, такі як можливість пошуку записів за подіями або сортування за часом і датою. DVR можна налаштувати на автоматичну заміну старих записів після заповнення сховища.

Кожній камері безпеки потрібен центральний відеореєстратор для передачі та архівування записаного матеріалу. Відеомагнітофони еволюціонували в моделі DVR, які потім були замінені технологією NVR, що дозволяє контролювати необмежену кількість камер, як в одному місці, так і по всьому світу.

Результати порівняння між відеореєстраторами та мережевими відеореєстраторами:

– Роздільна здатність записів. Відеореєстратори можуть записувати лише з роздільною здатністю 720p. З іншого боку, відеореєстратори пропонують можливості запису високої чіткості 1080p та неймовірну чіткість зображення. D1 – це стандартна якість відео, що використовується системами відеоспостереження замкнутого контуру, тоді як HD пропонує набагато чіткіше зображення з роздільною здатністю 1920×1080 пікселів деталізації.

– Підключення камери. Підключення аналогових систем відеоспостереження є досить складним, оскільки вони вимагають BNC-роз'ємів для кожної камери, підключеної до одного пристрою, що означає прокладання великої кількості проводів між пристроями (або використання PoE). Це також обмежує кількість камер, які ви можете мати одночасно, перш ніж знадобиться ще один пристрій.

Відеореєстратори (NVR) – це сучасний погляд на цифровий відеореєстратор (DVR), який вирішує багато його проблем. NVR усуває ці проблеми, оскільки він підключений безпосередньо до однієї мережі, тоді як IP-камери передають свої кадри через інше з'єднання. Це робить масштабування NVR набагато доступнішим, ніж у випадку з системою DVR, яка вимагає складніших оновлень обладнання та процесів встановлення, щоб нові системи належним чином працювали разом.

Відеонаглядові відеореєстратори (NVR) – це найпоширеніший спосіб зберігання та доступу до відео з IP-камер. Моделі з підтримкою Wi-Fi можуть передавати відео бездротовим способом без обмежень щодо близькості, якщо вони знаходяться в одній мережі. Однак, перед покупкою обладнання важливо знати про проблеми сумісності.

Гібридні відеореєстратори

Гібридні відеореєстратори (HVR) – це новітня технологія відеоспостереження. Вони можуть працювати як з аналоговими камерами, так і з IP-камерами, що дає їм перевагу над традиційними системами. Ця універсальність заощадить ваші кошти, коли справа доходить до купівлі нового обладнання.

Результати порівняння між відеореєстраторами та мережевими відеореєстраторами:

Підключення камери:

– Відеонаглядові відеореєстратори (NVR) – це найпоширеніший спосіб зберігання та доступу до відеозаписів з IP-камер.

– NVR набагато масштабованіші, ніж DVR.

Роздільна здатність записів:

– Відеореєстратори можуть записувати лише з роздільною здатністю 720p. З іншого боку, відеореєстратори пропонують можливості запису високої чіткості 1080p та неймовірну чіткість зображення.

Гібридні відеореєстратори

– Може працювати як з аналоговими камерами, так і з IP-камерами.

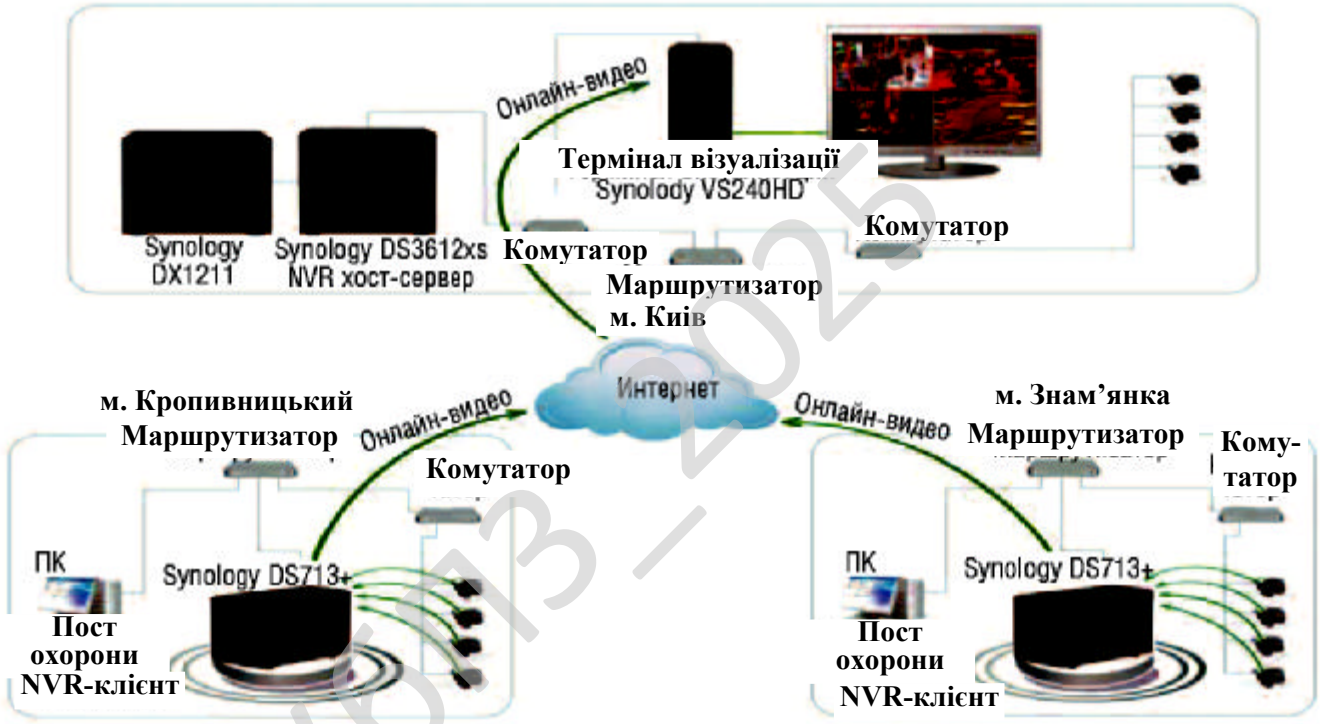

Рисунок 3.1 – Структурна схема системи

3.3 Розробка функціональної схеми

На рисунку 3.2 зображена функціональна схема системи. Нижче розглянемо її більш докладно.

Цифрові системи відеоспостереження виконують наступні функції:

– 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.

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

– 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 настроювання аудіореєстрації разом з відеозображенням, архів від цих камер записується спільно зі звуком. Аудіореєстрація необхідна у випадках недостачі тієї інформації, що надають відеокамери. Більше твердий контроль за підлеглими

й персоналом. Найбільш часті сфери й місця застосування аудіоконтроля – 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.

– 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 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 сигналізацій), і управляти зовнішніми виконавчими пристроями (сиреною, голосовим оповіщувачем або іншим пристроєм).

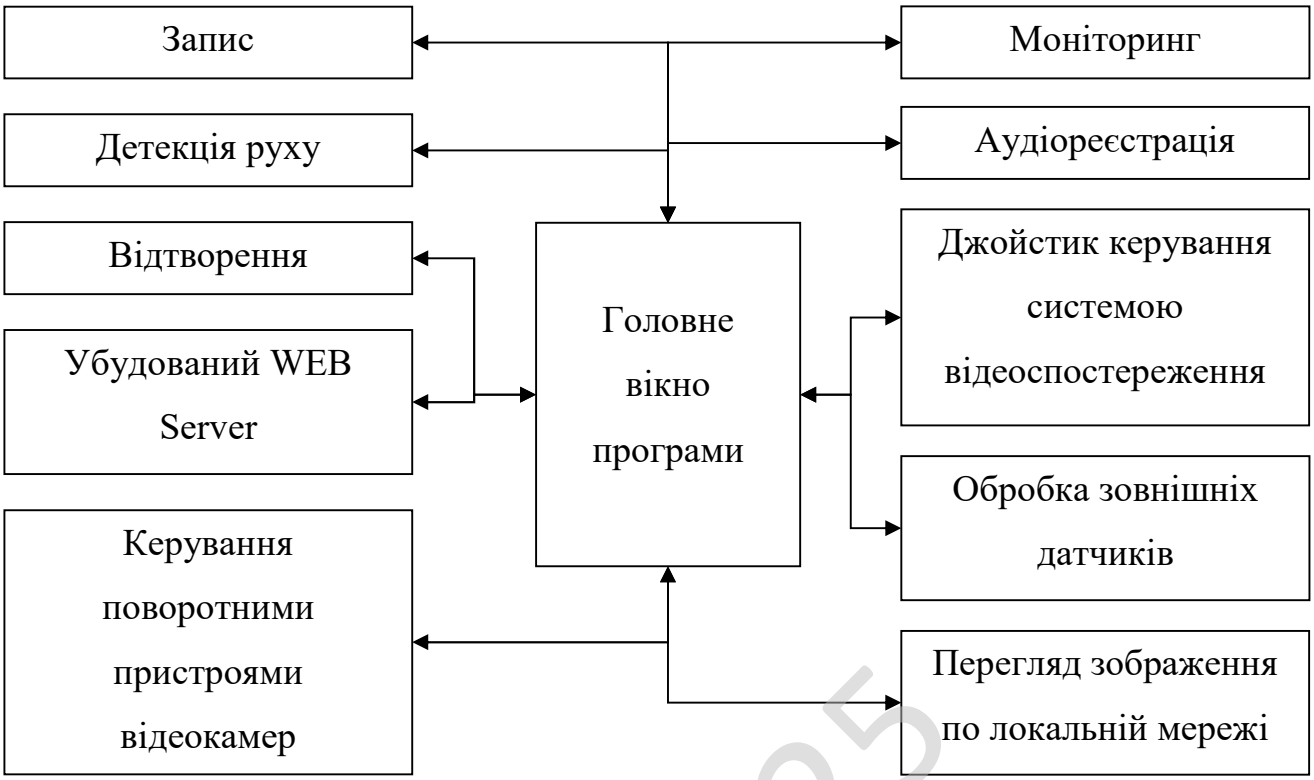

Рисунок 3.2 – Функціональна схема системи

– 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 настроювань операційної системи, настроювань програми відеоспостереження, установки ігор і сторонніх додатків. Рішенням даної проблеми є заміна стандартних пристроїв уведення (клавіатури й миші) на спеціалізований маніпулятор. Даний маніпулятор дозволяє працювати тільки із програмним забезпеченням.

Цифрова система відеоспостереження, що розроблена:

– забезпечує високу якість відтвореного відеозапису;

– високу швидкість доступу до відеоархіву;

– можливість цифрового збільшення й масштабування будь-якого кадру;

– 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;

– можливість експорту відеоінформації на сумісні зовнішні носії.

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

3.4 Розробка діаграми процесів

Діаграма процесів розробленої системи зображена на рисунку 3.3. При детальному її розгляді можна побачити як саме проходе взаємодія у розробленій системі. Використовується модель проектування, графічне представлення «потоків» даних в інформаційній системі.

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.

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.

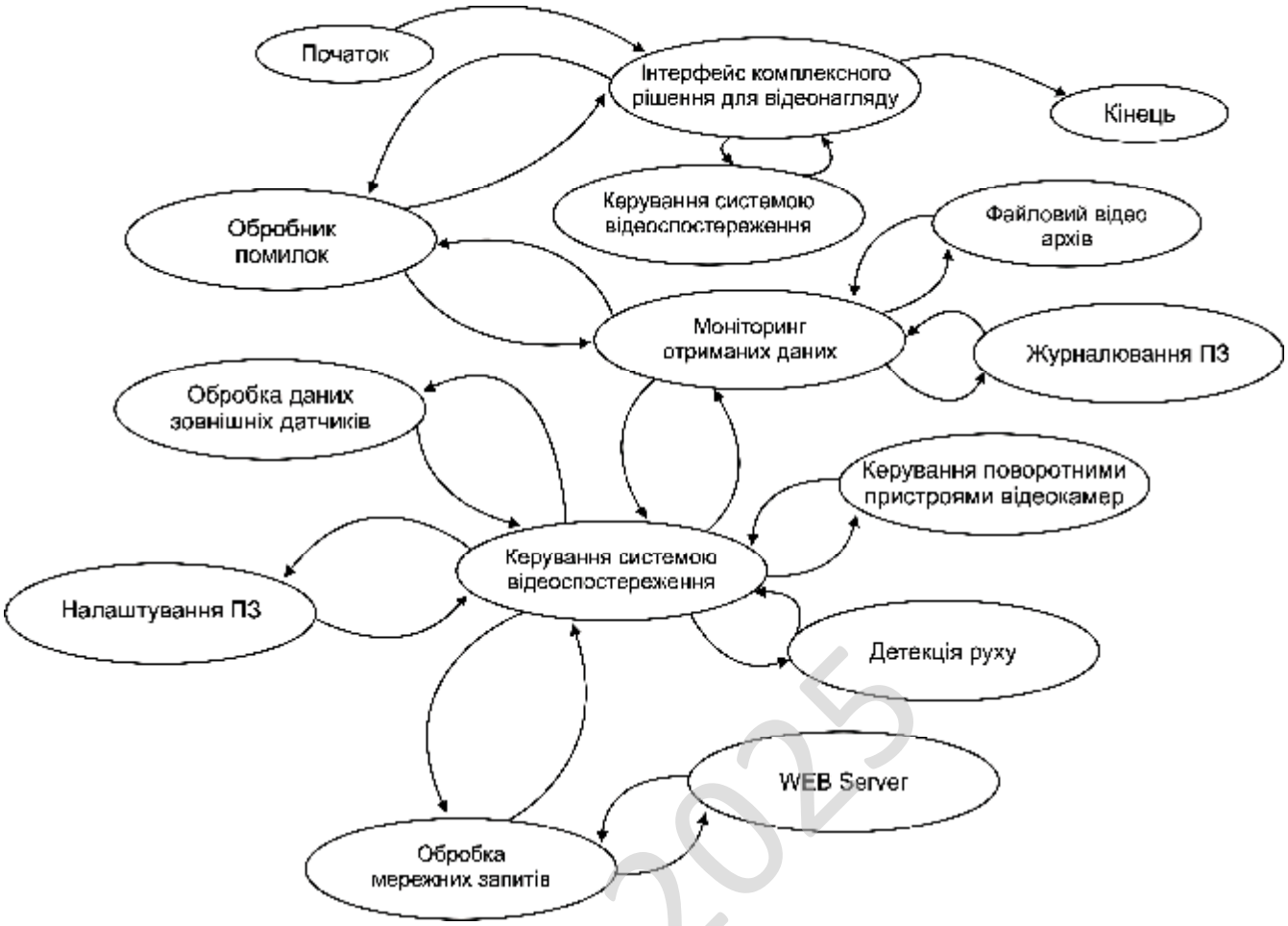

Рисунок 3.3 – Діаграма взаємодії процесів

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 РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ПРАВИЛЬНІСТЬ ПРОЕКТНИХ РІШЕНЬ

4.1 Блок-схеми та опис алгоритмів функціонування системи

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 – перевірка поточного стану з завершенням роботи розробленого ПЗ. При роботі підпрограми виконується основний функціонал системи з циклічними послідовностями, перевіркою поточного стану та поверненням в основну програму прапорів стану виконання.

Було використано підходи з використанням UML, це уніфікована мова моделювання, використовується у парадигмі об'єктно-орієнтованого програмування. Є невід'ємною частиною уніфікованого процесу розробки програмного забезпечення. UML є мовою широкого профілю, це відкритий стандарт, що використовує графічні позначення для створення абстрактної моделі системи, називаної UML-моделлю. UML був створений для визначення, візуалізації, проектування й документування в основному програмних систем. UML не є мовою програмування, але в засобах виконання UML-моделей як інтерпретованого коду можлива кодогенерація.

UML може бути застосовано на всіх етапах життєвого циклу аналізу бізнес-систем і розробки прикладних програм.

Різні види діаграм які підтримуються UML, і найбагатший набір можливостей представлення певних аспектів системи робить UML універсальним засобом опису як програмних, так і ділових систем.

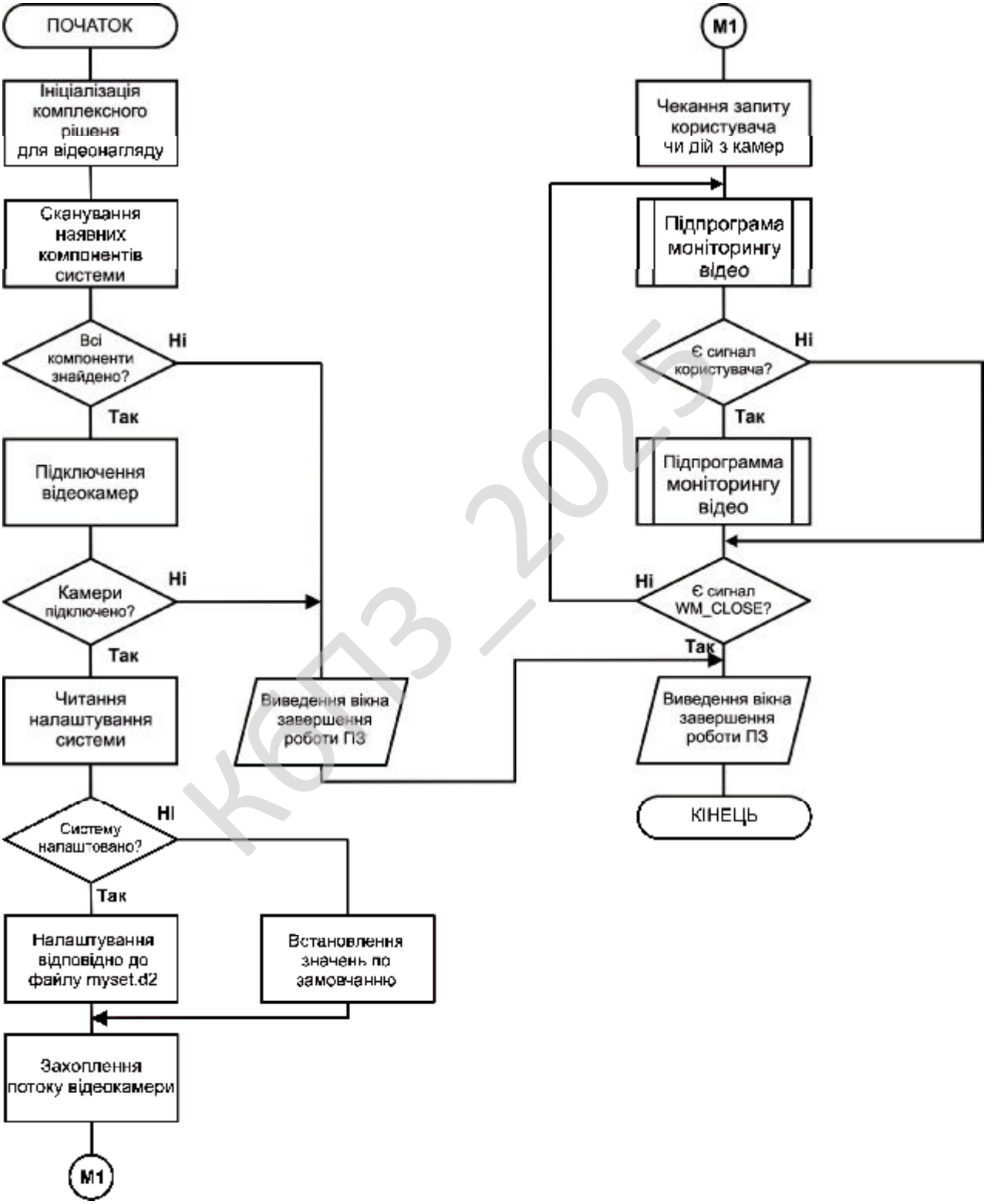

Рисунок 4.1 – Блок-схема основної програми

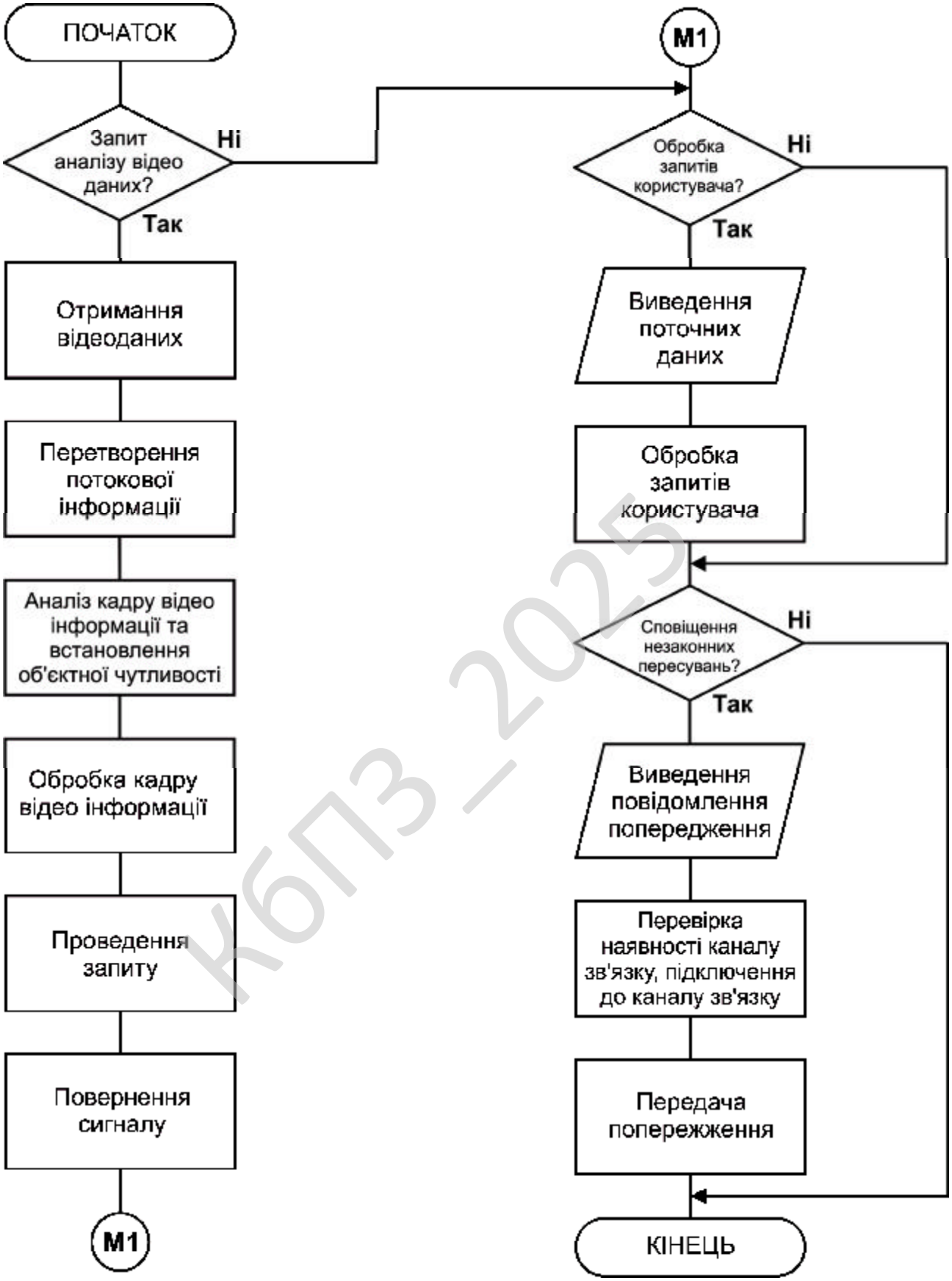

Рисунок 4.2 – Блок-схема роботи підпрограми

Діаграми дають можливість представити систему (як ділову, так і програмну) у такому вигляді, щоб її можна було легко перевести в програмний код. Основною причиною використання мови UML є спілкування розробників між собою.

Крім того, UML спеціально створювалася для оптимізації процесу розробки програмних систем, що дозволяє збільшити ефективність їх реалізації у кілька разів і помітно поліпшити якість кінцевого продукту.

UML прекрасно зарекомендувала себе в багатьох успішних програмних проектах. Засоби автоматичної генерації кодів дозволяють перетворювати моделі мовою UML у вихідний код об'єктно-орієнтованих мов програмування, що ще більш прискорює процес розробки. Практично усі CASE-засоби (програми автоматизації процесу аналізу і проектування) мають підтримку UML. Моделі розроблені в UML, дозволяють значно спростити процес кодування і направити зусилля програмістів безпосередньо на реалізацію системи.

Діаграми підвищують супроводжуваність проекту і полегшують розробку документації.

UML необхідний:

– Керівникам проектів, які керують розподілом завдань і контролем за проектом.

– Проектувальникам інформаційних систем які розробляють технічні завдання для програмістів.

– Бізнес-аналітикам, які досліджують реальну систему і здійснюють інжиніринг і реінжиніринг бізнесу компанії.

– Програмістам які реалізовують модулі інформаційної системи.

При модифікації системи об'єктний підхід дозволяє легко включати в систему нові об'єкти і виключати застарілі без істотної зміни її життєздатності. Використання побудованої моделі при модифікаціях системи дає можливість усунути небажані наслідки змін, оскільки вони не ламають структури системи, а тільки змінюють поведінку об'єктів.

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 програми високого рівня, також те, що при розробці програми слід надати особливу увагу модулю комплексних рішень для відеонагляду.

Під час роботи над магістерською роботою було створено блок-схеми. Перед їх розглядом необхідно провести роз’яснення який саме тип блок-схем використовується.

Блок-схема це представлення задачі для її аналізу або розв'язування за допомогою спеціальних символів (геометричних образів), які позначають такі елементи, як операції, потік, дані тощо. Блок вхідних та вихідних даних прийнято позначати паралелограмом, блок обчислень (обробки) даних – прямокутником, блок прийняття рішень – ромбом, еліпсом – початок та кінець алгоритму.

У інформаційних технологіях функціональна схема складається з функціональних блоків, які являють собою конструктивно відособлені частини (елементи або пристрої) автоматичних систем, які виконують певні функції. 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 схеми можуть виконуватися в укрупненому і розгорненому вигляді. У першому випадку на схемі зображають найважливіші блоки системи і зв'язки між ними.

У другому варіанті схема відображається більш детально, що полегшує її читання та ілюструє принцип роботи.

Основні елементи схем алгоритму це термінатор, процес, рішення, зумовлений процес (підпрограма), дані та з'єднувач.

Термінатор це елемент відображає вхід із зовнішнього середовища або вихід з неї (найчастіше застосування – початок і кінець програми). Всередині фігури записується відповідна дія.

Процес це виконання однієї або кількох операцій, обробка даних будь-якого виду (зміна значення даних, форми подання, розташування). Всередині фігури записують безпосередньо самі операції.

Рішення це показує рішення або функцію перемикального типу з одним входом і двома або більше альтернативними виходами, з яких тільки один може бути обраний після обчислення умов, визначених всередині цього елементу. Вхід в елемент позначається лінією, що входить зазвичай у верхню вершину елементу. Якщо виходів два чи три то зазвичай кожен вихід позначається лінією, що виходить з решти вершин (бічних і нижній). Якщо виходів більше трьох, то їх слід показувати однією лінією, що виходить з вершини (частіше нижній) елемента, яка потім розгалужується. Відповідні результати обчислень можуть записуватися поруч з лініями, що відображають ці шляхи.

Зумовлений процес (підпрограма) це символ відображає виконання процесу, що складається з однієї або кількох операцій, що визначені в іншому місці програми (у підпрограмі, модулі). Всередині символу записується назва процесу і передані в нього дані.

Дані це перетворення у форму, придатну для обробки (введення) або відображення результатів обробки (виведення). Цей символ не визначає носія даних (для вказівки типу носія даних використовуються специфічні символи).

З'єднувач це символ відображає вихід в частину схеми і вхід з іншої частини цієї схеми. Використовується для обриву лінії та продовження її в іншому місці (приклад: поділ блок-схеми, що не поміщається на листі). Відповідні сполучні символи повинні мати одне (при тому унікальне) позначення.

Архітектура системи та класи реалізації

Система має кілька основних підсистем. Підсистема конфігурації камер описує джерела відео та їх параметри. Підсистема генерації та попередньої обробки кадрів формує вхідні дані для аналізу.

Інтелектуальна підсистема виявлення руху та відстеження обєктів відповідає за перетворення сирих кадрів на структуровану інформацію. Підсистема подій реєструє усі значущі ситуації. Аналітичний модуль узагальнює події у вигляді звітів. Окремий модуль експериментів дозволяє досліджувати вплив параметрів на якість роботи системи. Над усіма цими компонентами працює головний клас оркестратор який координує обмін даними.

Базові структури даних

Усі ключові сутності системи описуються датакласами пайтон. Клас CameraConfig зберігає ідентифікатор камери назву зручну для оператора мережеве джерело сигналу текстовий опис зони спостереження прапорець активності параметр чутливості та максимально допустиму кадрову частоту. Таким чином система зберігає всю конфігураційну інформацію в одному місці та легко змінює параметри камер без зміни логіки аналізу.

Клас Frame подає окремий кадр як матрицю цілих чисел яскравості. Для дослідження не обовязково працювати з реальними відеофайлами. Тому кадр описується двовимірним списком значень від нуля до двохсот п’ятдесяти п’яти. Це дозволяє моделювати різні сцени та пороги спрацювання алгоритмів.

Клас DetectedObject описує результат роботи алгоритму виявлення. У ньому зберігається ідентифікатор обєкта межі прямокутника навколо обєкта рівень впевненості алгоритму текстова мітка наприклад рух та мітка часу. Така структура дає змогу в подальшому доповнювати систему іншими типами обєктів.

Клас Event описує події системи. Для кожної події зберігається ідентифікатор тип події наприклад intrusion або motion опис рівень важливості час виникнення та перелік обєктів які повязані з цією подією. Надалі Event слугує основою журналу подій та аналітичних звітів.

Клас SystemMetrics накопичує кількість оброблених кадрів кількість виявлених подій кількість хибних спрацювань а також час запуску системи. Метод uptime_seconds обчислює тривалість роботи що дозволяє оцінювати продуктивність та порівнювати різні конфігурації.

Модуль генерації кадрів та попередньої обробки

Клас FrameGenerator емітує роботу фізичної камери. Він використовує конфігурацію CameraConfig та генерує випадкову матрицю яскравості. Метод generate_frame створює новий кадр з урахуванням ідентифікатора камери та поточного часу. Така реалізація дозволяє досліджувати алгоритми аналізу без підключення реального обладнання та відтворювати різні сценарії.

Клас FramePreprocessor реалізує попередню обробку кадрів. Метод normalize контролює діапазон значень яскравості та запобігає виходу значень за межі. Це відповідає кроку нормалізації зображення перед роботою більш складних моделей. Метод reduce_noise виконує згладжування через усереднення значень у невеликому околі кожного пікселя. У реальній системі цей модуль замінюється на фільтри з бібліотек компьютерного зору але базова структура лишається незмінною.

Модуль виявлення руху та відстеження обєктів

Клас SimpleMotionDetector моделює інтелектуальний модуль виявлення. Для кожної камери система зберігає попереднє середнє значення яскравості кадру. Метод detect обчислює середню яскравість поточного кадру порівнює її з попереднім значенням та аналізує різницю. Параметр sensitivity задає чутливість. Якщо зміна перевищує поріг метод створює обєкти DetectedObject з імітованими координатами прямокутників. Рівень впевненості залежить від величини зміни середньої яскравості.

Клас ObjectTracker відповідає за підтримку списку активних обєктів. Для кожної камери зберігається словник який повязує ідентифікатори обєктів із останніми спостереженнями. Метод update приймає черговий список виявлених обєктів оновлює структуру даних та повертає актуальний перелік. Метод

get_active_objects дозволяє отримати всі обєкти які система вважає активними для обраної камери. У майбутньому на цьому місці легко інтегруються більш складні алгоритми відстеження.

Формування подій та зберігання інформації

Клас EventFactory перетворює списки обєктів на події. Метод create_event аналізує список DetectedObject. Якщо список порожній або рівень впевненості занадто малий подія не створюється. Якщо площа хоч одного прямокутника перевищує поріг фабрика формує подію типу intrusion з високим рівнем важливості. В інших випадках створюється подія типу motion із середнім рівнем важливості. Для кожної події формується опис у якому зазначається кількість виявлених обєктів.

Клас EventStorage зберігає події у памяті та забезпечує серіалізацію у формат JSON. Метод add_event додає нову подію до списку. Метод to_json перетворює всі події на структуру даних придатну до запису у файл. Метод save_to_file записує дані у файл з кодуванням юнікод. Завдяки цьому журнал подій легко інтегрується з зовнішніми системами моніторингу та звітності.

Аналітичний модуль

Клас AnalyticsEngine працює поверх EventStorage. Метод events_by_camera рахує кількість подій для кожної камери. Метод events_by_type рахує кількість подій кожного типу.

Метод events_by_severity формує розподіл за рівнями важливості. На основі цих словників у пояснювальній записці будуються таблиці та графіки які показують ефективність системи та навантаження на кожну камеру.

Модуль досліджень та експериментів

Для обґрунтування вибору параметрів чутливості у системі існує окремий модуль експериментів. Клас ExperimentScenario описує сценарій дослідження. Він містить назву список значень чутливості та кількість кадрів для кожної серії.

Датаклас ExperimentResult зберігає результат однієї серії. Для кожного значення чутливості фіксується кількість подій та кількість оброблених кадрів. Ці дані дозволяють оцінювати частоту спрацювань та вибирати оптимальні пороги.

Клас ExperimentRunner організує виконання експериментів. Метод run створює генератор кадрів та модуль попередньої обробки. Для кожного значення чутливості створюється новий детектор сховище подій та фабрика подій. Далі система в циклі генерує кадри нормалізує їх зменшує шум виявляє рух формує обєкти та за потреби події. Після завершення серії формується ExperimentResult та додається до списку результатів.

Головний клас системи

Клас SurveillanceSystem виконує роль оркестратора. Він зберігає конфігурації камер генератори кадрів екземпляри попереднього обробника детектора трекера фабрики подій сховище подій та структуру метрик.

Метод add_camera додає нову камеру до системи та створює відповідний генератор кадрів. Метод process_single_frame отримує конфігурацію камери перевіряє її активність генерує кадр виконує нормалізацію та зменшення шуму викликає детектор руху оновлює трекер формує подію та за потреби записує її у сховище. Одночасно оновлюються лічильники у SystemMetrics.

Метод process_all_cameras у циклі викликає process_single_frame для кожної камери задану кількість разів. Таким чином система моделює безперервну роботу в умовах реального відеонагляду.

Метод build_analytics_report використовує AnalyticsEngine та формує зведений звіт у вигляді словника з трьома вкладеними структурами. Метод export_report записує цей звіт у файл формату JSON що спрощує подальшу обробку.

Демонстраційний сценарій

Функція demo_run показує типове використання системи у магістерській роботі. Вона створює обєкт SurveillanceSystem додає дві камери для входу до будівлі та для зони паркінгу задає для них різні значення чутливості та запускає

цикл обробки кадрів. Після цього функція формує аналітичний звіт виводить його на екран та зберігає журнал подій і звіт у файлах events.json та report.json. Такі файли можна включати до пояснювальної записки як приклад результатів роботи системи.

У підсумку програмна реалізація показує повний цикл функціонування системи інтелектуального відеонагляду. Камери подають кадри які проходять попередню обробку. Модуль виявлення формує обєкти. Трекер підтримує інформацію про обєкти у часі. Фабрика подій створює структуровані записи. Сховище та аналітичний модуль готують дані для інтерпретації. Модуль експериментів дозволяє дослідити вплив параметрів. Оркестратор поєднує усе це у єдину систему придатну для використання у магістерській випускній кваліфікаційній роботі.

```
import time
import random
import json
from dataclasses import dataclass, field
from typing import List, Dict, Tuple, Optional

#Конфігурація однієї камери
@dataclass
class CameraConfig:
    camera_id: str
    name: str
    source: str
    region: str
    is_active: bool = True
    sensitivity: float = 0.5
    max_frame_rate: int = 15

#Кадр з камери у вигляді спрощеної матриці яскравості
@dataclass
class Frame:
    camera_id: str
    timestamp: float
    data: List[List[int]]

#Опис виявленого обєкта
```

```
@dataclass
class DetectedObject:
    camera_id: str
    object_id: int
    bbox: Tuple[int, int, int, int]
    confidence: float
    label: str
    timestamp: float

#Подія у системі
@dataclass
class Event:
    event_id: int
    camera_id: str
    type: str
    description: str
    severity: str
    timestamp: float
    objects: List[DetectedObject] = field(default_factory=list)

#Метрики роботи системи
@dataclass
class SystemMetrics:
    processed_frames: int = 0
    detected_events: int = 0
    false_alarms: int = 0
    start_time: float = field(default_factory=time.time)

    def uptime_seconds(self) -> float:
#Обчислення часу роботи системи
        return time.time() - self.start_time

class FrameGenerator:
#Емуляція отримання кадрів від камери
    def __init__(self, config: CameraConfig):
        self.config = config

    def generate_frame(self) -> Frame:
#Формування випадкового кадру для імітації сцени
        width = 32
        height = 24
        data: List[List[int]] = []
```

```
        for _ in range(height):
            row: List[int] = []
            for _ in range(width):
                value = random.randint(0, 255)
                row.append(value)
            data.append(row)
        frame = Frame(
            camera_id=self.config.camera_id,
            timestamp=time.time(),
            data=data,
        )
        return frame

class FramePreprocessor:
#Попередня обробка кадрів
    def normalize(self, frame: Frame) -> Frame:
#Нормалізація яскравості пікселів
        new_data: List[List[int]] = []
        for row in frame.data:
            new_row: List[int] = []
            for value in row:
                normalized = int(max(0, min(255, value)))
                new_row.append(normalized)
            new_data.append(new_row)
        result = Frame(
            camera_id=frame.camera_id,
            timestamp=frame.timestamp,
            data=new_data,
        )
        return result

    def reduce_noise(self, frame: Frame) -> Frame:
#Проста фільтрація шуму через усереднення
        height = len(frame.data)
        width = len(frame.data[0]) if height > 0 else 0
        new_data: List[List[int]] = []
        for y in range(height):
            new_row: List[int] = []
            for x in range(width):
                neighbors: List[int] = []
                for dy in (-1, 0, 1):
                    for dx in (-1, 0, 1):
```

```
                        ny = y + dy
                        nx = x + dx
                        if 0 <= ny < height and 0 <= nx < width:
                            neighbors.append(frame.data[ny][nx])
                if neighbors:
                    average = sum(neighbors) / len(neighbors)
                else:
                    average = frame.data[y][x]
                new_row.append(int(average))
            new_data.append(new_row)
        result = Frame(
            camera_id=frame.camera_id,
            timestamp=frame.timestamp,
            data=new_data,
        )
        return result

class SimpleMotionDetector:
#Виявлення руху на основі зміни середньої яскравості
    def __init__(self, sensitivity: float = 0.5):
        self.sensitivity = sensitivity
        self.previous_average: Dict[str, float] = {}

    def detect(self, frame: Frame) -> List[DetectedObject]:
#Формування списку обєктів з імітацією руху
        flat = [value for row in frame.data for value in row]
        if not flat:
            return []
        current_average = sum(flat) / len(flat)
        previous = self.previous_average.get(frame.camera_id, current_average)
        delta = abs(current_average - previous)
        self.previous_average[frame.camera_id] = current_average
        objects: List[DetectedObject] = []
        if delta > self.sensitivity * 10:
            object_id = random.randint(1, 1000000)
            x = random.randint(0, 20)
            y = random.randint(0, 10)
            w = random.randint(5, 10)
            h = random.randint(5, 10)
            bbox = (x, y, w, h)
            confidence = min(1.0, delta / 50.0)
            label = "motion"
```

```
            detected = DetectedObject(
                camera_id=frame.camera_id,
                object_id=object_id,
                bbox=bbox,
                confidence=confidence,
                label=label,
                timestamp=frame.timestamp,
            )
            objects.append(detected)
        return objects

class ObjectTracker:
#Проста ідентифікація обєктів за камерами
    def __init__(self):
        self.active_objects: Dict[str, Dict[int, DetectedObject]] = {}

    def update(self, objects: List[DetectedObject]) -> List[DetectedObject]:
#Оновлення стану відстежуваних обєктів
        result: List[DetectedObject] = []
        for obj in objects:
            camera_objects = self.active_objects.setdefault(obj.camera_id, {})
            camera_objects[obj.object_id] = obj
            result.append(obj)
        return result

    def get_active_objects(self, camera_id: str) -> List[DetectedObject]:
#Отримання активних обєктів для вказаної камери
        camera_objects = self.active_objects.get(camera_id, {})
        return list(camera_objects.values())

class EventFactory:
#Формування подій на основі виявлених обєктів
    def __init__(self):
        self._next_event_id = 1

    def create_event(self, objects: List[DetectedObject]) -> Optional[Event]:
#Створення події вторгнення за наявності обєктів
        if not objects:
            return None
        example = objects[0]
        if example.confidence < 0.2:
            return None
```

```
        if any(obj.bbox[2] * obj.bbox[3] > 40 for obj in objects):
            severity = "high"
            event_type = "intrusion"
        else:
            severity = "medium"
            event_type = "motion"
        description = f"Виявлено {len(objects)} обєктів"
        event = Event(
            event_id=self._next_event_id,
            camera_id=example.camera_id,
            type=event_type,
            description=description,
            severity=severity,
            timestamp=example.timestamp,
            objects=list(objects),
        )
        self._next_event_id += 1
        return event

class EventStorage:
#Зберігання подій у памяті та у файлі
    def __init__(self):
        self.events: List[Event] = []

    def add_event(self, event: Event) -> None:
#Додавання нової події до списку
        self.events.append(event)

    def to_json(self) -> str:
#Серіалізація подій до формату JSON
        serializable = []
        for event in self.events:
            event_dict = {
                "event_id": event.event_id,
                "camera_id": event.camera_id,
                "type": event.type,
                "description": event.description,
                "severity": event.severity,
                "timestamp": event.timestamp,
                "objects": [],
            }
            for obj in event.objects:
```

```
                obj_dict = {
                    "camera_id": obj.camera_id,
                    "object_id": obj.object_id,
                    "bbox": obj.bbox,
                    "confidence": obj.confidence,
                    "label": obj.label,
                    "timestamp": obj.timestamp,
                }
                event_dict["objects"].append(obj_dict)
            serializable.append(event_dict)
        return json.dumps(serializable, ensure_ascii=False, indent=2)

    def save_to_file(self, path: str) -> None:
#Збереження серіалізованих подій у файл
        text = self.to_json()
        with open(path, "w", encoding="utf-8") as f:
            f.write(text)

class AnalyticsEngine:
#Аналітика за подіями
    def __init__(self, storage: EventStorage):
        self.storage = storage

    def events_by_camera(self) -> Dict[str, int]:
#Підрахунок кількості подій для кожної камери
        counts: Dict[str, int] = {}
        for event in self.storage.events:
            counts[event.camera_id] = counts.get(event.camera_id, 0) + 1
        return counts

    def events_by_type(self) -> Dict[str, int]:
#Підрахунок подій за типами
        counts: Dict[str, int] = {}
        for event in self.storage.events:
            counts[event.type] = counts.get(event.type, 0) + 1
        return counts

    def events_by_severity(self) -> Dict[str, int]:
#Підрахунок подій за рівнями важливості
        counts: Dict[str, int] = {}
        for event in self.storage.events:
            counts[event.severity] = counts.get(event.severity, 0) + 1
```

```
        return counts

class ExperimentScenario:
#Сценарій дослідження якості виявлення руху
    def __init__(self, name: str, sensitivity_values: List[float], frames_count:
int):
        self.name = name
        self.sensitivity_values = sensitivity_values
        self.frames_count = frames_count
```

При розгляді розробленого ПЗ можно побачити що програма розбита на декілька важливих блоків, таких як:

– Блок ініціалізації динамічних бібліотек користувача.

– Блок підключення додаткових модулів.

– Блок читання файлів налаштування та керування.

– Блок захоплення потоку.

– Блок очікування дій користувача.

– Блок аналізу даних.

– Блок обробки сигналу.

Обробка відео потоку і виведення на екран в середовищі Windows при застосуванні основних методів виведення відео інформації на екран лінійки операційних систем Windows. Виникає гостра проблема в швидкості обробки потокового кадру, що приводить до уповільнення процесу висновку інформації на екран. Як відомо для перегляду відеопотоку необхідно не менше 24 кадрів в секунду.

Також при розробці магістерської роботи було використано наступні підходи UML: діаграма діяльності (діаграми поведінки типу); діаграма прецедентів (діаграми поведінки типу); Діаграма класів; Діаграма компонент; Діаграма об'єктів; Діаграма розгортання.

Діаграма діяльності. Це візуальне представлення графу діяльностей. Граф діяльностей є різновидом графу станів скінченного автомату, вершинами якого є певні дії, а переходи відбуваються по завершенню дій. Дія є фундаментальною

одиницею визначення поведінки в специфікації. Дія отримує множину вхідних сигналів, та перетворює їх на множину вихідних сигналів.

Одна із цих множин, або обидві водночас, можуть бути порожніми. Виконання дії відповідає виконанню окремої дії. Подібно до цього, виконання діяльності є виконанням окремої діяльності, буквально, включно із виконанням тих дій, що містяться в діяльності. Кожна дія в діяльності може виконуватись один, два, або більше разів під час одного виконання діяльності. Щонайменше, дії мають отримувати дані, перетворювати їх та тестувати, деякі дії можуть вимагати певної послідовності.

Специфікація діяльності (на вищих рівнях сумісності) може дозволяти виконання декількох (логічних) потоків, та існування механізмів синхронізації для гарантування виконання дій у правильному порядку.

Діаграма прецедентів це діаграма, на якій зображено відношення між акторами та прецедентами в системі. Також, перекладається як діаграма варіантів використання.

Діаграма прецедентів є графом, що складається з множини акторів, прецедентів (варіантів використання) обмежених границею системи (прямокутник), асоціацій між акторами та прецедентами, відношень серед прецедентів, та відношень узагальнення між акторами. Діаграми прецедентів відображають елементи моделі варіантів використання.

Суть даної діаграми полягає в наступному: проектована система представляється у вигляді безлічі сутностей чи акторів, що взаємодіють із системою за допомогою так званих варіантів використання. Варіант використання (use case) використовують для описання послуг, які система надає актору. Іншими словами, кожен варіант використання визначає деякий набір дій, який виконує система при діалозі з актором.

При цьому нічого не говориться про те, яким чином буде реалізована взаємодія акторів із системою.

У мові UML є кілька стандартних видів відношень між акторами і варіантами використання:

– асоціації (association relationship);

– включення (include relationship);

– розширення (extend relationship);

– узагальнення (generalization relationship).

При цьому загальні властивості варіантів використання можуть бути представлені трьома різними способами, а саме – за допомогою відношень включення, розширення і узагальнення.

Відношення асоціації – одне з фундаментальних понять у мові UML і в тій чи іншій мірі використовується при побудові всіх графічних моделей систем у формі канонічних діаграм.

Включення (include) у мові UML – це різновид відношення залежності між базовим варіантом використання і його спеціальним випадком. При цьому відношенням залежності (dependency) є таке відношення між двома елементами моделі, при якому зміна одного елемента (незалежного) приводить до зміни іншого елемента (залежного).

Відношення розширення (extend) визначає взаємозв'язок базового варіанта використання з іншим варіантом використання, функціональна поведінка якого задіюється базовим не завжди, а тільки при виконанні додаткових умов.

Діаграма класів це статичне представлення структури моделі. Відображає статичні (декларативні) елементи, такі як: класи, типи даних, їх зміст та відношення.

Діаграма класів, також, може містити позначення для пакетів та може містити позначення для вкладених пакетів. Також, діаграма класів може містити позначення деяких елементів поведінки, однак їх динаміка розкривається в інших типах діаграм.

Діаграма класів (class diagram) служить для представлення статичної структури моделі системи в термінології класів об'єктно-орієнтованого

програмування. На цій діаграмі показують класи, інтерфейси, об'єкти й кооперації, а також їхні відносини.

В UML існують наступні типи зв'язків які використовуються у діаграмі класів: Асоціації; Агрегація; Композиція.

Асоціації це якщо між двома класами визначена асоціація, то можна переміщатися від об'єктів одного класу до об'єктів іншого. Цілком припустимі випадки, коли обидва кінці асоціації відносяться до одного і того ж класу. Це означає, що з об'єктом деякого класу дозволено зв'язати інші об'єкти з того ж класу. Асоціація, що зв'язує два класи, називається бінарної. Можна, хоча це рідко буває необхідним, створювати асоціації, що зв'язують відразу кілька класів. Графічно асоціація зображується у вигляді лінії, що з'єднує клас сам з собою або з іншими класами.

Асоціації може бути присвоєно ім'я, яке описує природу відносини. Зазвичай ім'я асоціації не вказується, якщо тільки ви не хочете явно задати для неї рольові імена або у вашій моделі настільки багато асоціацій, що виникає необхідність посилатися на них і відрізняти один від одного. Ім'я буде особливо корисним, якщо між одними і тими ж класами існує кілька різних асоціацій.

Клас, що бере участь в асоціації, грає в ній деяку роль. По суті, це "обличчя", яким клас, що знаходиться на одній стороні асоціації, звернений до класу з іншого її боку. Можна явно позначити роль, яку клас грає в асоціації.

Часто при моделюванні буває важливо вказати, скільки об'єктів може бути пов'язано допомогою одного примірника асоціації. Це число називається кратністю (Multiplicity) ролі асоціації та записується або як вираз, значенням якого є діапазон значень, або в явному вигляді.

Вказуючи кратність на одному кінці асоціації, ви тим самим говорите, що на цьому кінці саме стільки об'єктів повинно відповідати кожному об'єкту на протилежному кінці. Кратність можна задати рівною одиниці (1), можна вказати діапазон: "нуль або одиниця" (0..1), "багато" (0.. *), "одиниця або більше" (1.. *). Дозволяється також вказувати певне число (наприклад, 3). За допомогою списку

можна задати і більш складні кратності, наприклад 0.. 1, 3..4, 6.. *, що означає "будь-яке число об'єктів, крім 2 і 5".

Агрегація це проста асоціація між двома класами відображає структурний відношення між рівноправними сутностями, коли обидва класу знаходяться на одному концептуальному рівні і ні один не є більш важливим, ніж інший. Але іноді доводиться моделювати відношення типу «частина/ціле», в якому один з класів має більш високий ранг (ціле) і складається з декількох менших за рангом (частин).

Ставлення такого типу називають агрегацією; воно зараховане до відносин типу «має» (з урахуванням того, що об'єкт-ціле має кілька об'єктів-частин). Агрегація є окремим випадком асоціації і зображується у вигляді простої асоціації з незафарбованим ромбом з боку «цілого». Графічно агрегація представляється порожнім ромбом на блоці класу, і лінією, яка від цього ромба до міститься класу.

Композиція це більш суворий варіант агрегації. Відома також як агрегація за значенням.

Композиція має жорстку залежність часу існування екземплярів класу контейнера та примірників містяться класів. Якщо контейнер буде знищений, то весь його вміст буде також знищено. Графічно представляється як і агрегація, але з зафарбовані ромбиком.

Діаграма компонент в UML це діаграма, на якій відображаються компоненти, залежності та зв'язки між ними.

Діаграма компонент відображає залежності між компонентами програмного забезпечення, включаючи компоненти вихідних кодів, бінарні компоненти, та компоненти, що можуть виконуватись.

Модуль програмного забезпечення може бути представлено в якості компоненти. Деякі компоненти існують під час компіляції, деякі – під час компонування, а деякі під час роботи програми.

Діаграма компонент відображає лише структурні характеристики, для відображення окремих екземплярів компонент слід використовувати діаграму розгортування.

Компоненти об'єднуються разом використовуючи структурні зв'язки (assembly connector) щоб об'єднати інтерфейси двох компонент. Це ілюструє зв'язок типу «клієнт-сервер».

Структурна взаємодія – «зв'язок двох компонент, який передбачає, що один з них надає послуги, потрібні іншому компоненту».

При використанні діаграми компонент щоб показати внутрішню структуру компонента, клієнтські та серверні інтерфейси можуть утворювати пряме з'єднання з внутрішніми. Таке з'єднання називається з'єднанням делегації.

Діаграма об'єктів в UML це діаграма, що відображає об'єкти та їх зв'язки в певний момент часу. Діаграма об'єктів може розглядатись як окремий випадок діаграми класів, на якій можуть бути представлені як класи, так і екземпляри (об'єкти) класів. Схожою за змістом є діаграма взаємодії (collaboration diagram).

Діаграми об'єктів не мають власної нотації. Оскільки діаграми класів можуть відображати об'єкти, то діаграма класів, на якій відображено лише об'єкти, та не відображено класи, може вважатись діаграмою об'єктів.

Діаграма об'єктів відображає об'єкти та зв'язки в певний момент роботи програми. Об'єкти можуть містити інформацію про власні значення а не про описання. Для відображення загальних шаблонів об'єктів та зв'язків, що можуть багаторазово створюватись під час роботи програми, слід використовувати діаграму взаємодії, яка може відображати характеристики об'єктів та зв'язків. Екземпляр діаграми взаємодії створює діаграму об'єктів.

Діаграма об'єктів не відображає еволюцію системи під час роботи. Натомість, слід використовувати діаграми взаємодії з повідомленнями, або діаграми послідовності.

Діаграма розгортання (deployment diagram) це діаграма в UML, на якій відображаються обчислювальні вузли під час роботи програми, компоненти, та

об'єкти, що виконуються на цих вузлах. Компоненти відповідають представленню робочих екземплярів одиниць коду. Компоненти, що не мають представлення під час роботи програми на таких діаграмах не відображаються; натомість, їх можна відобразити на діаграмах компонент. Діаграма розгортання відображає робочі екземпляри компонент, а діаграма компонент, натомість, відображає зв'язки між типами компонент.

4.2 Захист розробленого програмного забезпечення

Дані які використовуються у даній роботі захищаються алгоритмом ДСТУ 9041:2020. Його повна назва: ДСТУ 9041:2020. Інформаційні технології. Криптографічний захист інформації. Алгоритм шифрування коротких повідомлень, що ґрунтується на скручених еліптичних кривих Едвардса.

Цей алгоритм призначений для шифрування коротких (до 616 біт) повідомлень для будь-яких алгоритмів шифрування, в тому числі визначених національними стандартами України.

Як і стандарт цифрового підпису ДСТУ 4145:2002, новий алгоритм використовує криптографічні перетворення у групі точок еліптичних кривих, використовуючи замість кривих у формі Вейєрштрасса найновітніші розробки у галузі еліптичної криптографії – криві у формі Едвардса. Це дає суттєві переваги у швидкодії більш ніж у 3 рази.Новий стандарт розроблений з урахуванням усіх найсучасніших вимог до стійкості криптографічних алгоритмів. Так, нижня межа стійкості криптосистем у цьому стандарті дорівнює 2127 (≈1042) (це більш ніж у півтора рази вище, ніж у ДСТУ 4145) і можуть бути обрані інші рівні, такі як 2255 (≈1085), 2383 (≈10127) та 2767 (≈10255); крім того, строго обґрунтована його стійкість як до атак на відновлення відкритого тексту, так і до розрізнюючих атак.

Проект алгоритму шифрування, що ліг в основу цього стандарту, пройшов апробацію як в Україні, так і за її межами (Центрально-Європейська конференція з криптографії (червень 2020 року) – форум ведучих криптологів з усього світу).

Стандарт ДСТУ-9041 узгоджений з усіма діючими в Україні національними стандартами. Новиною стандарту є його сфера застосування – інкапсуляція ключів, найсучасніший математичний апарат, а також новий алгоритм генерації псевдовипадкових послідовностей, який, на відміну від аналогічного алгоритму генерації з ДСТУ 4145, використовує виключно національні криптографічні алгоритми національних стандартів та не містить посилань на відповідні пост-радянські стандарти, термін дії яких вже практично вичерпався.

Новий стандарт не належить до так званих постквантових стандартів. Але його стійкість буде під загрозою лише тоді, коли з'являться квантові комп'ютери з 700 і більше кубітами (на даний час кількість "робочих" кубітів, які вдалося створити, – близько 50). Його перевагою перед постквантовими алгоритмами є відносно невелика довжина ключа (у десятки або навіть у сотні разів менша, ніж у постквантових алгоритмах).

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

На рисунку 5.1 зображено інтерфейс програмного забезпечення, розробленого у результаті виконання магістерської роботи. Розроблене програмне забезпечення комплексних рішень для відеонагляду складається з наступних функціональних блоків:

– Навігаційне меню: Файл; Камери; Налаштування; Вікна; Довідка.

– Вікно виведення відеосигналу.

– Навігаційного меню яке викликається натисканням правої клавіші маніпулятора миші.

– Функціональних кнопок ПЗ.

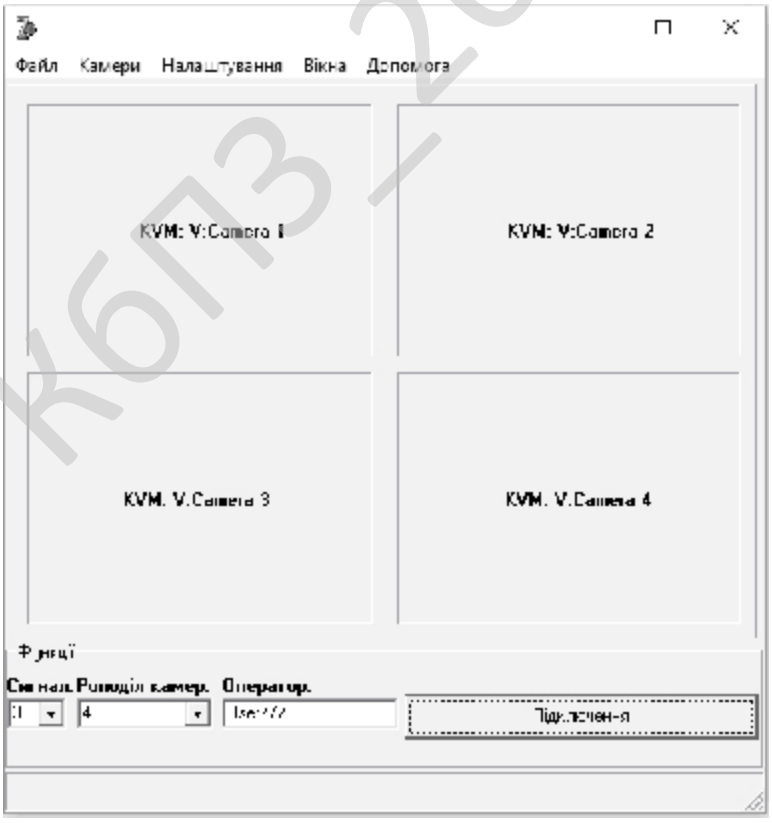

Рисунок 5.1 – Головне вікно розробленого ПЗ

Для перегляду короткої довідки про програму слід натиснути на основному вікні кнопку авторського права, після чого на екрані з’явиться вікно показане на рисунку 5.2.

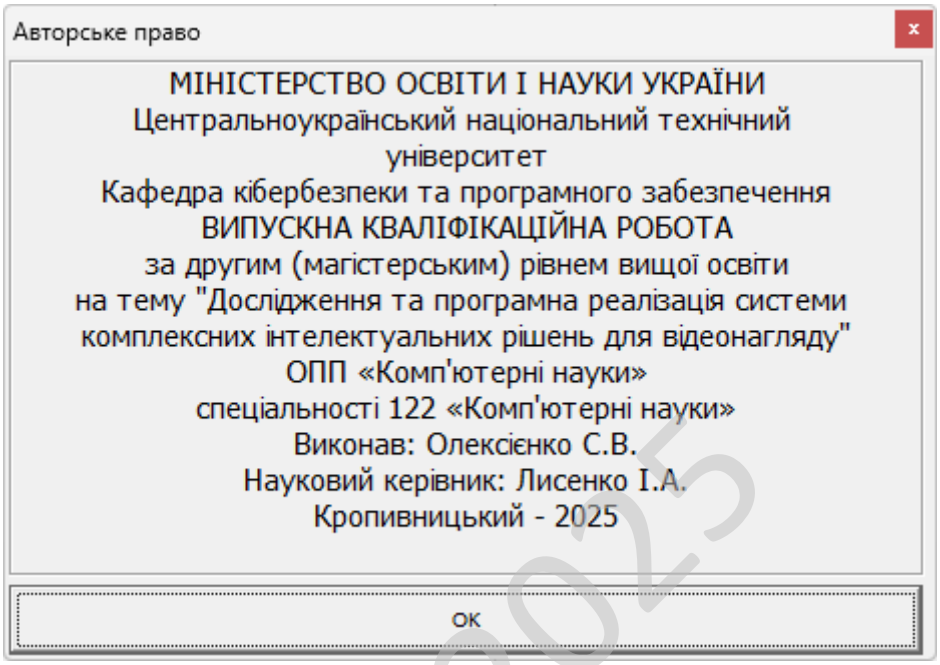

Рисунок 5.2 – Вікно розробника ПЗ

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.

Проводилось тестування форматом чорної скриньки. Основне місце програми тестів «чорної скриньки» – інтерфейс ПЗ. Відомі: функції програми. Досліджується: робота кожної функції на всій області визначення.

Ці тести демонструють:

– Як виконуються функції програми.

– Як приймаються вихідні дані.

– Як виробляються результати.

– Як зберігається цілісність зовнішньої інформації.

При тестуванні «чорної скриньки» розглядаються системні характеристики програм, ігнорується їхня внутрішня логічна структура. Вичерпне тестування, як правило, неможливе.

Наприклад, якщо в програмі 10 вхідних величин і кожна приймає по 10 значень, то кількість тестових варіантів становитиме 10^10. Тестування «чорної скриньки» не реагує на багато особливостей програмних помилок.

Тестування «чорної скриньки» (функціональне тестування) дозволяє отримати комбінації вхідних даних, які забезпечують повну перевірку всіх функціональних вимог до програми.

Програмний виріб тут розглядається як «чорна скринька», чию поведінку можна визначити тільки дослідженням його входів та відповідних виходів. При такому підході бажано мати:

– Набір, утворений такими вхідними даними, які призводять до аномалій у поведінці програми (назвемо його ITc).

– Набір, утворений такими вхідними даними, які демонструють дефекти програми (назвемо його OT).

Будь-який спосіб тестування «чорної скриньки» повинен:

– Виявити такі вхідні дані, які з високою ймовірністю належать набору ITc;

– Сформулювати такі очікувані результати, які з високою імовірністю є елементами набору ОТ.

Принцип «чорної скриньки» не альтернативний принципу «білої скриньки». Скоріше це доповнює підхід, який виявляє інший клас помилок.

Тестування «чорної скриньки» забезпечує пошук наступних категорій помилок:

– Некоректних чи відсутніх функцій;

– Помилок інтерфейсу;

– Помилок у зовнішніх структурах даних або в доступі до зовнішньої бази даних;

– Помилок характеристик (необхідна ємність пам'яті і т.д.);

– Помилок ініціалізації та завершення.

Обрано умови розповсюдження – commercial software.

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

Найважливішою особливістю комерційний 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 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.

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи комплексних інтелектуальних рішень для відеонагляду.

Метою розробки є дослідження та програмна реалізація системи комплексних інтелектуальних рішень для відеонагляду.

Об'єктом дослідження є процес комплексних інтелектуальних рішень для відеонагляду.

Предметом дослідження є методи комплексних інтелектуальних рішень для відеонагляду.

Методи дослідження базуються на методах розпізнавання образів, методах великих даних, методах комп'ютерних мереж,, методах математичної статистики, методах розробки програмного забезпечення.

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

– Удосконалено метод комплексних інтелектуальних рішень для відеонагляду.

– Розроблено вітчизняний продукт комплексних інтелектуальних рішень для відеонагляду, який має більш широкі можливості, на відміну від існуючих аналогів.

7 МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ ІТ-ПРОЄКТУ

7.1 Визначення цільової аудиторії кінцевого готового продукту

Результати дослідження та реалізації цієї системи можуть зацікавити широке коло користувачів – від великих промислових підприємств і торговельних мереж до державних установ, муніципалітетів і приватних компаній, які приділяють увагу безпеці. У сучасному світі, де питання контролю, кіберзахисту та моніторингу є надзвичайно актуальними, попит на автоматизовані системи відеоспостереження з інтелектуальними функціями зростає щороку. Особливу зацікавленість проявляють компанії, які працюють у сфері логістики, транспорту, банківського сектору та енергетики, адже для них навіть короткочасний збій безпеки може спричинити суттєві збитки.

Для органів державної влади та місцевого самоврядування такі рішення можуть бути корисними у реалізації концепції “розумного міста”. Інтелектуальні системи відеоаналітики допомагають не лише відстежувати правопорушення, а й прогнозувати поведінку натовпу, контролювати трафік, виявляти потенційно небезпечні об’єкти. Вони сприяють підвищенню ефективності роботи поліції та служб реагування.

Не менш важливим є інтерес з боку освітніх і медичних закладів, де питання безпеки мають особливу соціальну вагу. У таких установах інтелектуальна система відеонагляду може не лише фіксувати події, але й автоматично сигналізувати про підозрілу активність або порушення внутрішніх правил, що підвищує рівень безпеки персоналу й відвідувачів.

7.2 Оцінка привабливості шляхом застосування методів експертних оцінок

Оцінювання привабливості проєкту можна здійснити шляхом експертного аналізу, у якому беруть участь фахівці з різних галузей – IT-безпеки, відеоаналітики, управління ризиками, економіки та маркетингу. Кожен експерт оцінює проєкт за низкою критеріїв: рівень інноваційності, складність впровадження, потенціал комерціалізації, масштабованість і попит на ринку. Результати зводяться до інтегрального індексу привабливості, який дозволяє визначити, наскільки перспективним є продукт у контексті сучасних технологічних тенденцій.

За результатами оцінювання, система комплексних інтелектуальних рішень отримала високі оцінки за такими параметрами, як технологічна інноваційність і ринковий потенціал. Експерти зазначили, що використання штучного інтелекту у відеонагляді значно підвищує точність виявлення інцидентів, знижує людський фактор і дозволяє автоматизувати аналітичні процеси. Окрім цього, ринок таких рішень демонструє стабільне зростання, що забезпечує комерційну привабливість проєкту.

Таким чином, експертна оцінка підтвердила доцільність розробки й подальшого впровадження цієї системи. Вона не лише відповідає запитам ринку, але й формує нову нішу в галузі “розумних” технологій відеоспостереження, здатних не просто реагувати на події, а й передбачати їх.

7.3 Вибір методу оцінки вартості ПЗ

Для визначення вартості впровадження такої системи найдоцільніше застосовувати витратний метод у поєднанні з порівняльним підходом. Витратний метод дозволяє точно оцінити вартість розробки програмного забезпечення,

закупівлі обладнання (камер, серверів, мережевого обладнання), а також витрати на інсталяцію, навчання персоналу та технічне обслуговування.

Порівняльний підхід допомагає визначити конкурентну ціну, орієнтуючись на аналогічні рішення на ринку, такі як Hikvision, Dahua або Axis Communications. Завдяки цьому можна зрозуміти, наскільки вигідно позиціонується власна система за показниками “ціна–якість–функціональність”. У випадку, коли система має унікальні можливості (наприклад, інтеграцію з ШІ або автоматичний аналіз поведінки), це дозволяє обґрунтовано встановити вищу ціну, ніж у конкурентів.

Поєднання цих двох методів забезпечує найбільш об’єктивну оцінку, оскільки враховує як фактичні витрати на реалізацію, так і ринкові умови, у яких планується подальше просування продукту.

7.4 Розрахунок економічної ефективності від впровадження реалізованого ПЗ як фактору його привабливості

Підприємство планує оновити свою систему відеоспостереження, замінивши традиційні камери на інтелектуальні пристрої з вбудованим аналізом відео.

Система має забезпечувати: автоматичне розпізнавання облич і номерних знаків; виявлення підозрілої поведінки та небезпечних ситуацій (наприклад, залишені предмети, вторгнення в заборонену зону); аналітику потоків для прогнозування ризиків; централізоване управління всіма об’єктами через хмарну платформу.

Основна мета впровадження – зменшення витрат на фізичну охорону, підвищення рівня безпеки, а також скорочення втрат від інцидентів за рахунок швидкого реагування та профілактики загроз. Вхідні дані зафіксовано в таблиці 7.1.

Таблиця 7.1 – Вихідні дані для розрахунку

Показник	До впровадження	Після впровадження	Економічний ефект
Витрати на фізичну охорону (річні)	3 000 000 грн	1 500 000 грн	–1 500 000 грн
Кількість зареєстрованих інцидентів за рік	24	6	–75 %
Середні збитки від одного інциденту	50 000 грн	10 000 грн	–80 %
Витрати на обслуговування системи	200 000 грн	400 000 грн	+200 000 грн
Початкові інвестиції у впровадження	—	2 500 000 грн	—

Розрахунок економічного ефекту демонструє наступне: зменшення витрат на охорону – 1 500 000 грн/рік, зменшення збитків від інцидентів – 1 140 000 грн/рік, додаткові витрати на обслуговування системи – +200 000 грн/рік, річний економічний ефект – 2 440 000 грн/рік, чистий економічний ефект – –60 000 грн (окупність через 1 рік), термін окупності ≈ 1,03 року, рентабельність інвестицій – 97,6 %.

Нефінансові результати впровадження: підвищення рівня безпеки персоналу і клієнтів – завдяки автоматичному сповіщенню про інциденти система скорочує час реагування охорони у 5–10 разів, покращення управлінських рішень – аналітика з відеопотоків дозволяє керівництву

оптимізувати логістику, контроль доступу та навантаження на персонал, зниження людського фактору – система автоматизує процес моніторингу, мінімізуючи помилки операторів і втому під час чергувань, підвищення репутації компанії – високий рівень безпеки підвищує довіру клієнтів і партнерів, що може позитивно впливати на фінансові результати.

Крім прямого фінансового результату, впровадження таких рішень має стратегічне значення – воно формує цифрову екосистему безпеки, здатну інтегруватися з іншими корпоративними системами (СКУД, ERP, IoT), що в перспективі створює основу для побудови "розумного підприємства" з автоматизованим управлінням усіма процесами ризиків і безпеки.

7.5 Пропозиція алгоритму просування проєкту розробки ПЗ

Першим кроком у просуванні проєкту має бути створення детального прототипу системи з демонстрацією її ключових можливостей – розпізнавання облич, аналізу руху, прогнозування загроз. Така демонстраційна версія допоможе потенційним клієнтам наочно оцінити переваги рішення. На другому етапі варто організувати публічну презентацію або вебінар для представників бізнесу, IT-компаній і муніципалітетів. Це дозволить поширити інформацію про проєкт серед потенційних користувачів і партнерів.

Паралельно доцільно запустити цифрову маркетингову кампанію – створити вебсайт продукту, наповнений відеооглядами, аналітикою і прикладами використання системи. Соціальні мережі, галузеві форуми та профільні виставки допоможуть сформувати експертне враження про продукт.

Завершальним етапом має стати укладення партнерських угод із інтеграторами систем безпеки, які можуть пропонувати продукт своїм клієнтам як частину комплексних рішень. Саме партнерська мережа стане ключовим рушієм для масштабування продукту на ринку.

7.6 Оптимізація каналів збуту та шляхів реалізації ПЗ

Оптимізація каналів збуту повинна базуватися на поєднанні прямих і партнерських продажів. Власні продажі забезпечують контроль над брендом і якістю впровадження, тоді як партнерські дозволяють швидко розширити географію присутності. Розумним рішенням є створення моделі “white label”, яка дозволить іншим компаніям впроваджувати систему під власним брендом, забезпечуючи додатковий дохід без великих витрат.

Окрему увагу варто приділити цифровим каналам – SEO, контекстній рекламі та PR-кампаніям у професійних спільнотах. Для залучення корпоративних клієнтів ефективним буде прямий маркетинг через спеціалізовані виставки з безпеки, презентації на конференціях і спільні проєкти з технологічними гігантами, які працюють у сфері кібербезпеки або хмарних рішень.

З іншого боку, важливо продумати систему технічної підтримки й регулярних оновлень, оскільки це підвищить довіру клієнтів і дозволить утримати їх у довгостроковій перспективі.

7.7 Визначення ключових факторів успіху конкретного проєкту

Основним фактором успіху є технологічна надійність і точність роботи системи. Клієнти очікують, що система не просто фіксуватиме події, а й аналізуватиме їх у режимі реального часу без хибних спрацювань. Для цього необхідно використовувати сучасні алгоритми глибинного навчання, які здатні самостійно вдосконалюватися на основі накопичених даних.

Другим фактором є зручність інтеграції. Система має легко взаємодіяти з іншими рішеннями безпеки – системами контролю доступу, пожежної сигналізації, ERP або CRM. Простота інтеграції значно збільшує цінність продукту для бізнесу.

Третім важливим чинником є економічна доцільність. Замовники прагнуть бачити реальні фінансові переваги: зменшення витрат на охорону, запобігання інцидентам, швидку окупність. Якщо ці аргументи підкріплені розрахунками, ймовірність успішної комерціалізації зростає у кілька разів.

Зрештою, репутація бренду і наявність надійної команди підтримки також є критично важливими. Адже в галузі безпеки довіра клієнта визначає не лише успіх продажів, але й перспективу розвитку компанії на роки вперед.

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

Охорона праці – система збереження життя і здоров'я працівників у процесі трудової діяльності, що включає правові, соціально-економічні, організаційні, технічні, санітарно-гігієнічні, лікувально-профілактичні, реабілітаційні та інші заходи.

Згідно закону України “Про охорону праці” [1] кожна компанія впроваджує заходи з охорони праці. Реалізуються 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.

Законом України “Про охорону праці” [2] 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» [3], 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 роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98 [4].

Програмісти у процесі роботи мають негативний вплив на органи зору, а також мають значну розумову напругу і нервово-емоційне навантаження. Руки

(суглоби пальців та м'язи рук) при роботі з клавіатурою мають теж істотне навантаження. До шкідливих факторів, які впливають на робітників галузі інформаційних технологій (ІТ), спеціалісти відносять високочастотні електромагнітні коливання (випромінювання) при роботы апаратної частини ЕОМ та виділення шкідливих газів.

Ці шкідливі фактори можуть привести до професійних захворювань.

Науково-технічний прогрес суттєво вплинув на умови виробничої діяльності робітників розумової діяльності. Їх праця стала більш інтенсивною, напруженою і вимагає значних витрат розумової, емоційної і фізичної енергії. Це призвело до необхідності у знаходженні комплексного рішення проблем ергономіки, гігієни і організації праці, регламентації режимів праці та відпочинку. Охорона здоров'я робітників, забезпечення безпеки умов праці, ліквідація та профілактика професійних захворювань і виробничого травматизму складає одну з головних турбот людського суспільства.

8.2 Аналіз умов праці

У приміщенні розташовано 3 робочих місця з комп'ютерами (далі ПК). Відповідно до норм «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98 [2] площа, що відводиться для робочого місця з комп'ютером повинна бути не менше 6 м^2, об'єм не менше 20 м^3. Розміри даного приміщень складають: довжина – 6 м, ширина – 4,5 м, висота – 3,5 м, тобто загальна фактична площа складає 27 м^2. Необхідна площа на 3 робочих місця із установленими ПК складає 18 м^2, що не перевищує фактичну. Обсяг приміщення на одного працюючого складає 31,5м^3, отже відповідає нормі ДСанПіН 3.3.2-007-98 – не менше 20 м^3.

При роботі з ПК людина може піддатися впливу шкідливих та небезпечних факторів. Під шкідливими виробничими факторами розуміють фактори, тривалий вплив яких викликає розвиток професійних захворювань.

Небезпечні виробничі фактори – вплив яких на працюючого викликає травму, тобто пошкодження організму. Шкідливі і небезпечні чинники, з якими стикається бібліограф при роботі з ПК, приведені в таблиці 8.1.

Таблиця 8.1 – Перелік шкідливих та небезпечних виробничих факторів

Найменування факторів	Можливі джерела їх виникнення	Характер дії
Небезпека ураження електричним струмом	Мережа живлення	Небезпечний
Пожежонебезпечність приміщень	Наявність матеріалів, що згорають і джерел запалення (електроапаратура)	Небезпечний та шкідливий
Іонізація повітря	Статична електрика випромінювання	Шкідливий
Підвищений рівень шуму	Шум створюється перетворювачем напруги ЕОМ, її технічною периферією, а також людьми, що працюють в приміщенні	Шкідливий
Несприятлива освітленість	Недостатнє штучне і природне освітлення	Шкідливий
Незадовільні параметри мікроклімату	Незадовільний стан системи опалення і вентиляції	Шкідливий
Психофізіологічні напруження	Монотонність праці, перенапруженість зорових аналізаторів, розумова напруженість, незручність і статичність пози	Шкідливий

За категорією вибухо- і пожежонебезпеки, дане приміщення відноситься до категорії В – пожежонебезпечне, тому що присутні тверді матеріали, що горять, такі як дерев'яні столи, папір і інше. Виходячи з категорії пожежонебезпеки і поверховості будинку, ступінь вогнестійкості будівлі II. Згідно з ДБН В 1.1-7-2016 «Пожежна безпека об’єктів будівництва» [3] ЕОМ повинні розташовуватись в будівлі не менше ніж II ступню вогнестійкості.

За ступенем небезпеки поразки людей електричним струмом приміщення класифікується як приміщення з підвищеною небезпекою, тому що не виключена можливість одночасного дотику людини до маючих з'єднання з землею конструкціям будинку, з одного боку, і до металевих корпусів електроустаткування, що можуть виявити під напругою – з іншого.

Для забезпечення вищевказаних оптимальних метеорологічних умов у помешканні передбачена система опалення (загальне парове) в холодному періоді, та вентиляція і кондиціювання в теплий період року, згідно ДБН2.5–67–2013 «Опалення, вентиляція та кондиціювання» [4]. При виконанні замірів параметрів мікроклімату, значення їх відповідали оптимальним та допустимим параметрам відповідно до ДСанПіН3.3.2.007 – 98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин».

Припустимий рівень іонізації повітря помешкання відповідно до СН 21.52-80 повинен складати 1500 – 3000 один./м^3.

Нормування освітлення здійснюється відповідно до ДБН В.2.5 – 28 – 2006 «Природне та штучне освітлення». [5]

Відділ забезпечений комбінованим освітленням. В темний час доби передбачається загальне і/або місцеве рівномірне штучне, а в світлий – бокове одностороннє природне освітлення два віконних прорізи.

Одним з найбільш поширеніших чинників зовнішнього середовища, який несприятливо впливає на людину, є шум. Вплив шуму на організм людини

залежить від рівня звукового тиску, частотних характеристик, тривалості дії, а також індивідуальних особливостей людини.

При тривалій дії шуму у оператора ЕОМ виявляються симптоми утомленості, нервового збудження, що сприяють погіршенню працездатності і допущенні помилок при роботі. Для уникнення шкідливої дії шуму на організм працюючого, необхідне дотримання нормованих параметрів, які не повинні перевищувати допустимих величин. При роботі на комп'ютері рівень шуму не повинен перевищувати 50 дБА. Приміщення розташоване вікнами у двір і знаходиться далеко від проїжджої частини вулиці. Основними джерелами шуму в приміщенні є устаткування і люди. Розглянута кімната не призначена для прийому відвідувачів і тому в ній не спостерігається великого скупчення людей. Тому основним джерелом шуму є комп'ютерна техніка.

Джерелами шуму при роботі ЕОМ є механічні частини принтера, що рухаються, і вентилятори ($L_{пк}$ = 35 дБА, L_{prn}= 48 дБА) При роботі вентиляційної системи, що забезпечує оптимальний температурний режим електронних блоків ЕОМ і вмонтована в задню панель, створюється аеродинамічний шум. Шум, створюваний працюючим комп'ютером, може бути охарактеризований як широко смуговий постійний з аперіодичним посиленням при роботі принтера. Час роботи ПЕОМ – 6 – 8 год. за добу; принтери працюють не більш 1,5-2 год. за добу.

При наявності великої кількості джерел шуму еквівалентне значення шуму $L_{ЭКВ}$, дБА розраховують за наступною формулою:

$$L_{скв}=10\cdot\lg\left(\frac{1}{T}\sum_{i=1}^{n}\left(t_i\cdot 10^{0.1\cdot L_i}\right)\right) \qquad (8.1)$$

де

L_i – рівень шуму і-го джерела (пристрою),

t_i – час роботи і-го джерела (пристрою),

T – загальний час роботи,

n – кількість джерел шуму даного типу;

Для даного приміщення необхідні змінні складають:

Загальний час роботи – робітник день, тобто T=8 годин.

Для фонового шуму (вентиляторів):

$L_1 = 35$ дБА, $T_1 = 8$ годин, $n_1=15$ (5х3);

Для лазерного принтера Lexmark Jet:

$L_2 = 48$дБА, Т2 = 2 години, $n_2=1$, для сканера $L_3 = 46$дБА, Т3 = 2 години.

Підставляємо отримані величини у формулу (8.1):

$$L_{екв} = 10 \cdot \lg\left(\frac{1}{8} \cdot \left(15 \cdot 8 \cdot 10^{0,1 \cdot 35} + 1 \cdot 2 \cdot 10^{0,1 \cdot 48} + 1 \cdot 2 \cdot 10^{0,1 \cdot 46}\right)\right) = 46{,}3 \text{ дБА}$$

Таким чином, еквівалентний рівень шуму в приміщенні за робочий день $L_{екв} = 46{,}3$ дБА , тобто не перевищує норму 50 дБА.

8.3 Техніка безпеки та протипожежна профілактика

Відповідно ДБН В 1.1-7-2016 «Пожежна безпека об’єктів будівництва» будинок можна віднести до II групи по ступені вогнестійкості й до категорії Д по ступені пожежонебезпеки.

Від розподільного щита по праву й ліву сторони встановлені кондиціонери, зовнішня електропроводка, поміщена в ізольований кабель. Висота проводки становить 2,2 м від рівня підлоги, її кріплення здійснюється за допомогою металевих власників. Біля кожного стола організований розподільний щит, розташований на текстолітовій пластинці, закріпленої на стіні на рівні 1м від підлоги. Усього до складу входять п'ять розеток і дві клеми заземлення. Всі обчислювальні машини з'єднані із клемами заземлення. Чотири з п'яти розеток забезпечують подачу напруги 220 В, а одна, забезпечує подачу напруги в 36 В. Про це є відповідні написи на кожному розподільному щиті.

Робота обслуговуючого персоналу полягає в інсталяції необхідного програмного забезпечення й наступному його використанні в діалоговому режимі роботи з ЕОМ. Іноді може виникати необхідність написання допоміжних програм для поліпшення роботи вузла або для зниження витрат. З погляду забезпечення умов праці й вимог техніки безпеки для роботи програміста необхідно наступне: достатнє висвітлення екрана дисплея й робочого місця; повна технічна справність

устаткування, його електробезпечність; достатня пожежобезпечність приміщення; оптимальний мікроклімат, що сприяє продуктивній роботі; відповідність робочого місця вимогам ергономіки.

До небезпечних і шкідливих факторів, дії яких піддається програміст, можна віднести: можливість поразки електричним струмом, при електроні справності встаткування, порушенні заземлення або техніки безпеки; робота в мікрокліматі з неприпустимими параметрами; робота при недостатній освітленості екрана дисплея й робочого місця.

Відповідно НПАОП 40.1-1.21-98 “Правил безпечної експлуатації електроустановок споживачів” [6] приміщення можна віднести до приміщень без підвищеної небезпеки, оскільки це приміщення, сухе, з нормальною температурою й ізолюючими підлогами, що не має заземлених металоконструкцій.

Персональні ЕОМ можна віднести до першого класу електротехнічних виробів по способі захисту людини від поразки електричним струмом, оскільки їхні корпуси зроблені з ізолюючої пластмаси й кожен пристрій має заземлення. Відповідно правилам пристрою електроустановок ЕОМ можна віднести до електроустановок з робочою напругою до 1000 В.

Однією з достовірних причин пожежі в приміщенні з обчислювальною технікою може бути коротке замикання, що спричиняє спалах електропроводки. Для його попередження вся обчислювальна техніка, а також інші електричні пристрої повинні бути обладнані плавкими запобіжниками, а на вході електромережі повинен бути передбачений автомат захисту. Не слід користуватися електричними подовжувачами й трійниками, що не мають сертифікатів відповідності вимогам безпеки.

Необхідно передбачити наявність у межах досяжності первинних засобів гасіння пожежі (вогнегасників) для локалізації вогню власними засобами до приїзду команди пожежної охорони. Повинен бути розроблений план екстреної евакуації персоналу при виникненні загоряння. Кількість евакуаційних виходів

повинне бути не менш двох. Допускається використання одного евакуаційного виходу, якщо відстань найбільш віддаленого робочого місця до цього виходу не перевищує 25 м.

8.4 Розробка заходів з охорони праці

Перерахуємо проведені заходи щодо забезпечення умов праці на робочому місці програміста.

Для зменшення шуму в приміщенні пропонється використовувати замість матричного принтера, що створює багато шуму, більш тихий – лазерний принтер.

З точки зору забезпечення електробезпеки до цих заходів можна віднести: устаткування розподільних щитів спеціальними розетками з заземлюючими контактами; організація заземлення всіх приладів і пристроїв; періодична перевірка всіх приладів і пристроїв; щорічна здача іспитів з охорони праці.

З точки зору забезпечення оптимальних умов мікроклімату і освітленості до цих заходів можна віднести: організацію природної вентиляції, за допомогою дефлектора, для забезпечення необхідного повітрообміну в приміщенні вузла; організацію системи центрального опалювання, для підтримки оптимальної температури в холодний період року; організацію штучного загального освітлення, для забезпечення необхідних умов зорової роботи, що відповідають, оформлення паспорта на приміщення вузла, з занесенням в нього вимірювань освітленості, проведених відділом охорони праці.

Для робочої зони виробничих приміщень встановлюються оптимальні та допустимі мікрокліматичні умови з урахуванням важкості виконуваної роботи та періоду року. При одночасному виконанні в робочій зоні робіт різної категорії важкості рівні показників мікроклімату повинні встановлюватись з урахуванням найбільш чисельної групи працівників.

Як міри для зниження шуму можна запропонувати:

– облицювання стелі і стін звукопоглинаючим матеріалом (знижують

шум на 6-8 дб);

– екранування робочого місця (встановленням перегородок, діафрагм);

– установка в комп'ютерних приміщеннях устаткування, що утворює мінімальний шум;

– раціональне планування приміщення.

З точки зору забезпечення пожежної безпеки до цих заходів можна віднести наявність схеми евакуації з приміщення вузла, у випадку пожежі, повішену на вхідні двері.

8.5 Висновки до розділу

У даному розділі магістерської роботи були виконано аналіз умов праці користувачів ПК, які працюють у зазначеному приміщенні Проведено перевірка організації робочого місця із відповідними замірами параметрів мікроклімату, освітлення, рівня шуму та розрахунком рівня шуму.

Розроблені заходи щодо поліпшення умов праці дотримання техніки безпеки та проведення протипожежної профілактики дозволить створити умови, які будуть забезпечувати більш комфортну роботу.

9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи комплексних інтелектуальних рішень для відеонагляду.

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 комплексних інтелектуальних рішень для відеонагляду.

Рішення даного завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем комплексних інтелектуальних рішень для відеонагляду.

– Досліджена система комплексних інтелектуальних рішень для відеонагляду.

– На основі отриманих результатів досліджень створена програмна реалізація системи комплексних інтелектуальних рішень для відеонагляду.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання комплексних інтелектуальних рішень для відеонагляду.

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.

При створені 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 ДСТУ 9041:2020.

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 проекту.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Олексієнко С.В. Дослідження та програмна реалізація системи комплексних інтелектуальних рішень для відеонагляду // Збірник праць молодих науковців ЦНТУ. – Вип. 15. – Кропивницький: ЦНТУ, 2025.

2. Ranjan Parekh. Fundamentals of Image, Audio, and Video Processing Using MATLAB®With Applications to Pattern Recognition. CRC Press. 2021. 406 p.

3. Alasdair McAndrew. A Computational Introduction to Digital Image Processing. Chapman & Hall. 2021. 560 p.

4. Peter Shirley, Steve Marschner. Fundamentals of Computer Graphics. 2009

5. Михайло Пічугін, Іван Канкін, Володимир Воротніков Комп'ютерна графіка. Навчальний посібник / Центр навчальної літератури 346 с. 2019р.

6. Маценко В.Г. Комп’ютерна графіка: Навчальний посібник. – Чернівці: Рута, 2009 – 343 с.

7. 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.

8. 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. ун-т. – Д.: НГУ, 2016. – 187 с.

9. Kopf, Johannes and Lischinski, Dani. Depixelizing Pixel Art (англ.) // ACM Trans. Graph. – 2011. – Vol. 30, no. 4. – P. 99:1--99:8.

10. Giachetti, Andrea and Asuni, Nicola. Real-Time Artifact-Free Image Upscaling (англ.) // Trans. Img. Proc.. – 2011. – Vol. 20, no. 10. – P. 2760—2768.

11. 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.

12. 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

13. Смірнова Т.В., Конопліцька-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.

14. 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.

15. 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.

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

17. 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.

18. 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.

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

20. 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.

21. 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

22. 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.

23. 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.

24. 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.

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

26. 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.

27. 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.

28. 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.

29. 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.

30. 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.

31. 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.

32. 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.

33. 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.

34. Zhurakovskyi, 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.

35. Smirnov O. Kuznetsov A., Zaicenko 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.

36. 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.

37. 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.

38. 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.

39. 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.

40. 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.

41. 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.

42. 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.

43. 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.

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

45. 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.

46. 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.

47. Смірнов О.А., Дрєєва Г.М., Дрєєв 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.

48. 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.

49. Смірнов О.А., Конопліцька-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.

50. 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).

51. Смірнов О.А., Дрєєва 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

52. Дрєєва Г.М., Смірнов О.А., Дрєєв 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.

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