

Вінницький національний технічний університет  
Факультет інформаційних технологій та комп'ютерної інженерії  
Кафедра обчислювальної техніки

## **МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА**

на тему:

**«Програмний засіб виділення контурів рухомих об'єктів в системах  
комп'ютерного відеоспостереження»**

Виконав: студент 2 курсу, групи КІ-20мз  
напряму підготовки (спеціальності)  
123 — «Комп'ютерна інженерія»  
\_\_\_\_\_ Лавренюк В. С.

Керівник: к.т.н., доц. каф. ОТ  
\_\_\_\_\_ Колесник І. С.

«\_\_\_\_» \_\_\_\_\_ 2022 р.

Опонент: к.т.н., доц. каф ЗІ  
\_\_\_\_\_ Дудатьєв А. В.

«\_\_\_\_» \_\_\_\_\_ 2022 р.

**Допущено до захисту**  
Завідувач кафедри ОТ  
д.т.н., проф. Азаров О.Д.  
«\_\_\_\_» \_\_\_\_\_ 2022 р.

Вінницький національний технічний університет  
Факультет інформаційних технологій та комп'ютерної інженерії  
Кафедра обчислювальної техніки  
Рівень вищої освіти II-й (магістерський)  
Галузь знань 12 — Інформаційні технології  
Спеціальність 123 — «Комп'ютерна інженерія»  
Освітня програма — «Комп'ютерна інженерія»

**ЗАТВЕРДЖУЮ**

Завідувач кафедри  
обчислювальної техніки  
\_\_\_\_\_ проф., д.т.н. О.Д. Азаров

«\_\_\_» \_\_\_\_\_ 2022 р.

**З А В Д А Н Н Я**

**НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ**

Лавренюку Вадиму Сергійовичу

1 Тема роботи «Програмний засіб виділення контурів рухомих об'єктів в системах комп'ютерного відеоспостереження»

керівник роботи Колесник Ірина Сергіївна, к.т.н., доцент,

затверджені наказом вищого навчального закладу від 24.03. 2022 р. № 65.

2 Строк подання студентом роботи 10.06.2022 р.

3 Вихідні дані до роботи: частота надходження кадрів відеопотоку не менше 25 кадрів/сек, розмір зображення — не менше 720 x 640 пікселів, модель кольорів для представлення зображення — RGB, кількість градацій яскравості зображення — 256.

4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити). Вступ. Огляд і аналіз існуючих методів виділення контурів об'єктів. Розробка методу виділення контурів рухомих об'єктів. Розробка послідовності виділення контурів рухомих об'єктів. Розробка програми виділення контурів рухомих об'єктів. Розрахунок економічної доцільності створення програми виділення контурів рухомих об'єктів.

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

Послідовність виділення рухомих об'єктів. Методи виділення рухомих об'єктів. Методи виділення контурів об'єктів. Схема комп'ютерної системи відеоспостереження. Структурна схема програми виділення контурів об'єктів.

6 Консультанти розділів роботи представлені в таблиці 1.

Таблиця 1— Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1-4	Колесник І. С., к.т.н., доцент кафедри ОТ		
5	Руда Л. П., к.е.н., доцент кафедри ЕПВМ		

7 Дата видачі завдання 25.03.2022 р.

Календарний план наведено в таблиці 2.

Таблиця 2—Календарний план

№	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Постановка задачі роботи	25.03.22	
2	Огляд існуючих методів виділення контурів рухомих об'єктів	28.03-09.03.22	
3	Аналіз та вибір методів виділення контурів рухомих об'єктів	10.03-28.03.22	
4	Розробка способу виділення контурів рухомих об'єктів	29.03-11.04.22	
5	Розробка послідовності виділення контурів рухомих об'єктів	12.04-21.04.22	
6	Розробка програми виділення контурів рухомих об'єктів	22.04-30.04.22	
7	Розробка інтерфейсу та тестування програми виділення контурів рухомих об'єктів	02.05-10.05.22	
8	Підготовка матеріалів та опис розробки засобів виділення контурів рухомих об'єктів ини	11.05-16.05.22	
9	Розрахунок економічної частини роботи	17.05-31.05.22	
10	Оформлення пояснювальної записки та ілюстративного матеріалу	01.06-06.06.22	
11	Аналіз виконання роботи, висновки, додатки	07.06-08.06.22	
12	Перевірка якості виконання магістерської роботи та усунення недоліків	09.06.22	

Студент \_\_\_\_\_ Лавренюк В. С.  
Керівник роботи \_\_\_\_\_ Колесник І. С.

## АНОТАЦІЯ

УДК 004.93

Лавренюк В. С. Програмний засіб виділення контурів рухомих об'єктів у системах комп'ютерного відеоспостереження. Магістерська кваліфікаційна робота зі спеціальності 123 — комп'ютерна інженерія, освітня програма комп'ютерна інженерія. Вінниця: ВНТУ, 2022. 125 с.

На укр. мові. Бібліогр.: 40 назв; рис. 4; табл. 13.

У магістерській кваліфікаційній роботі розроблено програмні засоби пошуку та виділення контурів рухомих об'єктів у виділеній сцені спостереження. У роботі зроблено аналіз способів побудови комп'ютерних систем відеоспостереження, запропоновано здійснювати процес виділення контурів рухомих об'єктів із відеопотоку із використанням методу міжкадрової різниці, виділення контуру об'єктів бінарного зображення та застосування згорткової нейронної мережі. В роботі розроблено послідовність обробки відео зображення для виявлення та виділення контурів рухомих об'єктів та програмне забезпечення, яке його реалізує в режимі реального часу. У роботі також виконані економічні розрахунки стосовно доцільності створення нового програмного продукту.

Графічна частина складається із 7 плакатів із результатами експериментальних досліджень.

Ключові слова: міжкадрова різниця, відеопотік, контур об'єкта, згорткова нейронна мережа.

## ANNOTATION

Lavreniuk V. Software for contouring moving objects in computer video surveillance systems. Master's thesis in specialty 123 - computer engineering, computer engineering education program. Vinnytsia: VNTU, 2022. 125 p.

In Ukrainian language. Bibliographer: 40 titles; fig.: 4; table 13.

In the master's qualification work developed software tools for finding and selecting the contours of moving objects in a selected observation scene. The paper analyzes the methods of construction of computer video surveillance systems, proposes to carry out the process of isolating the contours of moving objects from the video stream using the interframe difference method, isolating the contour of binary images and using convolutional neural network. The paper develops a sequence of video image processing to detect and highlight the contours of moving objects and software that implements it in real time. The paper also performs economic calculations on the feasibility of creating a new software product.

The graphic part consists of 7 posters with the results of experimental research.

Key words: interframe difference, video stream, object contour, convolutional neural network.

## ЗМІСТ

<b>ВСТУП</b> .....	8
<b>1 АНАЛІЗ МЕТОДІВ ТА ЗАСОБІВ ВИДІЛЕННЯ РУХОМИХ ОБ'ЄКТІВ</b> .....	11
1.1 Комп'ютерний зір як засіб виділення об'єктів .....	11
1.2 Аналіз методів виділення рухомих об'єктів .....	14
1.3 Аналіз методів виділення контурів об'єктів цифрових зображень .....	19
1.4 Етапи формування пошуку об'єктів.....	27
1.5 Нейромережа глибокого навчання .....	28
<b>2 РОЗРОБКА ПОСЛІДОВНОСТІ ВИЯВЛЕННЯ РУХОМИХ ОБ'ЄКТІВ ТА ВИДІЛЕННЯ ЇХ КОНТУРІВ</b> .....	36
2.1 Послідовність виявлення рухомих об'єктів.....	36
2.2 Механізм виявлення рухомих об'єктів .....	43
2.3 Розробка алгоритму виділення контуру об'єктів.....	45
2.4 Використання нейронної мережі для виділення об'єктів .....	49
<b>3 РОЗРОБКА ПРОГРАМНИХ ЗАСОБІВ ДЛЯ ВИДІЛЕННЯ КОНТУРІВ РУХОМИХ ОБ'ЄКТІВ</b> .....	53
3.1 Вибір інструментальних засобів для розробки програми .....	53
3.2 Вибір середовища розробки.....	57
3.3 Розробка складу програми виділення об'єктів .....	60
3.4 Розробка програми виділення контурів рухомих об'єктів.....	66
3.5 Склад комп'ютерної системи відеоспостереження .....	70
<b>4 ПЕРЕВІРКА ЯКОСТІ РОБОТИ СТВОРЕНОГО ПРОГРАМНОГО ПРОДУКТУ</b> .....	74
4.1 Вибір засобів для виконання тестування програми.....	74
4.2 Результати експериментальних досліджень .....	76

					<i>08-23.МКР.003.00.000 ПЗ</i>			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розроб.</i>		<i>Лавренюк В. С.</i>			Програмний засіб виділення контурів рухомих об'єктів в системах комп'ютерного відеоспостереження. Пояснювальна записка	<i>Лім.</i>	<i>Арк.</i>	<i>Аркушів</i>
<i>Перевір.</i>		<i>Колесник І. С.</i>					6	125
<i>Реценз.</i>		<i>Дудатьєв А.В.</i>				ВНТУ, КІ – 20м		
<i>Н. Контр.</i>		<i>Швець С. І.</i>						
<i>Затверд.</i>		<i>Азаров О. Д.</i>						

<b>5 РОЗРАХУНОК ЕКОНОМІЧНОЇ ДОЦІЛЬНОСТІ СТВОРЕННЯ ПРОГРАМИ ВИДІЛЕННЯ КОНТУРІВ РУХОМИХ ОБ'ЄКТІВ .....</b>	<b>79</b>
5.1 Проведення комерційного та технологічного аудиту науково-технічної розробки.....	79
5.2 Оцінювання рівня новизни розробки .....	84
5.3 Розрахунок витрат на проведення науково-дослідної роботи .....	88
5.4 Розрахунок економічної ефективності науково-технічної розробки при її можливій комерціалізації потенційним інвестором.....	100
<b>ВИСНОВКИ .....</b>	<b>106</b>
<b>ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....</b>	<b>108</b>
<b>ДОДАТОК А Технічне завдання .....</b>	<b>112</b>
<b>ДОДАТОК Б Лістинг програми .....</b>	<b>116</b>
<b>ДОДАТОК В Послідовність виділення рухомих об'єктів .....</b>	<b>120</b>
<b>ДОДАТОК Г Методи виділення контурів об'єктів.....</b>	<b>121</b>
<b>ДОДАТОК Д Методи виділення руху.....</b>	<b>122</b>
<b>ДОДАТОК Е Схема системи комп'ютерного відеоспостереження .....</b>	<b>123</b>
<b>ДОДАТОК Ж Структурна схема програми виділення контурів об'єктів..</b>	<b>124</b>
<b>ДОДАТОК И Протокол перевірки кваліфікаційної роботи на наявність текстових запозичень .....</b>	<b>125</b>

					<i>08-23.МКР.003.00.000 ПЗ</i>	<i>Арк.</i>
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		7

## ВСТУП

Побудова систем пошуку, виділення й розпізнавання об'єктів із приміненням засобів комп'ютерного зору є актуальною задачею теперішнього стану розвитку інформаційних технологій. Вирішення задач у цій області інформаційних технологій знаходять широке застосування в самих різних сферах життя сучасної людини: відстеження появи автомобіля на парковці, пошук забутого пасажиром багажу в приміщеннях вокзалів й аеропортів, забезпечення захисту від несанкціонованого проникнення зловмисника на об'єкти, що охороняються, розпізнавання автомобільних номерів, різних написів, особи за зображенням обличчя та інше. Упровадження методів комп'ютерного зору у виконання цих операцій дозволяє автоматизувати процеси, що пов'язані із обробкою такої візуальної інформації.

Пошук й виявлення об'єктів, що переміщуються в просторі, є важливою задачею роботи засобів комп'ютерного зору. Цей процес складається із етапів отримання відео зображення, виконання його попередньої обробки, аналізу й подальшого розпізнавання виділених об'єктів у відео зображеннях, використовуючи статистичні методи та різні моделі, що сформовані при допомозі геометричних, фізичних й статистичних особливостей об'єктів [1,2].

Ключовим елементом сучасних систем комп'ютерного зору при розв'язанні ряду прикладних задач, що пов'язані із розпізнаванням об'єктів, є виділення контурів об'єктів зображень [3]. При аналізі відео зображень і розпізнаванні об'єктів, що присутні у ньому, вагому частку процесу обробки зображення приймають на себе методи й алгоритми виділення контурів об'єктів, так як вони значно спрощують роботу із зображенням та сформованим цифровим рядом. У ідеальному випадку результатом застосування до відео зображення детектора контурів об'єктів може бути отриманий набір з'єднаних кривих, що позначають межі аналізованих об'єктів, межі забарвлення їх поверхонь, а також усі інші криві, що відповідають розривам у орієнтації поверхонь об'єктів [4]. Використання алгоритму виділення контурів об'єктів у



зображенні може значно зменшити обсяг даних, що підлягають обробці, відфільтровуючи ту інформацію, яка може тепер розглядатися як менш вагома, але зберігати важливі структурні властивості початкового зображення.

Однак не завжди можливо виділити контури об'єктів у картинах реального світу середньої складності. Межі об'єктів, виділених у таких зображеннях, часто мають такі суттєві недоліки, як фрагментарність (криві контурів об'єктів не з'єднані між собою), відсутність меж або ж наявність помилкових контурів, які не відповідають фактичному об'єкту. Тому задача подальшого розроблення й вдосконалення методів пошуку й виділення контурів рухомих об'єктів у відео зображеннях, що є кращими за такими критеріями, як точність та достовірність, на теперішній час є **актуальною**.

Магістерська робота виконана по кафедральній науково-дослідній тематиці студентського наукового гуртка кафедри ОТ «Розробка комп'ютерної системи пошуку та розпізнавання обличчя людини», керівником якого є старший викладач кафедри Очкуров М. А.

**Метою дослідження** є вдосконалення методів пошуку й виділення контурів рухомих об'єктів у відео зображеннях із використанням знаходження області руху та виділення рухомих об'єктів.

**Задачі дослідження:**

— здійснити аналіз відомих методів обробки відео зображень для виділення області руху та знаходження контурів рухомих об'єктів;

— запропонувати поліпшений метод оброблення відео зображень для виділення контурів рухомих об'єктів;

— створити алгоритм й розробити програму обробки відео зображення для виділення контурів рухомих об'єктів;

— здійснити обґрунтування доцільності здійснення нового наукового рішення, виконати розрахунок економічних затрат для створення нових програмних засобів виділення контурів рухомих об'єктів відео зображень та оцінити економічні переваги від упровадження нового розробленого програмного продукту.

**Об’єкт дослідження** — процес обробки даних про об’єкти відео зображення шляхом пошуку й виділення області руху в аналізованій сцені спостереження для виділення контурів рухомих об’єктів.

**Предмет дослідження** — методи обробки відео зображень для пошуку й виділення контурів рухомих об’єктів.

**Методи дослідження:** використовувались методи дискретної математики для обробки матриць, що описують кадри відео зображення, методи диференційного числення для знаходження екстремумів функції, методи теорії множин для опису процесу виділення контурів рухомих об’єктів. Використано засади об’єктно-орієнтованого програмування для реалізації запропонованого підходу виділенню контурів рухомих об’єктів у відео зображеннях.

**Наукова новизна одержаних результатів** полягає у тому, що удосконалено метод виділення контурів рухомих об’єктів, який відрізняється від існуючих виконанням пошуку й виділення рухомих об’єктів у вибраній сцені відео зображення із використанням переведення зображення у бінарне та виділення контуру шляхом послідовного аналізу кожної із трьох суміжних точок контуру у вигляді прямокутного трикутника, що дозволяє із більшою точністю проводити виділення контурів рухомих об’єктів відео зображень.

**Практичне значення одержаних результатів:**

— створено алгоритм обробки відео зображень для виділення контурів рухомих об’єктів;

— розроблено програму обробки відео зображень для виділення контурів рухомих об’єктів.

**Апробація** результатів магістерської роботи зроблено доповідь на Всеукраїнській науково-практичній інтернет-конференції «Молодь в науці: дослідження, проблеми, перспективи (МН-2022)» (Вінниця, 2022 р.).

За результати магістерської роботи **опубліковано** 1 наукову працю [5].

# 1 АНАЛІЗ МЕТОДІВ ТА ЗАСОБІВ ВИДІЛЕННЯ РУХОМИХ ОБ'ЄКТІВ

Пошук та виділення рухомих об'єктів у вибраному зображенні сцени спостереження заснований на знаходженні окремих груп пікселів, які відрізняються вибраними параметрами в отриманих сусідніх відеокадрах. Цей розділ магістерської роботи присвячений огляду й аналізу існуючих методів пошуку рухомих об'єктів у зоні відеоспостереження та виділення їх контурів.

## 1.1 Комп'ютерний зір як засіб виділення об'єктів

Комп'ютерний зір — це сукупність наукових та інженерних компонентів, які використовуються для створення комп'ютерних систем виявлення, класифікації та відстеження об'єктів. Комп'ютерний зір відноситься до теорії та технології побудови штучних систем, які сприймають дані як зображення, які можуть бути представлені у вигляді відео послідовностей, зображень з різних камер спостереження або веб-камер. Одним з основних застосувань комп'ютерного зору є машинне зір.

Сферою застосування машинного зору як області комп'ютерного зору є цифрові пристрої введення/виведення даних для управління виробничим процесом і моніторингу за допомогою камер відеоспостереження. Системи обробки зображень використовують камери для візуального спостереження та відповідне програмне забезпечення для обробки зображень. Виявлення та підрахунок об'єктів, виявлення переміщення об'єктів у просторі, зчитування серійних номерів або виявлення дефектів деталей. Основною перевагою систем обробки зображень є можливість безперервної та високої швидкості та необхідна точність вимірювань.

Машинний зір — це процес використання комп'ютерного зору в промисловому виробництві, що складається з часто використовуваних апаратних і програмних компонентів. Типова побудова системи машинного зору складається із таких компонентів [1, 6]:

- одна або кілька цифрових або аналогових відеокамер (кольорових або чорно-білих) з відповідним оптичним обладнанням для захоплення зображення;

- програмне забезпечення для обробки отриманих зображень (із аналоговими камерами це також завдання оцифрувати зображення);

- процесор — сучасний персональний комп'ютер з багатоядерним або вбудованим процесором;

- спеціальне програмне забезпечення для обробки зображень, яке надає засоби для створення окремих додатків програмних продуктів;

- датчик для синхронізації захоплення, захоплення та обробки зображення (частіше всього застосовують оптичний або магнітний).

У результаті отримуємо комп'ютерний зір як частину комп'ютерного зору як частину комп'ютерної системи. Комп'ютерна система — це інформаційно-технічний комплекс, призначення якого — обробка, зберігання та введення та виведення інформації. Комп'ютерна система включає комп'ютери, пристрої введення/виводу, сервери із відповідним програмним забезпеченням. Комп'ютерні системи можуть бути інтегровані у комп'ютерну мережу. Обмін інформацією здійснюється через локальну або ж глобальну систему передачі даних через комп'ютерну мережу. Для опису таких систем використовуються технічні, документальні, організаційні, алгоритмічні, функціональні, програмні і інформаційні структури. Комп'ютерну систему можна інтегрувати з іншими технічними технологіями для розширення можливостей та створення єдиного середовища для управління різними процесами завдяки різноманітності та значній уніфікації комп'ютерного обладнання.

Комп'ютерна система складається з наступних компонентів:

- апаратне забезпечення комп'ютера, основною частиною якого є центральний процесор (CPU), що виконує інструкції комп'ютера;

— пам'ять комп'ютера, що зберігає програми та дані, а також пристрої введення-виведення або так звані, сприйняті користувачем — людиною чи іншими програмами;

— операційна система (Operating System) — основний інструмент управління та системного програмного забезпечення, який розподіляє контроль за використанням комп'ютерного обладнання користувачами та різними програмами;

— прикладне програмне забезпечення (додатки) — програми, що призначені для вирішення різноманітних типів задач.

До них належать компілятори, графічні бібліотеки, системи управління базами даних, ігрові програми, різні офісні програми. Прикладне програмне забезпечення створює наступний більш високий рівень порівняно із операційною системою та дозволяє вирішувати різні проблеми із додатками на комп'ютері. Завдання полягає в пошуку та виявленню руху об'єктів у відео послідовності, що й буде одним із наступних прикладних продуктів. Створення такого продукту є одним із завдань створення системи розпізнавання зображень.

Системи розпізнавання зображень, що використовують методи і засоби пошуку об'єктів, поділяються на детерміновані, ймовірнісні, логічні, нейронні, структурні та експертні системи. Відповідно до посилання на навчання, системи розпізнавання образів можна розділити на такі типи, як системи, що не навчаються, системи навчання та системи самонавчання [7, 8].

Ненавчальні системи, які мають достатньо інформації для побудови апріорного словника й алфавіту ознак класу та побудови опису всіх класів мовою цих ознак. Тому достатньо спочатку наблизити відстані або скінченні межі між цими класами ознак і встановити вирішальні правила, які дозволяють розділити сформований набір ознак на відповідні класи. Для того, щоб побудувати таку систему розпізнавання, необхідно мати повну, попередню, апріорну інформацію про об'єкти та системи навчання. У таких системах розпізнавання початкових інформаційних шаблонів достатньо лише для

побудови алфавіту апріорних класів і апріорного словника ознак, але недостатньо для опису класів мовою ознак. Доступна початкова інформація дозволяє класифікувати конкретні об'єкти за різними класами. «Учитель» неодноразово представляє систему розпізнавання педагогічними образами відомого класу. Якщо така система розпізнавання зображень дає неправильну відповідь, система змінює всі підсилення вибраних сигналів за відомим алгоритмом, щоб отримати правильну відповідь. Після таких випробувань і навчання система виявлення починає працювати самостійно — це системи самонавчання.

Попередньої апріорної інформації в цих системах достатньо лише для створення словника ознак, але не для будь-якої іншої класифікації даних об'єктів. На початковому етапі формування система представляє навчальний набір об'єктів, яким присвоюються значення їх характеристик. Система не може приймати інструкції, оскільки немає такої початкової інформації про те, до якого класу належать об'єкти цього початкового набору. На етапі самонавчання інструкції замінюються певним набором правил, а потім сама система розпізнавання образів починає створювати класифікацію, якої ми будемо дотримуватися в майбутньому. Метою такого самонавчання є створення певної інформації, якої було б достатньо для повноцінного функціонування системи розпізнавання. Для досягнення такого результату необхідний зворотний зв'язок для внесення виправлень і уточнення інформації, яка обробляється апріорі.

## 1.2 Аналіз методів виділення рухомих об'єктів

Існує багато різних типів алгоритмів вибору рухомих об'єктів. Здатність цих алгоритмів працювати і давати правильний результат залежить від різних значень вхідних параметрів. У зв'язку з такою постановкою задачі виникає потреба використовувати більше одного алгоритму для вирішення цієї проблеми. Тому завдання на сучасному етапі створення інструментів вибору рухомих об'єктів полягає в тому, щоб серед багатьох існуючих знайти одні з найбільш ефективних і правильних алгоритмів, запропонувати нові підходи та

досягти бажаного результату шляхом комбінування кількох алгоритмів. Існують основні підходи до пошуку та виділення рухомих об'єктів у кадрі, такі як визначення переміщення ділянок пікселів між сусідніми кадрами та визначення зміщення прямокутних блоків у зображенні [10, 11].

Зазначення переміщення ділянок пікселів між сусідніми кадрами. У цьому випадку виділення рухомих об'єктів ґрунтується на сегментації поточного та попереднього кадрів, пошуку відповідних ділянок на зображенні та їх оцінці шляхом зміни їх положення чи форми. Цей підхід вирішує проблему відстеження рухомих об'єктів та ґрунтується на апріорному припущенні, що рухомий об'єкт має однорідний колір або текстуру і достатньо контрастує із фоном зображення. Основними недоліками цього підходу є відносно низька стабільність в умовах низької контрастності рухомих об'єктів на стаціонарному фоні та високі обчислювальні витрати при виконанні операцій сегментації зображення.

Визначення переміщень прямокутних блоків на зображенні [12]. Пошук і виділення рухомих об'єктів у кадрі полягає у знаходженні векторів зсуву виділених прямокутних блоків для пари кадрів. Передбачається, що блоки вектора зсуву ненульової довжини містять рухомі об'єкти. Недоліками цього підходу є нестабільність результату для об'єктів з малим відношенням поверхні до периметра та для монохромних об'єктів за наявності перешкод.

Для виконання завдання управління зоною з невеликим трафіком і відносно стабільними умовами відео ефективно використовувати FPS, оскільки цей підхід вимагає низьких обчислювальних витрат, не має обмежень на кількість об'єктів, керованих одночасно, і відносно низьку чутливість до різниці. Має інструменти, маскування рухомих об'єктів.

Існує два основних підходи до визначення попиксельних змін у кадрі: знаходження прямої зміни яскравості пікселя в поточному та попередніх проаналізованих кадрах і знаходження зміни яскравості між поточним кадром і створеним фоном.

Докази прямої зміни піксельної чіткості нормальних та попередньо проаналізованих клітин, для яких записується значення програми, що окреслює сприйнятливість цього методу до міжклітинних відмінностей.

Основною перевагою цього методу є неможливість вільного виявлення деструктивних об'єктів із проекцією менше одного пікселя на відстань між двома сусідніми осередками. Для вирішення цієї проблеми використовується аналіз кількох попередніх клітинок. І те ж саме стосується додаткових витрат пам'яті на обробку даних та часу процесора на аналіз таких додаткових комірок, а також ще й появи у прапорцях прапорців обраного «плавного» переміщення такого об'єкта.

Знайдіть зміну прозорості між кліткою бочки та формованою підлогою. Після виконання операції генерування матриці різниці прапорів фон форматується на кожному кроці шляху. Якщо  $AF = 1$  або  $DF = 255$ , процес заснований на малюванні прямої зміни яскравості пікселя між загальним пікселем і попереднім пікселем.

Однак ця подія займає багато часу, щоб перетворити потрібний фон, випадково застосувавши поле зору камери до об'єкта, який охоплює найдовшу частину кадру (наприклад, обертається або поблизу від висоти пташиного польоту). Ситуація вирішується операцією примусового переформатування підкладки, коли кількість суттєвих відмінностей закінчується значенням заданого значення програми MDF. Зауважте, що замість піксельної різкості ви можете використовувати будь-який інший обмежений стандарт кольору обличчя або уточнити рух для кожного із паралельних кольорів RGB за допомогою логічного підсумовування [13].

Коли ви переміщуєте об'єкт у фільмі — це об'єкт, який змінює положення в кожному кутку фільму відносно інших об'єктів у сцені. Він рухається у плівці, це частина хаотичної плівки, в якій рухається об'єкт. Відомі методи класифікації об'єктів, що порушують відеопослідовність, засновані на методах виявлення об'єктів у цифровому зображенні, але це відео може бути сукупністю послідовних кадрів відеопослідовності на дисплеї. Однією із



головних вимог до таких помилкових методів вибору об'єктів є слабкість, але класифікація об'єктів має бути виконана. Основною вимогою для таких операцій є придбання необхідної мережі. Простота впровадження означає низьку складність обраного методу по відношенню до реалізації в обраному програмному забезпеченні.

У сучасних системах відеоспостереження, що використовують алгоритми вибору, виявлення та аналізу об'єктів, важливим моментом є операція правильного виділення рухомого об'єкта [14]. Якщо така система експлуатується в реальних умовах, то існуючі методи пошуку та вибору об'єктів повинні враховувати наявність перешкод, а також враховувати зміни рівня цих перешкод у просторі та часі; можливе часте пристосування до зміни фону або умов освітлення, що, швидше за все, впливає на кінцевий ефект роботи, який дуже часто виявляється неправильним. Тому необхідно також вирішити проблему дослідження відомих методів пошуку та виявлення рухомих об'єктів із метою їх подальшого вдосконалення та розробки нових підходів і методів.

Вибір рухомих об'єктів складається з кроку пошуку та виділення всіх пікселів переднього плану та кроку подальшого об'єднання вибраних пікселів в окремі об'єкти. Вирішено такі основні завдання пошуку та виявлення рухомих об'єктів:

- пошук і розпізнавання об'єкта в кадрі;
- визначення параметрів об'єкта за його властивостями;
- розрахунок напрямку та швидкості руху об'єкта;
- обчислення відстані спостерігача від обраного зображення вибраного об'єкта для виявлення параметрів об'єкта;
- висловлювати висновки щодо доцільності подальшої підтримки наявними коштами.

Якщо хочете записати відео рухомого об'єкта, необхідно переконатися, що швидкість та напрямок руху об'єкта відповідають динамічним параметрам вашої існуючої системи відеоспостереження.

При проектуванні систем переміщення та пошуку рухомих об'єктів слід враховувати обмежену швидкість камери та можливі обмеження об'єктів. Це дослідження важливо для розробки засобів відеоспостереження, як стаціонарних (контроль руху, системи безпеки тощо), так й мобільних, наприклад, для обробки відео із безпілотних літаючих апаратів.

Щоб визначити напрямок руху об'єкта, один кадр порівнюється із умовною системою координат, а два кадри відео потоку порівнюються для визначення областей, що відрізняються один від одного. Такий підхід відображає той факт, що об'єкт з'являється в кадрі. Певні критерії, такі як форма або ж колір, визначають, чи належить об'єкт, який слід відстежувати, до певного класу. Вектор напряму об'єкта визначається шляхом порівняння кадрів. Знаходження швидкості об'єкта здійснюється шляхом визначення часу між моментами фіксації двох кадрів та визначення співвідношення між двома положеннями вибраного об'єкта в даний момент часу.

Загальний алгоритм методу знаходження міжкадрової різниці для двох кадрів такий:

- в аналіз вводяться два відеокадри з відеопотоку;
- аналізуються дві послідовності байтів у форматі RGB;
- різниці зображень розраховуються піксель за пікселем;
- для кожного пікселя розраховується середнє значення;
- отримані значення порівнюються із встановленим порогом.

В результаті отримуємо двійкову маску, в якій одному елементу відповідають три компоненти кольору пікселя двох вихідних зображень [15]. У масці блок розміщують там, де може відбуватися рух, але на цьому етапі формування маски може виникнути неправильна обробка окремих частин маски, які неправильно вставлені в блок.

Як правило, послідовні кадри з відео потоку використовуються як вхідні кадри, й можна використовувати кадри зі значним інтервалом, наприклад, використовувати від одного до трьох кадрів. Цей підхід відносно простий у реалізації. Основним недоліком цього підходу є постійні неполадки обладнання

під час запису даних. Для точного визначення місця розташування та розпізнавання рухомих об'єктів буде розглянуто метод розрізнення поточного кадру та базового кадру, який є фоновим кадром. На якість виявлення руху впливає форма базової рамки, яка повинна мати певні властивості:

- мінімальний рівень шуму;
- якщо аналізований кадр є реальним кадром з відеопотоку, він повинен хоча б трохи відставати від поточного кадру;
- якщо базовий кадр створений штучно, він повинен мати мінімальну кількість рухомих елементів, інакше помилкові тривоги неминучі для тих об'єктів, яких більше немає в поточній клітці, але базова рамка містить деякі їх елементи.
- потрібно відфільтрувати перед процесом оновлення кадру, а при створенні фонової рамки головне завдання — обрізати ділянки руху.

Перевага цього методу в тому, що він менш чутливий до перешкод. Існує різниця між методами вибору між кадрами та фоном:

- аналізується кілька десятків кадрів для отримання фонової оцінки;
- у попередньому кадрі об'єкти розміщені майже так само, як і в наступному.

Для виконання операції оцінки фону потрібно обробити більше кадрів.

### 1.3 Аналіз методів виділення контурів об'єктів цифрових зображень

Межі об'єкта (вибір кордонів) — це термін в обробці зображень і теорії комп'ютерного зору, частково в області пошуку та вибору об'єктів на основі алгоритмів, які виділяють точки на цифровому зображенні, де є різкі зміни яскравості або інші види неоднорідності.

Основною метою виявлення різких змін яскравості зображення є захоплення об'єктів зображення. Вони можуть відображати різні припущення щодо моделі зображення, зміни яскравості зображення можуть свідчити [9, 15]

— зміна глибини, орієнтації поверхні, властивостей матеріалу, різниця в освітленні сцени.

В ідеалі вибір «Межі об'єктів» — це набір пов'язаних кривих, які вказують на межі об'єктів, областей і графіків на поверхні, а також кривих, що відображають зміни положення поверхні. Таким чином, застосування фільтра вибору кордонів до зображення може значно зменшити обсяг оброблених даних, оскільки відфільтрована частина зображення вважається менш критичною, а найважливіші структурні властивості зображення зберігаються. На жаль, при реальних зображеннях середньої складності не завжди можна виділити межі зображення. Межі, виділені із таких зображень, часто мають такі дефекти, як фрагментація (граничні криві не пов'язані), відсутність кордонів або ж наявність дефектних меж, які не відповідають об'єкту тестування.

Межі, що виділені на 2D-зображенні 3D-сцени, можна розділити на залежні або незалежні точки зору. Ті, які не залежать від меж, зазвичай відображають властивості, успадковані від тривимірних об'єктів сцени, такі як колір та форма поверхні. Межі точки огляду можуть змінюватися в міру зміни точки огляду, відображаючи геометрію сцени.

Границі дуже важливі в багатьох областях обробки зображень, особливо у системах обробки зображень, які аналізують сцени штучних об'єктів при постійному освітленні. Тим не менш, у деяких публікаціях розглядається можливість призначення ідеальних меж, хоча межі на основному зображенні не завжди однакові. На них завжди впливає один або кілька різних ефектів [3, 16] — напівтонове розмиття від неточкових джерел світла, розмиття фокусу через остаточну глибину різкості, затінення від гладких об'єктів, тому багато дослідників використовують згладжене по Гаусу межі кроку. як найпростіше наближення ідеальної реберної моделі для визначення нечітких меж у прикладних задачах.

Вибір кордонів є незвичайним завданням, оскільки зміна яскравості зображення верхніх сіл не завжди швидко змінюється. Тому не завжди легко встановити певний поріг зміни яскравості між двома сусідніми пікселями, щоб

могли бачити, що існує межа. Це одна із причин, чому вибір меж є нестандартним завданням, коли об'єкти сцени складні, а умови освітлення налаштовані дуже погано.

Існує багато підходів до розрізнення кордонів (меж) об'єктів, але майже всі можна розділити на дві категорії — методи пошуку піків та нульових показників. Методи, що засновані на максимумах, розрізняють кордони шляхом обчислення «міцності краю», зазвичай спочатку похідних термінів, таких як величина градієнта, а потім знаходження локальних піків міцності краю за допомогою передбаченого напрямку кордону, зазвичай перпендикулярного до вектора градієнта. Методи нульового пошуку шукають точки перетину абсцис другого похідного члена, як правило, нулів Лапсаса або нулів нелінійного диференціального члена. Згладжування зображення майже завжди використовується як етап попередньої обробки, щоб підкреслити межі, зазвичай за допомогою фільтра Гаусса [9, 16].

Обговорювані граничні методи відрізняються застосованими фільтрами згладжування та методами, які дозволяють знаходити межу країв об'єктів. У той час як багато методів виділення кордонів покладаються на обчислення градієнта зображення, вони відрізняються за типом фільтрів, які використовуються для обробки градієнтів у  $x$  (горизонтальному) і  $y$  (вертикальному) напрямках.

Окрім того, можете використовувати різні оператори градієнта для оцінки величини градієнта зображення або ж його згладженої версії. Найпростішим методом можна назвати використання центральних відмінностей. На таких фільтрах заснований відомий оператор Собеля.

Інші типи операторів для обчислення градієнта зображення були запропоновані Превітт і Робертсом [9, 16].

Після того, як ми знайшли граничне відношення (зазвичай значення градієнта), наступним кроком є застосування кордону, щоб визначити, чи знаходиться межа в певній точці зображення. Чим нижчий поріг, тим він буде краще, але тим більш чутливим до шуму буде результат, підкреслюючи

додаткові дані зображення. Навпаки, високий поріг не може зберегти слабкі краї або межі фрагмента.

Коли до градієнтного зображення просто застосовується поріг, межі в результаті виходять товстими і вимагають подальшої обробки, що робить межу тоншою і точнішою. За визначенням, якщо визначаєте межі із немаксимальним загасанням, межі стають тоншими та їх можна об'єднати у багатокутники за допомогою методу змішування країв (відстеження меж). У дискретній сітці немаксимальна фаза придушення може бути виконана, оцінюючи напрямок градієнта за допомогою кількох похідних, де напрямок округлюється до числа із кроком  $45^\circ$  градусів.

Лінійні пороги — це поширений підхід до вирішення проблеми пошуку потрібного порогу. Принцип використовує кілька порогів. Працюємо із верхнім порогом, щоб дізнатися початкову точку кордону. Отримавши початкову точку, аналізуємо межу точково до тих пір, поки значення граничної сили не перевищить нижній поріг. Цей алгоритм передбачає, що межі є тільки кривими, та дозволяє нам відстежувати слабкі ділянки кордонів, не припускаючи, що всі інші точки на зображенні будуть позначені як ребра. Однак при цьому методі все ще маємо проблему з вибором конкретних порогових значень, оскільки основні параметри можуть змінюватися від зображення до зображення [3, 9].

Уточнення контуру — це процес створення тонкого контуру шляхом відсікання небажаних та помилкових точок, які з'являються на краю. Ця техніка використовується після того, як зображення було пом'якшено (за допомогою медіанної або ж гаусової фільтрації). Оператор відсікання був використаний для розрахунку міцності краю, а також було застосовані деякі пороги після видалення кліпів. Цей метод видаляє будь-які зайві точки і, якщо використовувати його обережно, розкриває контури товщиною в один піксель.

Оператор Собеля використовується в області обробки зображень. Він частіше використовується в методах розмежування. Фактично це дискретний диференціальний оператор, який обчислює приблизне значення яскравості градієнта зображення. Оператор Собеля застосовується до згортки зображення

за допомогою невеликих фільтрів цілих чисел у горизонтальному та вертикальному напрямках, тому обчислити його більш-менш легко. З іншого боку, градієнтна апроксимація, яку він використовує, досить груба, особливо для різких, тонких змін яскравості зображення [9, 17].

Простіше кажучи, оператор обчислює градієнт яскравості зображення у кожній точці. Отже, потрібно знайти напрямок найбільшого збільшення яскравості та величину її зміни в цьому напрямку. Ефект полягає в тому, як «різко» або «м'яко» змінюється яскравість зображення в кожній точці, а разом із цим і можливість знаходити точку на межі, а також розташування межі. На практиці обчислення величини зміни яскравості є більш надійним і легшим для інтерпретації, ніж обчислення напрямку.

З математичної точки зору градієнт функцій двох змінних для кожної точки зображення є вектором, вертикальна та горизонтальна складові якого є похідними від яскравості зображення. Вектор градієнта вказує на максимальне збільшення яскравості в будь-якій точці зображення, а його довжина дорівнює величині зміни яскравості. Таким чином, результатом оператора Собела в точці області постійної яскравості є нульовий вектор, а в точці, що лежить на межі областей різної яскравості, вектор, що перетинає пряму в напрямку збільшення яскравості [9, 18].

Строго кажучи, оператор використовує ядра  $3 \times 3$ , які згортають вихідне зображення, щоб обчислити приблизні значення похідних по вертикалі і горизонталі. Початкове зображення  $A$  послідовно обробляється через маски  $G_x$  й  $G_y$ , а результатом є матриця з двох нових зображень, кожна точка має наближені похідні за координатами  $x$  і  $y$ . Вони обчислюються наступним чином, у якому знак  $*$  позначає метод двовимірної згортки:

$$G_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} * A,$$

та для обчислення приросту функції по координаті  $y$  використовується вираз

$$G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * A.$$

Тут координата  $x$  рухається «праворуч», а координата  $y$  — «вниз». У будь-якій точці зображення ви можете обчислити найближче значення градієнта, використовуючи приблизні значення похідних, які ви отримуєте:

$$g_{i,j} = |f_{i,j} - f_{i+1,j+1}| + |f_{i,j+1} - f_{i+1,j}|,$$

$$G = \sqrt{(G_x^2 + G_y^2)},$$

Застосовуючи дану інформацію, можемо також ще й розрахувати напрямок градієнта:

$$\Theta = \arctg\left(\frac{G_y}{G_x}\right),$$

у якому кут  $\Theta$  рівняється нулю для вертикальної межі, а от темна сторона знаходиться вже ліворуч.

Оператор Собеля є неточним наближенням градієнта зображення, але досить високим для практичного використання в багатьох задачах. Точніше, оператор використовує тільки значення інтенсивності близько  $3 \times 3$  кожного пікселя, щоб наблизити відповідний градієнт зображення, і використовує лише цілі значення коефіцієнтів яскравості [4, 9] для оцінки градієнта.

За допомогою оператора Собеля створюється двовимірна карта градієнта кожної з її точок. Його можна розрахувати та представити у вигляді зображення, на якому частини з крутими схилами (перепадами) показані білими лініями [3, 9].

Оператор Превітт — це метод визначення меж у обробці зображень, який обчислює максимальну відповідь на набір ядер згортки для отримання локальної орієнтації кордону в кожному пікселі. Його зробила Dr. Джудіт Превітт, лікар за професією, для визначення меж медичних зображень [8].



Для цієї операції використовуються різні ядра. Можна отримати вісімку з ядра, перетворивши коефіцієнти в коло. Кожен результат чутливий до напрямку обмеження від  $0^\circ$  до  $315^\circ$  із кроком  $45^\circ$ , при цьому  $0^\circ$  відповідає вертикальному обмеженню. Максимальна реакція кожного пікселя — це значення відповідного пікселя в вихідному зображенні. Його значення коливаються у діапазоні від 1 до 8 залежно від кількості ядер, які дали найкращий результат.

Цей метод визначення меж також називають узгодженням з шаблоном краю, оскільки зображення порівнюється із серією шаблонів, кожен з яких представляє орієнтацію кордону. Розмір та орієнтація краю в пікселі потім визначаються шаблоном, який найкраще відповідає локальній околиці аналізованого пікселя [3, 9, 17].

Детектор кордонів Prewitt — це чудовий спосіб оцінити розмір і орієнтацію вашої межі. В даний час, оскільки детектор диференціального градієнта вимагає орієнтації в горизонтальному і вертикальному напрямках, що важко обчислити, детектор межі Прюїтта відстежує курс безпосередньо від ядра з максимальними результатами. Набір ядер обмежений 8 можливими напрямками, але, як зазвичай, виявляється, що більшість оцінок орієнтації не дуже точні [4, 9].

З іншого боку, набір ядра вимагає 8 напрямків для аналізу кожного пікселя, тоді як ядро методу градієнта вимагає лише 2 — горизонтальні та вертикальні. Рішення, що представляє силу меж, дуже схоже в обох принципах, якщо використовуються одні й ті самі ядра згортки.

Використання оператора Prewitt дає кращі результати, коли необхідно визначити межу на зображенні з домінуючим напрямком орієнтації. Розглянуті оператори використовуються при створенні програм для підкреслення контурів комп'ютерних зображень. Не всі оператори градієнта використовують другу похідну яскравості зображення для роботи з градієнтом. Звичайно, ідеально, щоб наявність нулів другої похідної дозволяла знаходити максимуми локального градієнта.

Іншим підходом є знаходження границь об'єктів зображення, що представлено як бінарне зображення. Для виділення границь об'єктів у такому представленні існують скануючі та відстежуючі алгоритми [9]. Відстежуючі алгоритми засновані на тому, що на зображенні відшукується об'єкт (перша точка об'єкту, що зустрілася) і контур об'єкту відстежується і векторизується. Перевагою цих алгоритмів є їх простота, до недоліків можна віднести їх послідовну реалізацію і деяку складність при пошуку і обробці внутрішніх контурів.

Жук розпочинає рух з білої області у напрямку до чорної, як тільки він потрапляє на чорний елемент, він повертає ліворуч і переходить до наступного елемента. Якщо цей елемент білий, то жук обертається направо, інакше - наліво. Процедура повторюється до тих пір, поки жук не повернеться у вихідну точку. Координати точок переходу з чорного на біле і з білого на чорне і описують межу об'єкту.

Скануючі алгоритми засновані на перегляді (скануванні) усього зображення і виділення контурних точок без відстежування контуру об'єкту. Для обробки інформації в кожному рядку матриці, що описує зображення, послідовно переглядають рядок та фіксують момент переходу із чорної точки на білу або навпаки. Для кращого аналізу одночасно зберігаються два рядки зображення (поточна і попередня). Аналізуються X координати чорних серій обох рядків в порядку їх зростання (зліва направо) і виявляються ситуації, які можуть виникнути. Підходи, що реалізовані у цих алгоритмах, використаємо для виділення контурів знайдених об'єктів.

Виділення меж об'єкта зображення не є остаточним підсумком у комп'ютерних системах для визначення та аналізу інформації. Стрімкий розвиток технологій цифрової обробки зображень призвів до появи нових методів і підходів до вирішення різноманітних завдань визначення меж об'єктів зображення та їх застосування при розв'язанні багатьох практичних завдань. В даний час існує ряд алгоритмів визначення та розмежування меж цифрових об'єктів зображення. Залежно від складності зображення, а також від його

використання, підходи до розмежування об'єктів бувають різними. Вибір алгоритму вибору меж об'єкта залежить від умов, пов'язаних зі швидкістю роботи та необхідністю втручання користувача в процес визначення меж об'єкта. У нашому випадку найкращими методами є підходи до визначення меж об'єкта, засновані на визначенні країв об'єкта, представленого у вигляді бінарного зображення, що розробляється як програмне забезпечення.

#### 1.4 Етапи формування пошуку об'єктів

Відомі стандартні методи пошуку та виділення об'єктів — підсвічування яскравістю; вибір на основі фактур, вибір кольору або ручний вибір. Найпоширеніший і найпростіший метод виділення об'єктів, який часто використовується — виділення об'єктів за яскравістю. У цьому методі позначання об'єктів або окремих фаз кадрів здійснюється шляхом виділення тих точок зображення, які потрапляють у певний діапазон значень яскравості. Цей вибір можна зробити на гістограмі яскравості для цього цифрового зображення. Цифрове зображення — це набір точок, яскравість яких у 8-бітовому представленні зображення коливається від значення 0 (відповідає повністю чорному) до значення 255 (відповідає повністю білому). Гістограма яскравості зображення показує відсоток точок при певному значенні яскравості. Вибір об'єктів за значеннями яскравості зазвичай здійснюється шляхом встановлення меж яскравості, і всі ті точки зображення, яскравість яких знаходиться в межах заданих значень, будуть виділені на зображенні як маски об'єкта, які потім будуть оброблені.

Пошук і виділення об'єктів за кольором здійснюється за допомогою колірних моделей, процес складніший, ніж простий вибір за яскравістю, але дає кращі результати, коли об'єкти з однаковими характеристиками яскравості відрізняються за кольором. У сучасних системах аналізу зображень відбір кольорових об'єктів за кольором здійснюється за допомогою методу навчання, при якому користувач системи визначає ділянки цифрового зображення, які необхідно виділити. Підготовлена таким чином система на наступному

цифровому зображенні автоматично виділить ті об'єкти, які за кольором відповідають певним ділянкам. Колір містить дуже важливу інформацію про схожість об'єктів, хоча зазвичай важко приймати рішення на основі лише кольору. Зображення об'єктів реального світу містять багато різних відтінків кольору, багато з яких дуже близькі та схожі, немає сенсу аналізувати всі доступні відтінки кольорів, оскільки пікселі, що належать навіть одному об'єкту, часто відрізняються за кольором, хоча це невидимо до людського ока.

Пошук і відбір об'єктів за властивостями їх текстур найбільш близький до механізмів ідентифікації об'єктів, які використовує мозок людини при їх розпізнаванні. Цей підхід ґрунтується на розпізнаванні об'єкта або фази сцени не за яскравістю чи характеристиками кольору, а за деякими зовнішніми характеристиками чи характеристиками текстури. Це стосується ситуацій, коли в межах однорідної структури об'єкта необхідно виділити шар або ділянку з орієнтованою структурою, яка не відрізняється від аналізованого шару не яскравістю чи кольором, а лише зовнішнім виглядом. Цей метод вибору об'єктів є найскладнішим у реалізації, і не всі існуючі цифрові аналізатори зображень підходять для цього типу задач вибору текстури.

Ручний метод пошуку та виділення об'єктів або окремих фаз сцени використовується в тих випадках, коли об'єкти не можуть бути виділені за допомогою жодного з перерахованих вище підходів. Використовуючи метод ручного вибору, користувач сам вказує системі, що слід вважати об'єктом.

Щоб отримати кількісну інформацію про об'єкти зображення за допомогою аналізатора, необхідною умовою є виділення об'єктів на фоні або сторонніх елементах. Вибір і подальше розпізнавання об'єктів зображення є важливими етапами обробки зображень. Тільки після вибору цікавих для нас об'єктів слід розглядати наступні кроки їх обробки.

### 1.5 Нейромережа глибокого навчання

Процес вибору та виявлення рухомих об'єктів у відеопослідовності можна виконати за допомогою нейронної мережі глибокого навчання. Операції

глибокого навчання належать до більш загального сімейства методів машинного навчання, які покладаються на подання навчальних даних, а не на спеціальні методи та алгоритми. Машинне навчання може бути неконтрольованим, частково контрольованим або контрольованим [7].

Машинні методи глибокого навчання характеризуються як клас алгоритмів:

- форми в шарах процесу навчання на кількох рівнях представлення, що відповідають різним рівням абстракції;
- шари створюють певну ієрархію понять;
- використовує багатозначну систему нелінійних фільтрів для виділення трансформованих ознак;
- є частиною більш широкого поля машинного навчання для результатів дослідження та представлених даних;
- має кілька рівнів для перегляду або визначення даних (навчання без вчителя) з функціями, організованими ієрархічно, що визначають вищий рівень виробництва.

Кожен наступний рівень системи отримує вхідні дані із попереднього рівня. Система глибокого навчання може працювати з алгоритмами навчання як із учителем, так й без нього, тому аналіз показує навчання без вчителя, а класифікація — навчання з викладачем. Алгоритми глибокого навчання відрізняються від алгоритмів неглибокого навчання кількістю параметризованих перетворень, які стабілізують сигнал, що поширюється від вхідного рівня до вихідного, розглядаючи параметризовані перетворення як блок обробки даних, який має такі параметри навчання, як ваги або пороги.

Згортка нейронна мережа (CNN) — це спеціальна архітектура штучної нейронної мережі, налаштована на ефективне розпізнавання зображень і є частиною інструментів і технологій глибокого навчання [19, 20].

Основою для побудови згорткових нейронних мереж є чергування згорткових шарів та шарів субдискретизації (шарів субдискретизації або об'єднання шарів, шар підвибірки). Структура складної нейронної мережі

односпрямована (без зворотного зв'язку) і по суті багат шарова. Для навчання мережі використовуються відомі стандартні методи, в основному це метод поширення помилки.

Робота згорткової нейронної мережі заснована на інтерпретації переходу від специфічних ознак зображення до більш загальних і абстрактних деталей, а на наступних етапах до ще більш абстрактних деталей для створення і відбору концептів верхнього рівня. У цьому випадку нейронна мережа сама вирівнюється і створює найнеобхіднішу ієрархію абстрактних ознак, відкидаючи дрібні деталі та підкреслюючи основне.

У нормальному перцептроні, який є повністю пов'язаною нейронною мережею, кожен наступний нейрон пов'язаний з усіма нейронами попереднього шару, і кожне з цих зв'язків має свій особистий коефіцієнт ваги. У згортковій нейронній мережі для виконання операції згортки використовується лише кінцева матриця малих ваг, яка переміщується по обробленому шару (на самому початку — прямо на вхідному зображенні), генеруючи відповідний сигнал активації для кожного нейрона, наступний шар після кожного шару. Тому одна і та ж матриця ваг, яку також називають ядром згортки, використовується для різних нейронів вихідного шару мережі. Ця матриця інтерпретується як графічне кодування певної ознаки, наприклад, наявність лінії під певним кутом. Потім у наступному шарі, отриманому в результаті переплетення такої вагової матриці, він вказує на наявність цього об'єкта в обробленому шарі та його координати, створюючи так звану карту ознак. У згортковій нейронній мережі такий набір ваг являє собою не одну, а цілий набір ваг, які кодують елементи зображення (дуги та лінії під різними кутами). Такі ядра згортки не вказуються дослідником, а формуються самостійно в решітці шляхом навчання відомого класичного методу зворотного поширення помилки. Кожна ітерація цього набору вагових коефіцієнтів створює власний екземпляр карт ознак, перетворюючи нейронну мережу в багатоканальну мережу (кілька незалежних карт ознак в одному шарі) [21]. Також слід зазначити, що при обробці площини з ваговою матрицею вона переміщується не на повний крок

(розмір цієї матриці), а на невелику відстань. Наприклад, розмір матриці з вагами  $5 \times 5$  змінюється до одного або двох нейронів (пікселів) замість п'яти, щоб не пропускати необхідні символи.

Інша операція підвибірки (підвибірка або ж об'єднання) зменшує розмірність згенерованих карт об'єктів. У цій архітектурі нейронної мережі передбачається, що дані про наявність потрібної ознаки важливіші за точне знання її координат, тому максимальне значення ознаки вибирається із кількох сусідніх нейронів карти ознак і розглядається як один нейрон. компактної карти менших об'єктів. Виконуючи цю операцію, окрім прискорення подальших обчислень, нейронна мережа стає більш інваріантною щодо масштабу вхідного зображення.

Типова структура згорткової нейронної мережі складається із великої кількості шарів. Після того, як вхідне зображення потрапляє на початковий шар, сигнал проходить через послідовність шарів зв'язування, в яких по черзі виконуються фактичне прив'язування та підвибірка (витягування). Чергування таких шарів дозволяє генерувати «карти ознак» з карт особливостей на кожному шарі, при цьому карта на кожному наступному шарі зменшується в розмірі, але збільшується кількість каналів, які підлягають обробці. На практиці можна визначити складні ієрархії ознак. Як правило, після проходження кількох таких шарів карта об'єктів стає векторною або навіть скалярною, але таких карт об'єктів сотні. На виході згорткових шарів нейронної мережі додатково створюється кілька шарів повнозв'язаної нейронної мережі (у вигляді перцептрона), вхід яких передається на створені остаточні карти ознак.

Характеристики обробки даних у згорткових нейронних мережах такі:

- зменшення розмірності — завдяки шарам субдекретизації просторова розмірність зображення поступово зменшується;
- загальні ваги — існує невеликий набір ваг для великої кількості сполук;
- локальне сприйняття — на вхід кожного окремого нейрона подається не кожне зображення (виходи попереднього шару), а лише певна його частина.

Такий підхід дозволяє не змінювати і залишати топологію зображення з одного рівня на інший.

Розпізнавання об'єктів у такій мережі не залежить від масштабу зображення. У цьому випадку сам факт знаходження знака важливіший, ніж знання його точного положення на зображенні. На малюнку 1.1 показана типова архітектура згорткової нейронної мережі [19].



Рисунок 1.1 — Структура згорткової нейронної мережі

В згортковому шарі мережі над зображеннями, що подаються на вхід, здійснюється операція згортки або ж пошук зваженої суми обраних фрагментів зображення навколо ядра згортки.

$$(f * g)[m, n] = \sum_{k, l} (f[m - k, n - l] \times g[k, l]),$$

де  $f$  — вихідна матриця зображення, що обробляється;

$g$  — ядро згортки;

$k$  — порядковий номер рядка ядра згортки;

$l$  — порядковий номер стовпця ядра згортки.

Роботу операції згортки продемонстровано на рисунку 1.2. Ядро згорки насправді є набір питомої ваги. В результаті цієї операції ми також отримуємо зображення, яке тепер називають картою об'єктів. Залежно від коефіцієнтів вибраного ядра сплетення карта ознак розрізняє деякі ознаки початкового вхідного зображення.

Для найбільш повного та якісного вибору властивостей початкового вхідного зображення використовується кілька різних ядер згортки, й таким



чином на виході шару згортки генерується кілька карт ознак. Ядро Plexus — це фільтр або вікно, яке переміщується по попередній карті та знаходить певні властивості об'єктів зображення.

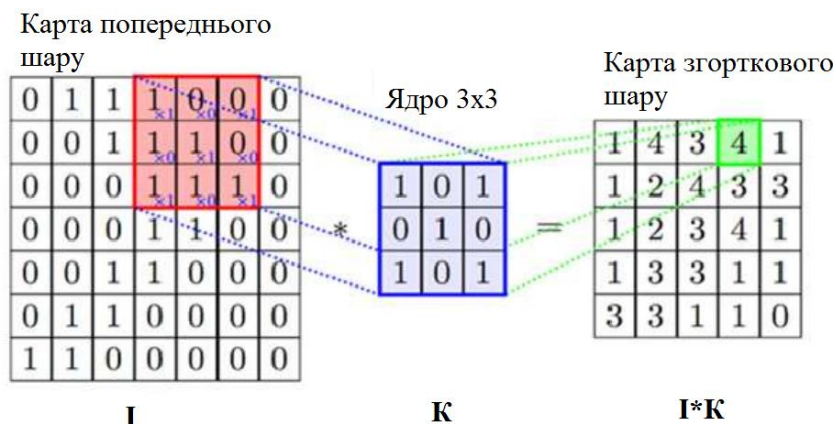


Рисунок 1.2 — Операція згортки

Отриманий скалярний результат операції згортки припадає на функцію активації. Для цієї мети традиційно використовується гіперболічна або сигмоїдна дотична функція. Зараз ReLU частіше використовується для прискорення процесу навчання мережі та спрощення обчислень завдяки своїй простоті.

Наступний шар підвибірки — це операція стиснення нелінійної карти ознак, і група пікселів (зазвичай  $2 \times 2$ ) стискається до одного пікселя після нелінійного перетворення. Для цього найчастіше використовується функція знаходження максимуму. Перетворення складається із прямокутників, що не перетинаються або квадратів, кожен із яких перекладається в піксель та вибирається піксель з максимальним значенням. Виконання операції підвибірки може значно зменшити обсяг зображення в просторі. Опір інтерпретується наступним чином — якщо ж певні ознаки вже були розпізнані під час попередньої операції згортки, таке детальне зображення більше не потрібне для подальшої обробки і тому стискається до менших деталей. Відфільтрування непотрібних частин картини допомагає мережі не вчитися з нуля. Як правило, шар часткового візерунка розміщується після тканого шару, перед наступним тканним шаром.

Після кількох операцій перетворення зображення, а потім його стиснення за допомогою операції підвибірки, система переформатується із зазначеної початкової сітки пікселів високої роздільної здатності, щоб створити більш абстрактні карти ознак. Кожен наступний шар мережі збільшує кількість каналів і зменшує розмір аналізованого зображення в кожному каналі. Результатом є великий набір каналів, що зберігають невелику кількість даних або навіть один параметр, які вважаються найбільш абстрактними поняттями в оригінальному зображенні [22, 23].

Потім ці дані об'єднуються та надсилаються до повністю взаємопов'язаної нейронної мережі, яка також може мати кілька шарів. У цьому випадку повністю взаємопов'язані мережеві шари вже втрачають свою початкову просторову структуру пікселів і малі відносно кількості пікселів у вихідному вихідному зображенні.

Відомою технікою морфологічної обробки є процес аналізу зображення на предмет форми. Математична морфологія використовується в різноманітних системах обробки зображень для досягнення різних цілей і на різних етапах роботи:

- розпізнавання контурів об'єктів зображення;
- покращення візуальних властивостей цифрових зображень (контрастність, яскравість тощо);
- відновлення пошкоджених зображення;
- зниження шуму.

Кореляційний метод в основному використовується для знаходження ступеня подібності виділених областей у різних кадрах. В результаті функція кореляції двох зображень дає максимум лише тоді, коли два зображення однакові. Максимально допустиме значення кореляційної функції має дорівнювати одиниці. Основним недоліком цього методу є відносно високі обчислювальні зусилля.

Першим кроком у ідентифікації об'єкта є вибір об'єкта в результируючому відеопослідовності на основі деяких його властивостей, які використовують

математичні прийоми для квантування, фільтрації, нормалізації, диференціації, фрагментації та інтеграції ознак. Для деяких типів зображень попередньою обробкою є соляризація або ж бінаризація, і тоді можна вибрати об'єкти обробки для систем реального часу на основі використання вертикальних зрізів.

У даному розділі роботи аналізується поточний стан поля спостереження комп'ютерного зору по відношенню до виділення рухомих об'єктів. Проаналізовано такі методи, як визначення переміщення ділянок пікселів між сусідніми кадрами, метод визначення зміщення прямокутних блоків на зображенні, виявлення рухомих об'єктів у відеопослідовності за допомогою нейронної мережі глибокого навчання. Розглянуто основні підходи до виділення контурів об'єктів зображення, визначено їх основні переваги та недоліки. Були обрані методи, які здатні краще впоратися із завданням виділення контурів рухомих об'єктів.

## 2 РОЗРОБКА ПОСЛІДОВНОСТІ ВИЯВЛЕННЯ РУХОМИХ ОБ'ЄКТІВ ТА ВИДІЛЕННЯ ЇХ КОНТУРІВ

Розробці послідовності виявлення рухомих об'єктів та алгоритму пошуку їх контурів у отриманій сцені відео спостереження із використанням підходу по знаходженню множини точок руху у виділеному зображенні та використанню нейронної мережі присвячений даний розділ.

### 2.1 Послідовність виявлення рухомих об'єктів

Першим кроком у системі виявлення рухомих об'єктів є пошук області руху. Для цього можна використовувати будь-який із використаних підходів. Це метод розрахунку фону, знаходження різниці кадрів, знаходження різниці в часі або ж знаходження оптичного потоку. Для того щоб вибрати найбільш підходящий для нашого випадку метод, давайте коротко розглянемо особливості їх роботи.

Простим способом вирішення проблеми виділення рухомих об'єктів є використання механізму віднімання фону від відеозображення [12]. Процедура віднімання передбачає, що для цього фільму створено фонову модель. Насправді така модель має бути сценою без рухомих об'єктів. Її слід регулярно оновлювати, щоб врахувати зміни в умовах освітлення та камери, наприклад такі як панорамування, нахил і масштабування. Іноді використовується метод для оновлення фонові моделі із часом. Процедура розрахунку фонові моделі складається з двох етапів. Відніміть фонове зображення з поточного відеокадру. Цей крок складається з віднімання, піксель за пікселем, інтенсивності кадру відео та фонового зображення.

Розділення пікселів за рахунок фону та об'єкта, тобто побудова бінарного зображення (маски). Вважається, що піксель належить об'єкту і є білим у масці, якщо різниця між інтенсивністю фону та поточним кадром для цього пікселя перевищує певний поріг, інакше вважається, що піксель належить до фону. Якість побудованої фонові моделі залежить від якості визначення положення

рухомих об'єктів за допомогою методу віднімання фону. Усі методи обчислення фону поділяються на дві групи (нерекурсивні та рекурсивні методи) залежно від механізму побудови фонового зображення:

- нерекурсивні методи використовують дані про інтенсивність пікселів заданого набору попередніх фонових моделей (або кадрів) і поточного кадру при оновленні моделі фону для поточного кадру [17];

- метод віднімання поточного та попереднього кадрів (цей метод передбачає, що для даного  $k$ -кадру фонові модель  $f_k$  у точці з координатами  $(x, y)$  збігається з попереднім кадром), а також у першій фазі алгоритм віднімання обчислює різницю між двома послідовними кадрами відеопослідовності;

- метод усереднення кількості попередніх кадрів, у цьому випадку кількість кадрів, на яких побудована фонові модель, позначається  $s$ , тому фонові модель  $f_k$  для  $k$ -кадру може бути виражена як середня яскравість  $k$  усіх вибрані попередні кадри;

- метод визначення медіани постійної кількості попередніх кадрів і метод усереднення припускають, що  $s$  — кількість кадрів, на основі яких оновлюється фонові модель;

- метод, заснований на використанні мінімаксного фільтра — використовуючи навчальну послідовність без рухомих об'єктів, визначають три значення для кожного пікселя фонового зображення (мінімальна яскравість  $i_{min}(x, y)$ , максимальна яскравість  $i_{max}(x, y)$ ) і максимальна зміна яскравості між сусідніми клітинами  $i_{max}(x, y)$  [21]).

Ці значення визначаються із параметрів кількох кадрів та періодично оновлюються для фонових моделей. Основною перевагою нерекурсивних методів є простота реалізації та швидкість оновлення фонових моделей при переході від одного кадру до іншого у відеопослідовності. Однак точність цих методів значною мірою залежить від швидкості об'єктів. Чим повільніше рухається об'єкт, тим більш невизначеним він стає. Методи з цієї групи дуже малоефективні, коли фон рухається (листя дерев, струмінь води тощо). Щоб

згладити ці ефекти, використовується так звана суміш. У цій процедурі оновлена фонова модель для кадру  $k_I$  представлена опуклою оболонкою фонової моделі  $f_k$ .

Рекурсивні методи оновлення моделі фону використовують лише дані про інтенсивність пікселів поточного кадру. Рекурсивні методи включають метод гістограми, метод представлення моделі суміші Гаусса, метод кодової книги та метод Visual Background Extractor (ViBe) [24-26].

За допомогою методу гістограми весь колірний простір розбивається на окремі біни (кошики). У растровому зображенні цього простору передбачається відрізок зміни інтенсивності, а в кольоровому зображенні — тривимірний куб. Гістограма створюється для всіх кадрів у відеопослідовності. Метод гістограми аналізує всі пікселі зображення. Залежно від інтенсивності/кольору, що спостерігається в пікселі, значення біну гістограми збільшується на одиницю. Якщо значення певного зернистості менше заданого порогу, пікселі, які є частиною цієї зернистості, належать до фону. В іншому випадку вважається, що пікселі належать об'єкту. Основною проблемою використання методу гістограми є необхідність використання додаткової пам'яті та виконання значної кількості операцій доступу до пам'яті в процесі реалізації.

Метод, заснований на використанні приблизного медіанного фільтра. Такий метод був використаний для моніторингу міського трафіку транспорту, використовувався простий підхід рекурсивного фільтра для оцінки медіани [24]. Ця техніка збільшує поточну медіану на одиницю, якщо піксель кадру перевищує вказану оцінку, і зменшує на одиницю в іншому випадку. В результаті середня оцінка зменшується до значення, коли половина вхідних пікселів більша, а половина менша. Недоліком цього методу є повільна адаптація з досить великими змінами фону. Для налаштування фону, залишеного об'єктом, який займав його тривалий час, потрібна значна кількість кадрів.

Модель гаусової суміші (GMM) [25]. При побудові фону за допомогою цього методу передбачається, що для кожного пікселя зображення я знаю історію зміни його інтенсивності/колір у всіх попередніх кадрах. Тоді ймовірність спостереження значення  $x_k$  представлена сумішшю гауссових розподілів.

Кольори компонентів вважаються незалежними і мають однакову стандартну помилку. На основі цього висновку можна зменшити обчислювальні зусилля методу, оскільки немає необхідності обчислювати матрицю, обернену до коваріаційної матриці. Таким чином визначається розподіл спостережуваних значень кольору для кожного пікселя. Нове значення кольору може бути представлено як один з головних компонентів отриманої суміші гауссових розподілів. Його також можна використовувати для оновлення параметрів моделі. Побудовані розподіли відсортовані в порядку спадання.

Метод Visual Background Extractor [26]. У цьому методі фонові модель у  $k$ -му кадрі представлена набором речень. Для класифікації пікселя  $p$  у колірному просторі будується куля  $S(x,y,R)$ , радіус якої дорівнює  $R$ , а також обчислюється кількість векторів множини  $M(p)$ , що потрапляють у побудовану сферу.

Якщо умова істинна, вважається, що піксель  $p$  належить до фону, інакше він належить об'єкту. Першим кроком є ініціалізація наборів  $M_p$  для всіх  $p$  пікселів зображення.

Інша група алгоритмів визначення руху об'єктів сцени заснована на оптичному потоці. Оптичний потік (ОП) в обробці відео — це векторне поле, яке визначає швидкість окремих точок або фрагментів поточного кадру послідовності зображення відносно попереднього кадру. Більшість алгоритмів виявлення рухомих об'єктів приймають рішення, просто порівнюючи пороги. Такий підхід обмежує мінімальну швидкість виявленої цілі, яка може бути нижчою за фонову швидкість (хмари, гілки).

Оптичний потік — це зображення видимого руху об'єкта, поверхні або краю сцени, що виникає в результаті руху спостерігача (ока або камери) відносно сцени. Алгоритми на основі оптичного потоку (запис руху, сегментація об'єктів, кодування руху) використовують цей рух об'єктів, поверхонь та країв. По суті, оптичний потік є полем швидкостей, оскільки зміщення точки між двома зображеннями відповідає миттєвій швидкості. Розрахунок оптичного потоку — визначення векторів (швидкостей) для оцінки руху об'єктів [27]. Застосування цього підходу вимагає виконання двох основних вимог:

- яскравість кожної точки об'єкта не змінюється з часом;
- найближчі точки, що належать об'єкту, рухаються з однаковою швидкістю в площині зображення.

Методи оптичного потоку обчислюють трафік між двома кадрами в кожному пікселі. Кадри знімаються при  $t$  і  $t$ . Ці методи називаються диференціальними, оскільки вони засновані на апроксимації сигналу відрізком ряду Тейлора. Тому ці підходи використовують часткові похідні за часом і просторові координати. Для двовимірного виміру (також для більшого розміру) піксель у позиції  $(x, y, t)$  з інтенсивністю  $I(x, y, t)$  зміщується на  $x$ ,  $y$  та  $t$  на кадр через те, що інтенсивність точки не змінюється. Оскільки зсув невеликий, ряд Тейлора виводить рівняння оптичного потоку [27].

Існує багато різних методів визначення ОП. Фазова кореляція є зворотною до нормованого поперечного спектру. Методи блоку — знайти положення певних областей (блоків) поточного кадру в попередньому кадрі. Алгоритми Лукаса-Канади, Горна-Шунка (Horn-Schunck) і Бакстона-Бакстона (Buxton-Buxton) є диференціальними методами для оцінки оптичного потоку від часткових похідних сигналу. При використанні алгоритму Горна-Шунка враховується, що оптичний потік є плавним по всьому зображенню. Введено дві функції, які визначають похибку та характеризують різницю між інтенсивністю пікселів вихідного та зміщеного пікселів.



З відомих методів вибираємо метод міжкадрової різниці. Це дозволяє легко виявити присутність рухомих об'єктів у сцені спостереження. Наступним кроком є пошук діапазону руху об'єкта. Найкращий спосіб ідентифікувати об'єкти є підхід виявлення контуру об'єктів. Тут потрібно підкреслити контур об'єкта і зробити його впізнаваним. Одним із кращих способів виявлення такого об'єкта є використання згорткових нейронних мереж. Останнім часом для цієї операції найбільше підходить згортка нейронна мережа YOLO. Її і використаємо у якості детектора по пошуку об'єктів.

Розвитком архітектури YOLO є модифікації YOLOv2 і YOLO9000 [28], які включають — нормалізацію даних, що дозволяє не використовувати технологію скринінгу без страху перекласифікації, зменшуючи розмір класифікатора для YOLO v2 до  $[448 \times 448]$  ImageNet збільшено на 10 епох, використання мережі, пропозиція знайти регіони, що представляють інтерес, аналогічно моделі Faster R-CNN, використовуючи k-середні для попереднього сегментування об'єктів у кожній області інтересу, що дозволяє розрізняти п'ять класів об'єктів, багато великих — масштабне навчання дозволяє мережі сегментувати та класифікувати об'єкти в різних роздільних можливостях, нова модель класифікації Darknet-19, яка включає 19 веб-шарів і 5 шарів підвибірки.

Мережа YOLO9000 має ту ж архітектуру, що й YOLO v2, але кількість початкових гіпотез обмежена трьома. Модель також передбачає використання карт функцій різних розмірів як вхідних даних, що призводить до кращого виявлення об'єктів, які змінюють масштаб під час обробки. Однак витрати на обчислення значно вищі, ніж у мережі YOLO. У той же час для деяких класів об'єктів, таких як «люди», «одяг», ймовірність правильної класифікації зменшується, що накладає обмеження на використання цієї мережевої моделі в задачі ідентифікації людей у приміщенні.

Модель YOLO третьої версії [28] використовує покращену архітектуру для виділення характеристик Darknet-53, яка включає 53 шари та 23 підключення пропускну здатності, що дозволяє скинути вплив рівня на

детектор, тобто архітектуру мережі можна змінити таким чином що кінцева кількість змін для даного завдання в процесі навчання.

Тому в запропонованому підході YOLOv3 використовується як згортка мережевої моделі для виявлення об'єктів, оскільки ця архітектура має хорошу точність виявлення та задовільний час обробки. Пошук і відстеження руху об'єктів — це два різні процеси. Під час пошуку та знаходження об'єкта виконується операція визначення місця розташування об'єкта в кадрі. Детектор пошуку об'єктів зазвичай більш складний у обчислювальному відношенні, а отже, повільніший за алгоритм подальшого відстеження руху об'єктів. Прикладами алгоритмів пошуку та виявлення об'єктів є Naara Cascades, HOG + Linear SVM і детектори глибокого навчання, такі як YOLO, Faster R-CNN і SingleShotDetectors (SSD) або інші.

Виявлення та відстеження руху об'єктів включає в себе наступні кроки:

- початковий пошук і виявлення об'єктів і створення вхідного набору даних з координатами, що обмежують розмір блоків у вигляді об'єктів;
- присвоєння унікального ідентифікатора кожному виявленому об'єкту;
- подальше відстеження кожного виявленого об'єкта під час його руху в кадрі з обов'язковим збереженням призначених йому унікальних ідентифікаторів.

Визначення траєкторії руху об'єктів, що відстежують рух, має першорядне значення. У цій роботі були визначені наступні вимоги до алгоритму відстеження об'єктів, які повинні бути виконані:

- крок виявлення об'єкта необхідно виконати один раз на початку роботи (об'єкт повинен бути виявлений першим);
- швидкість алгоритму відстеження об'єкта повинна бути швидшою за швидкість детектора пошуку об'єкта;
- алгоритм повинен бути стійким до оклюзії (приховувати, перекривати);
- можливість повторної обробки, якщо відстежений об'єкт «зникне» або переміститься за існуючі межі відеокадру;

— можливість продовження відстеження об'єктів, які з якихось причин «загубилися» між кадрами.

## 2.2 Механізм виявлення рухомих об'єктів

Розроблені алгоритми виявлення рухомих об'єктів мають поєднувати єдиний механізм концепції пошуку, виявлення та відстеження рухомих об'єктів, який зазвичай поділяється на дві окремі фази: виявлення та стеження.

Першою фазою алгоритму є процес відкриття. На етапі виявлення більш складний обчислювальний алгоритм для виявлення об'єктів у кадрі використовується для:

- визначити, чи є нові об'єкти в результуючому кадрі;
- дізнатися, чи були ідентифіковані об'єкти, які були «втрачені» з різних причин під час фази відстеження.

Відповідний тег відстеження створюється або оновлюється новими координатами для кожного об'єкта, виявленого у кадрі. Оскільки прослуховувач пошуку об'єктів зазвичай має вищу обчислювальну складність, його операція виконується один раз з  $N$  кадрами. Другим етапом алгоритму є процес відстеження. Це робиться без тенденції часто повертатися до процесу захоплення. Тег створюється для кожного з об'єктів, знайдених під час фази виявлення, для подальшого відстеження об'єкта, коли він рухається в послідовності отриманих кадрів. Механізм відстеження об'єктів повинен працювати швидше та ефективніше, ніж на попередньому етапі використання детектора пошуку об'єктів. Процес відстеження триває до тих пір, поки не буде досягнуто вказаний  $N$ -й кадр, після чого перезапускається детектор пошуку об'єктів. Надалі весь процес роботи алгоритму повторюється.

Перевага цього гібридного підходу полягає в тому, що можна використовувати високоточні методи для пошуку та виявлення об'єктів, не додаючи значного обчислювального навантаження на весь процес відстеження рухомих об'єктів. Існують методи автоматичного аналізу цифрових зображень [21], які дозволяють знаходити та вибирати рухомі об'єкти в полі зору камери,

використовуючи для цього відносно прості обчислювальні пристрої. Ці методи зазвичай передбачають створення моделі нерухомого зображення, а потім порівняння кожного нового кадру з відеопотоку зі створеною моделлю сцени та пошук відмінностей між новим кадром і сценою нерухомого зображення. При цьому зазвичай розташовуються й ізольовані не тільки рухомі об'єкти, а й перешкоди, викликані різними видами перешкод — зміщенням деяких об'єктів сцени під впливом вітру, зміною освітлення сцени через зовнішні рухомі ілюмінатори або зміною хмар. У багатьох випадках однією з перешкод є переміщення птахів, тварин і комах у безпосередній близькості від встановлених відеокамер. Тому високошвидкісні диференціальні методи для визначення місця розташування та ідентифікації рухомих об'єктів у послідовності відеокадрів мають певні труднощі з розрізненням шуму на зображенні.

Відомий алгоритм Online Boosting Tracker [14] є одним із перших алгоритмів відстеження об'єктів з досить високою точністю. Цей алгоритм використовує версії алгоритму AdaBoost і каскадів Хаара. Іншою розробкою є алгоритм MILTracker [18], який використовує навчання викладачів для пошуку місця розташування об'єкта та інших областей, де такий об'єкт все ще може існувати.

Останнім часом для аналізу цифрових зображень використовується апарат згорткової нейронної мережі [19, 22], за допомогою якого можна знаходити рухомі об'єкти під впливом перешкод. Достовірні результати аналізу цифрових зображень на основі таких нейронних мереж отримують за допомогою обчислювальних пристроїв з високою обчислювальною потужністю, а адаптація згорткових нейронних мереж до операцій фільтра перешкод вимагає значного набору зображень, на яких така нейронна мережа буде навчатися. Це створює деякі додаткові передумови для комплексного використання нейронних мереж. Щоб полегшити цей процес, радимо підкреслити обриси рухомих об'єктів.

### 2.3 Розробка алгоритму виділення контуру об'єктів

Для виділення контуру об'єктів пропонується використовувати бінарні зображення. Контур у такому зображенні можна знаходити шляхом пошуку зміни яскравості двох суміжних пікселів зображення. Пошук таких пікселів будемо здійснювати шляхом рядкового сканування, тобто це буде варіант скануючого алгоритму. Граничною точкою для аналізованого растрового зображення є точка, яка містить суміжну точку іншої яскравості. Множина усіх таких граничних точок є контуром об'єктів аналізованого зображення. Дана множина складається із деяких зв'язаних підмножин, що відносяться до окремих замкнених контурів різних об'єктів зображення. Під пошуком контуру об'єктів будемо мати на увазі задачу виділення однієї пари таких суміжних точок для кожного знайденого граничного контуру. Знаходження таких пар точок можна проводити рядковим скануванням бінарного зображення. Усі знайдені таким чином пари послідовних різнобарвних точок у сканованому рядку говорять про наявність контуру об'єкта. Коли знаходимо таку пару точок, то активується алгоритм перегляду контуру, який виділяє усі граничні точки зображення, що утворюють даний контур об'єкта. По завершенню аналізу виявленого контуру об'єкта подальший пошук та сканування триває із того місця, де саме була знайдена початкова гранична пара різнобарвних точок. Процес цього пошуку контурів об'єктів закінчується, коли завершуються сканування по рядках отриманого зображення.

Також, щоб не було повторного простежування уже виявлених контурів об'єктів, починаючи із різних граничних пар, необхідно позначити усі ті граничні точки, що були знайдені в процесі аналізу зображення. Із таким записом новий контур вважається уже виявленим, якщо у аналізованому рядку матриці знаходиться пара суміжних різнобарвних точок, для яких хоча б одна із них позначена. Таке позначення виявлених граничних точок об'єкта може бути проведене за допомогою використання допоміжної матриці зображення такого

ж розміру, що й бінарне початкове зображення. Із самого початку роботи усі точки цієї додаткової матриці мають білий колір, а в процесі простежування такі точки послідовно зафарбовуються у чорний колір. Таким чином, усі контури наявних об'єктів будуть виявлені за один перегляд отриманого бінарного зображення та при цьому жоден із контурів не буде переглядатися двічі.

Виділення контуру починається із пошуку граничної пари - двох суміжних різнобарвних точок, які знаходяться в одному рядку аналізованого бінарного зображення. Така пара різнобарвних суміжних точок зображення буде називатися парою, що переглядається. Для іншої пари точок, яка аналізується, задається напрямок від чорної точки до білої точки. Виконання процесу аналізу починається із послідовного переміщення із одного кінця пари, який простежується, у точку зображення, яка знаходиться від цієї пари ліворуч. Це надає змогу виконувати огляд контурів об'єктів таким чином, що чорний об'єкт розміщується ліворуч від шуканої границі, а біле тло розміщене праворуч.

На кожному кроці простежування обробляється одна нова точка, яка є суміжною із обома точками пари точок, і після цього дана точка заміщає у парі точок, яка переглядається, точку такого кольору, якою є вона сама. При такому підході пробна точка появляється однозначно із таких виводів. Переглянемо множину восьми суміжних точок для чорної точки із пари, яка простежується. Біла точка із пари, яка переглядається, відноситься до даної множини. В якості наступної пробної точки оберемо слідуєчу точку за цією білою точкою у напрямку обходу. Дане положення переглянutoї пари точок показано безперервною лінією, а усі нові положення точок в залежності від кольорів вибраної пробної точки приведені пунктиром.

Нехай  $(p, q)$  – простежуюча пара,  $p = (p_x, p_y)$  - чорна, а  $q = (q_x, q_y)$  - біла точка. Пробна точка  $r = (r_x, r_y)$  рахується таким способом. Якщо точки простежуючої пари знаходяться в одному стовпці або ж в одному рядку, то нова точка вибирається з даними координатами:

$$r_x = q_x + p_y - q_y, r_y = q_y + q_x - p_x.$$

В іншому випадку пробна точка буде мати координати:

$$r_x = (p_x + q_x + p_y - q_y) / 2, r_y = (p_y + q_y - p_x + q_x) / 2.$$

Можна відмітити, що два послідовних положення спостерігаючої пари точок повинні створити прямокутний трикутник. Через це маршрут простежування точок можна інтерпретувати як черговість таких положень цього трикутника, що послідовно «перекочується» по контуру об'єкта аналізованого бінарного зображення.

У підсумку одержані результати простежування точок утворюють упорядкований список. Послідовно об'єднавши між собою окремо чорні точки та окремо білі точки із цього списку в результаті отримаємо замкнутий коридор, у якому ліва сторона зображення складається із чорних точок, а права сторона утворюється аналогічно із білих точок. Безперервний контур границі об'єкта знаходиться, як полігон мінімального периметру, який розміщується усередині цього коридору та ділить різнобарвні сторони сформованого коридору. При цьому вершини полігону мінімального периметру перебувають серед множини граничних точок коридору, що описує границі виділених об'єктів.

Дані підходи повністю формують та задають алгоритм «витягування контуру». Покрокове виконання послідовності пошуку, простежування й витягування контурів об'єктів дозволяє побудувати границю об'єктів аналізованого зображення для отриманого бінарного зображення. Запропонований підхід для виділення контурів об'єктів аналізованого зображення можна показати таким алгоритмом.

**Крок 1** — початок роботи алгоритму.

**Крок 2** — формуємо матрицю аналізованого зображення розміром  $m \times n$  пікселів.

Крок 3 — створюємо нову матрицю контуру розміром  $k \times t$  пікселів для занесення даних про координати пікселів, що перебувають на границі знайденого контуру.

Крок 4 — виділяємо початковий піксел у матриці зображення  $i:=0, j:=0$ .

Крок 5 — якщо обраний піксел дорівнює 1, то перехід на 6, інакше на 21.

Крок 6 — виконуємо аналіз піксел з координатами  $i := i - 1, j := j + 1$  на виконання рівності 1. Якщо це так, то переходимо на крок 7, в іншому випадку переходимо на крок 10.

Крок 7 — вписуємо координати граничного пікселя до матриці контурів.

Крок 8 — збільшуємо координати пікселя.

Крок 9 — якщо значення  $j > n$ , то переходимо на крок 6, інакше на 23.

Крок 10 — здійснюємо аналіз пікселя із координатами  $i:=i, j:=j + 1$  на виконання рівності 1. Якщо це так, то переходимо на крок 11, в іншому випадку перехід на крок 13.

Крок 11 — записуємо координати вибраного граничного пікселя до матриці контурів. Збільшуємо координати цього пікселя. Перехід на крок 6.

Крок 12 — здійснюємо аналіз пікселя з координатами  $i:=i + 1, j:=j + 1$  на рівність 1. Якщо це так, то переходимо на крок 14, в іншому випадку походимо на крок 16.

Крок 13 — записуємо координати вибраного граничного пікселя до матриці контурів. Збільшуємо координати цього пікселя. Виконуємо перехід на крок 6.

Крок 14 — проаналізуємо піксел із координатами  $i:=i - 1, j:=j - 1$  на виконання рівності 1. Якщо це так, то виконуємо перехід на крок 17, в іншому випадку переходимо на крок 18.

Крок 15 — записуємо координати вибраного граничного пікселя до матриці контурів. Виконуємо перехід на крок 6.

Крок 16 — здійснюємо аналіз пікселя з координатами  $i:=i, j:=j - 1$  на виконання рівності 1. Якщо це так, то на крок 19, в іншому випадку виконуємо перехід на крок 20.



Крок 17 — записуємо координати граничного пікселя до матриці контурів. Виконуємо перехід на крок 6.

Крок 18 — здійснюємо аналіз пікселя з координатами  $i := i + 1$ ,  $j := j - 1$  на рівність 0. Якщо це так, то на крок 5, в іншому випадку переходимо на крок 6 алгоритму.

Крок 19 — створюємо координати нового пікселя для аналізу.

Крок 20 — якщо значення координати  $i > m$ , то переходимо на крок 23, інакше перехід на крок 5 алгоритму.

Крок 21 — переходимо на наступний рядок матриці зображення.

Крок 22 — якщо значення координати  $j > n$ , то переходимо на крок 25, інакше на крок 5.

Крок 23 — виконати вивід значення координат пікселів, що є границею контуру.

Крок 24 — кінець роботи алгоритму.

Запропонований алгоритм аналізу зображення лежить в основі роботи програми по визначенню контурів об'єктів виділеного зображення. Даний підхід дозволяє побудувати безперервний контур для будь-якого об'єкта растрового бінарного зображення.

Підводячи підсумки, можна відмітити, що даний підхід до виділення контурів об'єктів цифрового зображення є досить простим у порівнянні із методом виділення контурів із використанням властивості градієнтного підходу у аналізі зображення, тому що кількість затраченого машинного часу на виконання даного підходу по виділенню контурів об'єктів є меншою.

## 2.4 Використання нейронної мережі для виділення об'єктів

Повнозгорткові нейронні мережі використовуються для завдань, де необхідно визначити форму та положення потрібного об'єкта або об'єктів. Такі проблеми досить проблематично вирішити за допомогою простих обгорнутих нейронних мереж. Щоб визначити, коли і чому краще використовувати

повністю згортові мережі замість традиційних згорткових мереж, необхідно порівняти ці типи нейронних мереж.

Найбільш очевидна відмінність між повнозгортковими мережами та іншими типами нейронних мереж — кінцевий результат цих мереж. Прості сітки згортки використовуються для класифікації, до якого класу належить аналізоване зображення, і для визначення місця розташування об'єкта на зображенні. Кожен із цих методів створює числа або масиви чисел як кінцеві дані. Таким чином ми можемо отримати лише дуже обмежену інформацію про зображення, але не можемо внести його в необхідну форму.

Повністю згорнуті нейронні мережі виводять сегментоване зображення, розмір якого відповідає вхідному зображенню. Тому інша назва таких мереж — сегментні нейронні мережі. Сегментація — це групування об'єктів за певними загальними ознаками. Завдяки цьому ми отримуємо від такої мережі набагато більше корисної інформації, а отриманий кінцевий результат можна обробити за допомогою досить простих евристичних методів.

Щоб краще зрозуміти, як працюють повнозгорткові мережі та для яких завдань вони можуть бути використані, важливо зрозуміти їх загальну архітектуру. Під час проектування обтікання мереж в архітектуру можуть бути введені різні інструменти для підвищення точності виявлення (рівень нормалізації локальної відповіді, рівень відключення тощо). Уявіть собі базову архітектуру, яка практично не змінилася і керує роботою повної згорткової мережі. Основою сіток повної згортки є згортка цифрового зображення. Ключовими шарами є шари тканин. У шарі сплетення записують кількість виходів із шару, ядро сплетення та його поділ, розміри ядра, поглиблення.

Під час операції згортки ядро обходить усе зображення, що призводить до значення відповіді для ядра згортки в кожній точці зображення. Кількість ядер у кожному шарі тканини дорівнює добутку кількості входів і виходів із шару. Отримані результати обробляються наступним веб-шаром, в результаті чого виходять значення для решти ядер. До кожного шару згортки можна додати шари нормалізації або регуляризації (залежно від бажання та вибору

розробника). Після того, як зображення пройшли через багато шарів обробки, ми отримуємо значну кількість різних можливих інтерпретацій вихідного зображення (рис. 2.1) [22]. Після того, як оброблене зображення пройшло необхідну кількість шарів тканини, воно потрапляє в шар колекції. На цьому рівні вхідні зображення зменшуються без зміни їх кількості. Цей шар має ядро, яке рухається подібно до ядра згортки і обчислює одне значення для кожної такої області зображення. Зменшення вихідного зображення допомагає згортковій мережі обробляти більше даних. Це дозволяє додавати більше результатів до послідовних шарів згортки, а також підвищує точність результатів. Справа в тому, що на такому зменшеному зображенні ядра котушки однакового розміру тепер здатні захопити набагато більшу площу потрібного об'єкта.

Послідовність дій у вигляді шарів — згортка /.../ згортка / об'єднання (де розробник вказує кількість шарів згортки) може виконуватися кілька разів, поки не буде досягнутий потрібний мінімальний розмір зображення. Цей розмір зображення визначається експериментально.

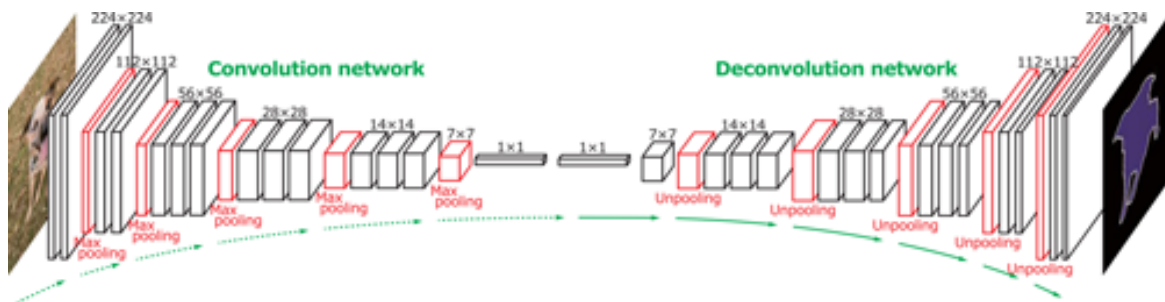


Рисунок 2.1 — Повнозгорткова нейронна мережа

Щоб знайдені об'єкти відповідали початковому масштабу, отримане зменшене зображення необхідно повернути до початкового розміру. Шар вищої вибірки бере на себе це завдання і збільшує оброблене зображення. Є два вхідних зображення для кожного виходу зміни розміру — перше — це оброблене зображення як конкретне зображення з попереднього шару (це може бути шар згортки або змішування), друге — відповідне зображення з шару

змішування, вихідний рахунок, і розмір вихідного пулу зображень дорівнює розміру підвищення дискретизації вхідного зображення.

Таким чином ми отримуємо симетричну архітектуру мережі з точки зору останнього рівня об'єднання та підвищення дискретизації першого рівня. Між шарами також розміщують шари тканини, щоб збільшити розмір малюнка, але кількість виходів з них поступово зменшується.

Послідовність шарів *upsample / convolution /... / convolution* необхідна для того, щоб привести оброблене зображення до його початкового розміру і в той же час обмежити кількість можливих інтерпретацій цифрового зображення до кількості груп об'єктів, які шукають.

У розділі розроблено послідовність обробки цифрового зображення для пошуку рухомих об'єктів у вибраній сцені. Запропонований у роботі підхід по використанню детекторів для виділення зони руху, виділення контурів об'єктів у виділеній зоні руху та застосування нейронної мережі глибокого навчання для подальшого розпізнавання об'єктів дозволяє виконати процес обробки комп'ютерних зображень із метою знаходження рухомих об'єктів. Розроблено алгоритм роботи програми для виділення контурів рухомих об'єктів для його подальшого використання у процесі розпізнавання об'єктів з використанням нейронної мережі.

### **3 РОЗРОБКА ПРОГРАМНИХ ЗАСОБІВ ДЛЯ ВИДІЛЕННЯ КОНТУРІВ РУХОМИХ ОБ'ЄКТІВ**

Розробці програмних засобів виділення контурів рухомих об'єктів у виділеній зоні відео спостереження із використанням згорткової нейронної мережі для пошуку рухомих об'єктів присвячений даний розділ магістерської роботи.

#### **3.1 Вибір інструментальних засобів для розробки програми**

Для розробки програмного продукту маємо вибрати інструментальні засоби та середовище розробки. Вибір інструментальних засобів для створення програмного продукту складається із вирішення задачі вибору мови або мов програмування та визначення необхідних додаткових бібліотек, що у певній мірі дозволять спростити процес формування нового програмного продукту, використавши вибір уже створених модулів для програми по вирішенню поставленої задачі. Для усунення проблем по можливій несумісності наявних апаратних й програмних платформ та можливих додаткових інших проблем, що можуть виникнути при створенні програми, існують різноманітні мови програмування.

Для створення бажаного програмного продукту в сучасному програмуванні найчастіше використовують об'єктно-орієнтовані мови програмування. У теперішньому програмуванні розробники програм приділяють значну увагу об'єктно-орієнтованій мові програмування, тому що вони надають можливість використати значні плюси об'єктно-орієнтованої конструкції при створенні програми на етапах конструювання й проектування програмного продукту, а також при їх реалізації та здійсненні тестування для практичному застосуванню. Тому із такими мовами будемо мати справу для створення нашого програмного продукту.

Для вибору мови та середовища програмування необхідно висунути певні критерії, як приклад, кросплатформенність, й також підтримка бібліотеки із

відкритим кодом OpenCV. Значна частина сучасних мов програмування підтримує цю бібліотеку OpenCV, й тому вибір буде виконуватися між теперішніми основними мовами програмування C#, C++ та Python.

Мова C++ є однією із сучасних мов програмування. Вона відноситься до мов високого рівня, що підтримує такі базові парадигми програмування, як узагальнену, об'єктно-орієнтовану та процедурну [29]. До стандартного складу бібліотеки мови програмування C++ входить стандартна бібліотека із мови C, зрозуміло із певними деякими змінами, які більше вже відносяться до мови C++. Значна частина бібліотеки мови програмування C++ ґрунтується на стандартній бібліотеці шаблонів. Можливості цих шаблонів надають такий корисний інструментарій у вигляді контейнерів. Ця бібліотека шаблонів дозволяє працювати також із такими типами поширених контейнерів, як черги, списки, що пов'язані із асоціативністю, та стеками даних.

Інша мова програмування C# являю собою об'єктно-орієнтовану мову, що гарантує безпечну роботу системи типізації для платформи при формуванні веб-застосунків й простих програм з використанням .NET [29]. Мова програмування C# дуже схожа по синтаксису до таких відомих мов програмування як C++ й Java. Мова має розширений перелік різних можливостей, який вона надає та підтримує. Це показники на функції класів, перевантаження операторів, підтримка поліморфізму, події, різні властивості, коментарі в форматі XML, винятки та атрибути.

Ще одна мова Python є мовою програмування високого рівня, що призначена для загального користування з використанням динамічної типізації й автоматичним підходом до керування пам'яттю [30, 31]. Ця мова створена для підвищення продуктивності роботи програмістів при створенні програм, зручності орієнтації у створеному кодї та його прочитуванні й підвищенні їх якості, й також створення фактора переносимості сформованих програм. Мова Python відноситься до мов об'єктно-орієнтованого програмування, у цій мові все є об'єктами.

Мова Python є привабливою для швидкого розроблення програм, й також як засіб поєднування наявних компонентів програм. Мова Python підтримує пакети модулів й модулі, і тому саме це сприяє повторному використанню модульності й коду. Стандартні бібліотеки й інтерпретатор мови Python доступні як у початковій, так і в скомпільованій формі на усіх основних платформах. У мові програмування Python підтримується декілька парадигм програмування, а саме об'єктно-орієнтована, процедурна, аспектно-орієнтована й функціональна.

Однією із особливостей мови Python є спосіб виділення блоків створюваного коду програм пропускними відступами. У цієї мови максимально спрощений синтаксис, що при практичному використанні значно зменшує кількість звертань до документації. Сама мова Python є інтерпретованою й використовується у тому числі й для написання скриптів.

Стандартна бібліотека мови програмування Python має значну кількість дуже корисних функцій, починаючи від функціоналу для використання роботи із текстом й завершуючи різними засобами для створення необхідних мережевих додатків. У бібліотеці мови програмування Python є значний ряд додаткових можливостей, що можуть бути використанні при застосуванні значної кількості сторонніх бібліотек, створених мовами програмування C або ж C++.

Інтерпретатор мови програмування Python може бути інтегрованим в різні проєкти, що реалізовані на цій мові. Також відомий спеціалізований депозитарій у вигляді програмного забезпечення PyPI, що утворений спеціально для цієї мови програмування. Він дозволяє виконувати досить просту установку пакетів в операційну систему й є стандартом для мови програмування Python.

Написаний мовою програмування C++ код програм є більш продуктивним у порівнянні із кодом на мові Python, але при цьому займає більше рядків. Недоліком мови програмування Python є більш значне

використання оперативної пам'яті при реалізації написаних на ній програм в порівнянні із такими компільованими мовами, як C++ або ж C.

Однією із причин вибору цієї мови для створення нашого програмного продукту є підтримка основних парадигм програмування, які вкрай бажані для виконання даної магістерської роботи, значна кількість різних корисних бібліотек та легкий код, що досить просто прочитується.

Переваги мови Python заключаються у наявності великої кількості корисних бібліотек й розширень мови, що можна використовувати у створюваних проєктах використовуючи програмні інтерфейси та гранично уніфіковані механізми імпорту; надає ресурси динамічної семантики й можливості виконання швидкого прототипування. Також у цій мові є добре продуманими механізми модульності, що надає можливості їх легкого використання.

Маючи таку значну кількість описаних позитивних переваг в порівнянні із іншими відомими мовами програмування для створення програмного продукту магістерської роботи вибираємо мову програмування Python.

Бібліотека NumPy [31] є бібліотекою у складі мови програмування Python, що формує підтримку для виконання роботи із значними багатовимірними масивами й матрицями, а також разом із великою бібліотекою математичних й інших високорівневих функцій для здійснення операцій із цими масивами.

Однією із чудових реалізацій значної кількості різних відомих методів й алгоритмів для обробки зображень й розпізнавання образів є бібліотека із відкритим кодом OpenCV. Бібліотека Open CV (OpenSource ComputerVision Library) є відомою бібліотекою для реалізації значної кількості різних методів та алгоритмів комп'ютерного зору для оброблення різного типу цифрових зображень [32].

Однією із функцій бібліотеки OpenCV є надання зручного інтерфейсу, який дозволяє програмістам швидко й ефективно реалізовувати бажані функції у засобах комп'ютерного зору. Ліцензія про надання можливостей



використання засобів із цієї бібліотеки OpenCV була реалізована таким чином, щоб надати дозвіл для створення різних комерційних продуктів із застосуванням ресурсів OpenCV.

Розроблені засоби бібліотеки OpenCV можуть застосовуватися для пошуку й ідентифікації різного типу об'єктів, виконання їх розпізнавання у різних умовах спостереження, для здійснення оперативного відстеження за переміщенням об'єктів у виділеній зоні простору з застосуванням камер відеоспостереження. Із програмної точки зору бібліотека OpenCV є чудовим сховищем високопродуктивних й низьковитратних операцій для виконання різних завдань для оброблення цифрових зображень. Перераховані можливості засобів бібліотеки OpenCV надають широкий простір дій для використання бажаних програм для створення нашого програмного продукту по виділенню контурів рухомих об'єктів та будуть задіяні при створенні програми магістерської роботи.

### 3.2 Вибір середовища розробки

Для розробки програмного забезпечення широко використовується комплекс програмних засобів у вигляді інтегрованого середовища розробки. Інтегроване середовище розробки (IDE) як правило складається із інструментів для автоматизації створення та налагодження програм, й також редактора для корегування початкового коду. Також велика кількість сучасних середовищ розробки у своєму складі мають функції для виконання можливості автодоповнення коду.

Головною метою при створенні інтегрованого середовища розробки було завдання максимізувати продуктивність розробника, щоб надати йому пов'язані інструменти розробки із подібними інтерфейсами у вигляді однієї програми, у якій буде здійснюватись весь процес розробки й яка надаватиме усі необхідні функції для виконання модифікації, компілювання, здійснення розгортання й налагодження програмного забезпечення.

Одним з завдань інтегрованого середовища розробки є зменшення кількості затраченого часу, який необхідний для формування конфігурації різних інструментів розробки, і при цьому пропонуючи цей самий набір у вигляді єдиного цілого. Це дає можливість для збільшення продуктивності роботи розробника і тільки у випадку, коли навчання та праця у інтегрованому середовищі розробки є швидшим, ніж освоєння усіх можливих інструментів. Крім цього, наявність більшої інтеграції між вбудованими інструментами середовища може сприяти додатковому збільшенню продуктивності роботи. Так, наприклад, синтаксичний аналіз створюваного коду може проходити під час його ж редагування, й при цьому будуть виявлятися можливі помилки ще до виконання трансляції цього коду.

Деякі із інтегрованих середовищ розробки використовують для певної визначеної мови програмування (або ж двох й більше споріднених мов програмування), та надають набір додаткових можливостей, що більше підходять до парадигми програмування на відповідній мові. Такими середовищами розробки є, як приклад, середовища Xcode, Hojo, PhpStorm та Delphi.

Деякі середовища містять у своєму складі інтерпретатор, компілятор або ж їх обидва (наприклад NetBeans й Eclipse), інші можуть не містити жодного із них (SharpDevelop й Lazarus). Деякі із інтегрованих середовищ розробки містять інструменти для полегшення етапу розробки інтерфейсу користувача (Embarcadero Delphi, XCode) або ж систему керування версіями. Багато сучасних середовищ містять інспектор класів, інспектор об'єктів, схему ієрархії класів для полегшення та спрощення розроблення об'єктно-орієнтованого програмного забезпечення. А з іншої сторони існує багато більш універсальних середовищ розробки, що є багатомовними, наприклад Eclipse, Microsoft Visual Studio, ActiveStateKomodo, MyEclipse, OracleJDeveloper, IntelliJ IDEA, NetBeans та Codenvy.

Після виконання аналізу усіх вищезгаданих середовищ розробки було вирішено використовувати середовище розробки PyCharm, що було створено спеціально для використання мови програмування Python.

Середовище PyCharm [33] представляє собою інтегроване середовище розробки для мови Python із використанням додаткового набору програмних засобів для здійснення оптимізації продуктивності роботи. Середовище PyCharm додає інтелектуальний редактор коду із виконанням функції автоматичного завершення рядків та готові блоки. Середовище PyCharm було випущено на ринок інтегрованих середовищ цілком для мови Python, щоб утворити конкуренцію із середовищем PyDev та ще більш поширеного середовища розробки Komodo IDE. Середовище PyCharm Community Edition, версія із безкоштовним доступом та до того ж ще із відкритим вихідним кодом, була опублікована 22 жовтня 2013 року. Дане програмне забезпечення виконує інспекцію створеного коду на льоту, також ще є автодоповнення, у тому числі використовуючи інформацію, що отримана під час виконання цього коду, під час навігації по коду, ну й ще також забезпечує доволі значну кількість рефакторингу. Ключові можливості даного середовища розробки є :

- доволі велика кількість виконання інспекцій для створеного коду;
- потужна й проста навігація в створеному кодї із підсвічуванням синтаксису;
- швидкий перегляд документації для будь-якого елемента коду прямо у вікні редактора;
- допомога при написанні коду програми, яка включає у себе поширені шаблони коду, авто доповнення та перевірка на сумісність версій інтерпретатора використовуваної мови;
- підтримку Google App Engine;
- наявність редакторів SASS, HTML/CSS, Javascript, HAML, LESS та Coffescript;
- наявність сильного рефакторингу коду, що надає широкі можливості для виконання дуже швидких й глобальних змін у створюваному проєкті;

- формування діаграм моделей Django, UML діаграм класів й Google App Engine;
- здійснення підтримки схем для найбільш широко використовуваних IDE/редакторів;
- надання інтерактивних консолей для Python, SSH, Django, баз даних й відладчика;
- надання крос-платформенності (Windows, Linux, MacOSX) [33].

### 3.3 Розробка складу програми виділення об'єктів

Процес пошуку, виділення, розпізнавання та відстеження пересування об'єктів у зоні відео спостереження складається із послідовного ряду етапів. Ці етапи формують загальну архітектуру програмного продукту для функціонування системи відео спостереження, яка виділяє задані об'єкти та виконує їх відстеження пересування, якщо це входить до завдань роботи такої системи.

Пошук та виділення об'єктів у нашому випадку здійснюється на основі знаходження зони руху у отриманому відео зображенні та виділення у ній контуру об'єктів для подальшого розпізнавання. Для цього використаємо вже деякі готові файли із відкритих бібліотек, у яких знаходиться необхідна для функціонування нашої програми інформація.

Такі відомі бібліотеки, як Dlib, OpenCV та NumPy, надають можливість роботи із відкритим кодом, і у них знаходимо значну кількість вже готових вирішень для формування нашої програми по обробці відеозображення. Створена програма по виділенню контурів об'єктів та відстеженню їх пересування сформована на основі мов C++ та Python із залученням засобів із бібліотек OpenCV, Dlib та NumPy [32, 33], що у певній мірі дозволяє полегшити процес написання програми завдяки можливостям використати напрацювання із перерахованих бібліотек для обробленню цифрових зображень по виділенню рухомих об'єктів.

Програма виділення контурів рухомих об'єктів працює у режимі реального часу — зображення надходять кадр за кадром із установленної камери або камер відео спостереження, виконується їх оброблення у запропонованій послідовності операцій, і отриманий результат роботи програми відображається на екрані монітору користувача. Перед початком роботи програми необхідно завантажити потрібні для функціонування файли із бібліотеки OpenCV у інтерпретації для виконання роботи із мовою Python у формі бібліотеки із назвою opencv-python, а також бібліотеку NumPy. Для роботи програми також підключимо дані із бібліотеки про згорткові нейронні мережі YOLO, а також базу даних для початкового налаштування та навчання нашої нейронної мережі.

Розроблена структура програмного продукту є результатом об'єднання основних модулів програмної системи для ведення відеоспостереження, а також включенням модулів, що призначені для реалізації запропонованого підходу виділення контурів рухомих об'єктів. Програмний продукт для виділення контурів рухомих об'єктів, що розробляється у даній магістерській роботі, складається із ряду модулів. Модульність створення системи відео спостереження дозволить замінювати реалізацію того або ж іншого алгоритму чи зовсім замінити один алгоритм роботи системи на інший без внесення суттєвих змін до структури системи відео спостереження в цілому. Така особливість системи виділення контурів рухомих об'єктів за отриманим зображенням просто необхідна, оскільки всі складові системи, що виконують процес виділення контурів рухомих об'єктів, часто застосовуються у різних умовах й для отримання найкращого результату в конкретних умовах може виникнути необхідність заміни деяких компонентів системи відео спостереження іншим підходящим компонентом.

Для виявлення об'єктів, що змінили своє положення у зоні спостереження, створений програмний продукт, який складається із ряду модулів. Програмна частина комп'ютерної системи відео спостереження складається із таких модулів: модуля введення даних із відео потоку, модуля

попередньої обробки, модуля виявлення руху, модуля виділення контурів об'єктів, модуля розпізнавання об'єктів та модуля виведення даних.

Модуль введення даних із відеопотоку призначений для захоплення відеозображення із відеокамери спостереження та нього покладається завдання отримати послідовність кадрів для здійснення аналізу й подальшого виявлення та відстеження рухомих об'єктів. Він забезпечує зв'язок між апаратними та програмними засобами системи відео спостереження.

Модуль попередньої обробки є одним із центральних елементів у структурі програмного продукту, так як від результатів та якості початкової обробки та аналізу вхідного потоку даних багато у чому залежить подальша робота усєї системи опрацювання відео, швидкість опрацювання даних та коректність отриманих результатів роботи. В модулі попередньої обробки введено функції підвищення якості зображень шляхом фільтрації, попереднє оцінювання та підготовка даних для наступних етапів обробки та перетворення зображення. У більшості випадків зображення для подальшого спрощення обробки, так як воно кольорове, переводиться у відтінки сірого кольору. Також на даний модуль покладається функція нормалізації яскравості й контрастності отриманого зображення, налаштування найбільш відповідного динамічного діапазону яскравості для більш точного виділення зон із елементами руху. Нормалізоване зображення передається наступному модулю для подальшого опрацювання. Частина функцій цього модуля були взяті із бібліотек із відкритим кодом. В результаті оброблення вхідного зображення цим модулем є підготовка основних параметрів зображення для виконання детекції рухомих об'єктів.

Модуль виявлення руху в сцені відео спостереження виділяє такі фрагменти в отриманому відео кадрі, де є рух, виділяє ці фрагменти зображення та передає наступному модулю для виділення об'єктів. Модуль виявлення руху є одним із головних модулів програмного продукту, в якому реалізовані вибрані підходи для виявлення руху об'єктів у відеопотоці.

Модуль виділення контурів об'єктів здійснює процес знаходження контурів рухомих об'єктів у зоні спостереження, де був виділений рух. Для цього зображення переводиться у градації сірого кольору, а потім із використанням методу Отсу виконується бінаризація виділеного фрагменту зображення. У бінарному зображенні виконується виділення контуру об'єктів зображення із використанням підходу, що запропонований у даній магістерській роботі. Для виділення контурів використовується аналіз трьох сусідніх точок, що утворюють прямокутній трикутник. Дві із цих точок є простежуваними і були вже задіяні при попередньому аналізі аналогічних трьох точок зображення. Виділені контури рухомих об'єктів аналізованого фрагменту зображення передаються на наступний модуль розпізнавання об'єктів.

Модуль розпізнавання об'єктів виконує ідентифікацію заданих об'єктів. У більшості випадків це розпізнавання людини, а при достатніх умовах спостереження і наявній базі даних зображень виявляє у отриманому фрагменті усіх осіб, що відобразилися на отриманому поточному кадрі за допомогою виявленого силуету людини, та виконує розпізнавання особи. Основою роботи цього модуля є використання загорткової нейронної мережі. Цей модуль ідентифікації відіграє роль сполучного ланцюга між усіма компонентами програмної системи та містить ядро всієї системи виділення та розпізнавання об'єктів. Для взаємодії із підсистемою розпізнавання у підсистемі ідентифікації особи має бути присутнім модуль взаємодії із підсистемою розпізнавання, який відповідає за переведення отриманого зображення для подальшої обробки. Модуль розпізнавання обробляє інформацію, що надходить до цієї підсистеми розпізнавання на предмет це група об'єктів або ж зображення особи. Залежно від типу аналізованого зображення останнє або передається у підсистему обробки результатів розпізнавання у випадку, якщо це цілий кадр, або в модуль розпізнавання особи, якщо це зображення людини.

Робота модуля розпізнавання об'єктів покладається на згорткову нейронну мережу глибокого навчання типу tiny-YOLO, у якій на основі

виділення антропометричних ключових точок обличчя людини, формування ознак для її розпізнавання та на основі попередньо виконаних дій для навчання цієї нейронної мережі здійснюється пошук, виявлення й розпізнавання особи людини.

Модуль розпізнавання об'єктів на етапі встановлення особи людини за ознаками її зовнішності виконує серію запитів до модуля бази даних для отримання необхідної для виконання операції розпізнавання кількості біометричних портретів зареєстрованих осіб. Так як умови отримання зображення сцени зони спостереження кожен раз можуть у деякій мірі відрізнятись, а зовнішність особи людини також не завжди є незмінною, то слід враховувати той факт, що при виконанні операції розпізнавання може бути присутня деяка похибка у позиціях виділених антропометричних точок особи і відповідно у значеннях виділених ознак розпізнавання. Практично всі алгоритми ідентифікації особи людини ґрунтуються на порівнянні визначеної міри близькості або ж подібності між двома групами невідомих та відомих осіб. Як зворотна міра близькості для прийняття рішення може бути використана евклідова відстань, що сформована у просторі ознак.

Якщо міра близькості при порівнянні не перевищує деяке задане число  $Q$ , то приймається рішення про те, що у даній базі даних системи не міститься біометричного портрета особи людини, для якої виконується розпізнавання. Очевидно, варіюючи значення числа  $Q$ , з'являється можливість регулювати значення ступеня або порогу роботи системи по розпізнаванню особи. Теоретично для систем розпізнавання об'єктів це число  $Q$  називають іноді порогом прийняття рішення даної системи.

База даних системи відео спостереження має перелік осіб та їх особливі характеристики для виконання операції розпізнавання осіб. У цій базі використовуються статистичні моделі у вигляді моделі форми із описом ключових точок та моделей розподілу характерних особливостей навколо цих ключових точок. Попередньо до цих моделей було виконано навчання на тестових зображеннях, де були вказані позиції ключових точок.



Модуль виведення даних виконує процедуру виведення інформації про виділений рухомий об'єкт. Якщо модуль розпізнавання налаштований на виділення та розпізнавання людини, то даний модуль інформує про особу, дані про яку розміщені у наявній базі даних. Якщо ця особа не внесена до допустимих, що можуть перебувати у зоні спостереження, то надається особливе повідомлення про додаткову увагу до даної особи. Якщо ж виявлена та розпізнана особа має право знаходитись у зоні спостереження, то сигнал тривоги не формується. Це надає можливість не відволікати увагу оператора системи відео спостереження від виконання своїх обов'язків. Модуль виведення даних є додатковим модулем, що дозволяє організувати більш зручну та комфортну роботу оператора-користувача із системою відспостереження. Даний модуль надає можливості користувачу обрати бажаний формати та способи виведення повідомлень про результатів роботи програми та інформувати користувача про виявлення процесу руху заборонених осіб у зоні відео спостереження. Відеопослідовність для аналізу формується на основі встановлених користувачем зони спостереження та параметрів для виявлення заборонених осіб у цій зоні. Модуль виведення даних виконує завдання по виведенню результатів ідентифікації людини та призначений для виконання зумовлених дій, які формуються від конкретної предметної області застосування системи відео спостереження, це може бути дозвіл доступу до обчислювальних ресурсів або даних у конкретному приміщенні, або дозвіл на відкриття дверей у приміщенні, що знаходиться під охороною, або може бути оповіщення служби охорони об'єкта про появу певної особи.

У цілому розроблена структурна схема складу модулів програми дозволяє виявляти контури рухомих об'єктів і у випадку відслідковування пересування людини у виділеній зоні відео спостереження формувати необхідну інформацію про стан об'єктів у заданій зоні.

### 3.4 Розробка програми виділення контурів рухомих об'єктів

Завдання пошуку та відстеження переміщення об'єктів, у томі числі і людини, а потім і розпізнавання особи за отриманим зображенням її обличчя розглядалися ще на початкових етапах становлення і розвитку систем комп'ютерного зору. Протягом останніх десятиріч сформувалася значна кількість різних компаній, фірм й організацій, що активно створюють різні автоматичні системи пошуку й розпізнавання об'єктів, у тому числі і людей за отриманим цифровим зображенням їх облич. У цій сфері створення програмних продуктів для пошуку та відстеження за переміщенням заданих об'єктів працюють такі фірми, як Google, ImagisTechnologies, Huawei, Hikvision, Visionics, AGAxioVision, Cisco, CorelCorporation та ряд інших. Ними створено значну кількість різного типу програмних продуктів, таких як Intelligent Scissors, Imagis, FaceNet, Visionics, Face-Intelligent, Truefaceta, RandomWalker, Epic Solutions та інші [34-36].

Одним із етапів роботи таких програм є пошук рухомого об'єкта та виділення його контуру для виконання операції подальшого розпізнавання. Саме розпізнавання може виконуватися різними засобами. У даній роботі пропонується виконати цей етап із використанням нейронної мережі. У магістерській роботі створено програмний продукт по обробленню отриманої послідовності відеозображень із метою пошуку й відстеженню пересування об'єктів у отриманій сцені спостереження. Сформована програма здійснює процес пошуку області із рухом, знаходження у виділеній сцені об'єктів, виділення контуру цих об'єктів та використанням нейронної мережі для подальшого розпізнавання об'єктів. Процес пошуку й відстеження пересування об'єктів у отриманій сцені спостереження виконується шляхом аналізу отриманої послідовності відеокadrів й відстеження за процесом пересування об'єктів у виділеній сцені відеоспостереження.

Для роботи програми спочатку імпортуємо необхідні ресурси із бібліотеки OpenCV й процеси, також дані про згорткову нейронну мережу

YOLOv3, її конфігурацію й ряд коефіцієнтів для початкового налаштування цієї мережі. Також ініціалізується відеопотік у вигляді відеофайлу або неперервного потоку відеоданих безпосередньо із веб-камери. Спочатку запускається VideoStream, завантажується бажана камера, та, нарешті, починається відлік відеокадрів у секунду. Клас VideoStream є частиною пакету imutils. Така послідовність дій виконується у багатьох програмах по обробці відео послідовностей і використовує для цього уже сформовані напрацювання для виділення окремих кадрів із відео потоку.

```
import numpy as np
import VideoStream
import yolov3.config
import yolov3.utils
import cv2
from yolov3.model import YOLO3
from pathlib import Path
```

Для здійснення операції по обробленню зображення на початковому етапі виділяємо кадри із отриманого відеопотоку. Перед цим отримуємо цей відеопотік із установленної камери спостереження. Для знаходження області руху використовуємо метод виділення між кадрової різниці. Для цього виділяємо сусідні кадри відеопотоку та присвоюємо їм відповідно назви frame1 та frame2. Цей підхід необхідний для отримання факту про знаходження об'єктів, що змінили своє положення на цих сусідніх кадрах. Пошук такого факту здійснюється за рахунок обчислення різниці у цих вибраних сусідніх кадрах. Так як процес виділення кадрів слід виконувати із певною періодичністю, то формуємо відповідний цикл оброблення отриманого відеопотоку.

Для знаходження відмінності як різниця між сусідніми кадрами відеопотоку із бібліотеки OpenCV із відкритим кодом використовуємо метод cv2.absdiff(). Виконуємо конвертування зображення із моделі кольорів BRG, що застосовується у бібліотеці OpenCV, у стандартну модель RGB.

Зручніше виконувати обробку отриманого зображення в градаціях сірого кольору, тому обчислену раніше різницю між значеннями пікселів сусідніх кадрів переведемо в відтінки сірого кольору із використанням методу `cv2.cvtColor()`, в якому укажемо уже утворену модель кольорів BRG, що досить часто застосовується при обробці кольорових зображень.

Для зменшення дії завад після отримання зображення в відтінках сірого кольору здійснюємо розмиття цього зображення. Для цього скористаємося функцією Гауса, що створена як метод `cv2.GaussianBlur` у бібліотеці OpenCV. Для свого виконання ця функція приймає ряд початкових значень — задається розмір початкового зображення, розмір результуючого цифрового зображення після виконання функції обробки, вказується розмір ядра функції Гауса та стандартні відхилення по осях  $x$  й  $y$  і вказується тип границі.

Із використанням методу `cv2.threshold()` встановлюємо порогові значення для виявлення руху у кадрах зображення. Це дозволяє сегментувати фон зображення та передній план й виявити факт наявності об'єктів, що змінюють своє положення. Метод `cv2.threshold()` вказує такі значення як тип зображення, порогове значення для виявлення факту руху, мінімальне й максимальне значення для зображення `THRESH_BINARY`.

```
video = cv2.VideoCapture("vtest.avi")
ret, frame1 = video.read()
ret, frame2 = video.read()
while video.isOpened():
    difference = cv2.absdiff(frame1, frame2)
    rgb_image = frame[:, :, :-1]
    gray = cv2.cvtColor(difference, cv2.COLOR_BGR2GRAY)
    blur = cv2.GaussianBlur(gray, (5,5), 0)
    threshold = cv2.threshold(blur, 20, 255, cv2.THRESH_BINARY)
```

Наступним кроком є етап виділення контурів рухомих об'єктів. Для цього у зоні виділеного зображення із рухом використовуємо підхід, що викладений у параграфі 2.3 даної роботи. Здійснюємо виділення контурів усіх об'єктів, що

попали у зону руху. Контури знаходимо шляхом послідовного аналізу кожної групи із трьох суміжних точок зображення, дві із яких уже були задіяні при аналізі попередньої групи точок. Таким чином формуються контури рухомих об'єктів, які подаються для подальшого розпізнавання на нейронну мережу.

Для ідентифікації рухомого об'єкту використовуємо нейронну мережу YOLOv3. Для цього попередньо перетворюємо модель цієї мережі із застосуванням навчальної бібліотеки keras. Утворюємо модель нейронної мережу YOLOv3 в режимі виводу та завантажуюмо цю попередньо навчену модель. Потім передаємо зображення моделі YOLOv3 для отримання підсумкового результату.

```
python convert.py= yolov3.cfg yolov3.weights model_data/yolo.h5
model = YOLO3 (mode="inference»,
model_dir = MODEL_DIR,
config = YOLO3Config())
model.load_weights(YOLO3_MODEL_PATH, by_name=True)
results = model.detect([rgb_image], verbose=0)
```

Виконуємо операцію розпізнавання й визначення координат об'єктів, використовуючи знайдений попередньо фрагмент зображення з застосуванням функції `image_to_process`. По завершенню операції розпізнавання об'єкту виводимо його зображення у сформованому прямокутнику. Із використанням виділених фрагментів зображення запускаємо процес виділення об'єктів у отриманому зображенні й вказуємо координати області навколо знайденого об'єкту. Виконуємо вибір бажаного формату зображення. Малюємо об'єкти, що входять до визначеного класу ( в нашому випадку це силует знайденої людини, що рухається у зоні спостереження).

```
if classes[class_index] in classes_to_look_for:
    objects_count += 1
    image_to_process = draw_object_bounding_box(image_to_process,
class_index, box)
final_image = draw_object_count(image_to_process, objects_count)
```

```
return final_image
```

Для виконання операції виводу виділеного зображення об'єкта застосовуємо рамку у відповідному вікні. Також долучаємо функцію, що дозволяє навести у знайдених у аналізованому зображенні об'єктах координати їх границь, які були визначені із застосуванням методу `apply_yolo_object_detection`. Виконуємо фільтрацію результатів для установлення рамок та формуємо отримані результати аналізу. Малюємо рамку для виділеного об'єкта та виводимо зображення кадру на екран монітору.

Програма для своєї роботи вимагає встановлення в операційній системі комп'ютера середовища .NET, й також підключення камери відео спостереження із налаштованими драйверами.

Сформований програмний код описаної вище послідовності обробки зображення із методів й класів приведений у додатку Б магістерської роботи.

### 3.5 Склад комп'ютерної системи відеоспостереження

Під системами відеоспостереження розуміють програмне та апаратне забезпечення, програмне або апаратне забезпечення, що використовується, серед іншого, методами комп'ютерного зору для автоматичного збору даних на основі аналізу потокового відео (відеоаналіз). Системи відеоспостереження можуть покладатися на алгоритми обробки та розпізнавання зображень для аналізу відео без прямого втручання людини.

Системи відеоспостереження використовуються як частина інтелектуальних систем відеоспостереження та відеопошуку. Системи відеоспостереження можуть виконувати різноманітні функції в залежності від конкретних цілей. Виявлення об'єктів, відстеження руху об'єктів, класифікація об'єктів, ідентифікація об'єктів, виявлення або виявлення різних ситуацій, включаючи турбулентні ситуації. Завданнями, які можна вирішити за допомогою систем відеоспостереження, є відстеження переміщень (переміщень) об'єктів за допомогою камер, розташованих в зоні керування, а також надання такої камери в полі зору, обраний рухомий об'єкт може

з'явитися після виходу з поля. перегляду поточної камери, відеозображення якої зараз переглядається оператором.

Найпоширенішою конфігурацією корпоративної або офісної системи спостереження є конструкція, що включає відеокамери (цифрові, в тому числі IP-камери, що передають кольорові або чорно-білі зображення), відеореєстратори (у вигляді серверів), відеоспостереження з РК-дисплеєм або ЕПТ. Монітори і ряд додаткових пристроїв, включаючи джерела безперебійного живлення, маршрутизатори, кабелі для підтримки необхідного зв'язку між основними компонентами системи, кронштейни для кріплення компонентів і корпуси для захисту від зовнішніх впливів. Метою такої системи є ідентифікація та, у разі потреби, ідентифікація та аутентифікація людей, що рухаються.

Найбільш поширена конфігурація корпоративної (офісної) системи відеоспостереження будується наступним чином:

- відеокамери (аналогові, цифрові, в тому числі IP-камери, кольорові, чорно-білі);
- відеореєстратор (сервер);
- пульт дистанційного керування для телевізійного моніторингу з моніторами (LCD, CRT);
- додаткове обладнання (джерела безперебійного живлення, маршрутизатори, кабелі, кронштейни, кришки).

Камери можна розташовувати як зовні, так і всередині приміщення. Їх можна повертати і фіксувати. Установка відеоспостереження може бути організована відкрито або приховано. Зовнішні камери відеоспостереження зазвичай розміщуються в захисних футлярах. Цифрові системи відеоспостереження працюють з відеосигналами в дискретній (цифровій) формі. Ця технологія перетворює аналогові відеосигнали в цифрові за допомогою відеопроекторів. Це необхідно для подальшої обробки та зберігання на комп'ютері.

IP-камера відеоспостереження має вбудований відеопроцесор. Сигнал, який генерує IP-камера, відправляється безпосередньо на сервер для обробки. Тут відеосигнал стискається (надлишковість видаляється) і зберігається (архівується). Глибина архівування може варіюватися від кількох днів до кількох місяців залежно від кількості камер відеоспостереження та якості зображення (швидкості стиснення інформації). На даний момент на ринку доступні пристрої для трьох типів систем, таких як системи AHD/HD-TVI, аналогові системи та цифрові IP-системи.

Існує два типи цифрових IP-систем — без використання реєстратора та з використанням реєстратора. Безрекордерна система складається з відеокамер і хмарного сервісу. Система включає в себе IP-камери (може мати від 1 до 12 мегапікселів), мережевий комутатор (може мати від 2 до 48 портів даних). Комп'ютер використовується для локального відображення даних, за допомогою якого ви можете відображати відеодані за допомогою браузера або програми. Ви також можете налаштувати архів для зберігання даних на вашому комп'ютері. Для віддаленого перегляду використовується Інтернет. У системі є камери з вбудованим хмарним сервісом, який дозволяє безкоштовно переглядати дані (запис у хмарний сервіс зазвичай є платною послугою). До складу системи входять блок живлення камери та кабелі виті пари (для підключення камери до Інтернету чи локальної мережі, а також, можливо, з додатковими проводами живлення) [11, 37].

Класична схема складається з реєстратора і відеокамер. Рекордер може бути багатоканальним, від 4 і більше каналів до 32, 64 і до 192 каналів. NVR реєстратори використовуються в системах IP-відеоспостереження. Для зберігання даних в архіві використовуємо жорсткий диск ємністю від 320 Гб до 10 ТБ. Монітор або телевізор — для локального перегляду, Інтернет — для віддаленого перегляду, IP-камера — мережевий комутатор, блок живлення камери та кабель «вита пара» схожі на попередню схему підключення. Для виконання задачі спостереження вибираємо відео камери типу HikvisionDS-2CD2T42WD-I8 [36]. Для здійснення зв'язку із центральним сервером системи



відео спостереження вибираємо медіа конвертер типу TP-LINKMC110CS. Для підключення відео камер спостереження до медіа конвертера будемо використовувати комутатор UTP3-SW08-TP120. Збереження даних про зону спостереження будемо здійснювати із використанням відео реєстратора типу DS-7732NI-K4. Ці складові допоможуть виконувати завдання відео спостереження за заданою зоною та вести облік подій, що відбуваються у цій зоні.

Крім того, IP-системи можуть включати перемикач POE як джерело живлення для всіх IP-пристроїв. У цьому випадку камери не потребують живлення. Крім того, перемикач POE бере на себе роль простого перемикача. Однак відеокамери повинні мати підтримку POE.

У даному розділі магістерської роботи здійснено вибір інструментальних засобів та середовища для розробки програми, сформовано структуру програми для виділення контурів рухомих об'єктів та розроблено програму для пошуку рухомих об'єктів виділення їх контурів та подальшого розпізнавання у отриманій сцені спостереження.

## 4 ПЕРЕВІРКА ЯКОСТІ РОБОТИ СТВОРЕНОГО ПРОГРАМНОГО ПРОДУКТУ

У даному розділі магістерської роботи здійснено вибір засобів для проведення тестування роботи розробленого програмного продукту та приведені результати експериментальних досліджень.

### 4.1 Вибір засобів для виконання тестування програми

Для здійснення тестування розробленої програми було обрано комп'ютерну систему середнього рівня, до якої входить комп'ютер із периферійним забезпеченням, веб-камера та засоби зв'язку із нею. Комп'ютер має такі технічні характеристики: процесор: Intel Core i5-1035G1 на чіпсеті Intel® B240, оперативна пам'ять 8GB (2x4GB) DIMM DDR4 (із частотою шини 2666 MHz) Crucial, жорсткий диск 256 GB (1x256GB) SATA 6Gb/s, відеокарта nVidia GeForce MX330, із об'ємом відео-пам'яті 2GB. На материнській платі комп'ютера встановлена мережа виробника Gigabyte із швидкістю передачі даних 1Gb/s. Використовувався пристрій відображення зображень (DELL P2014H 17"3) із такими технічними характеристиками матриця IPS із роздільною здатністю 1920×1080 та частотою поновлення кадрів 60 Гц і яскравістю 300 кд/м<sup>2</sup>.

Перераховані технічні параметри комп'ютера є достатніми для перевірки якості роботи розробленої програми й дозволять отримати результати для виконання аналізу коректності роботи створеної програмної системи.

Для здійснення тестування й для об'єктивного оцінювання запропонованого підходу було обрано ряд відеофайлів, які відрізняються такими критеріями, як тривалість, освітленість та кількість рухомих об'єктів.

Тривалість у всіх відео файлах була приблизно однакова із відмінністю не більше 5с. Такий інтервал часу дозволяє здійснити оцінку поведінку створеної програмної системи. У даному випадку були обрані файли, у яких не містяться значні перепади освітленості за короткий інтервал часу, тобто фактично

відсутні різкі перепади освітленості, наприклад, при включенні освітлення у темну пору суток. Кількість рухомих об'єктів був необхідний для перевірки критерію, як відреагує програмна система на значну кількість рухомих об'єктів у отриманому відеозображенні. При тестуванні були обрані відео файли, у яких кількість рухомих об'єктів становила від 1 до 5 осіб, від 6 до 10 та понад 10 осіб, що знаходяться у зоні спостереження.

При проведенні тестування програми усі експерименти із відео були виконані по 10 раз і потім результати тестування були усереднені. Такий підхід забезпечив більш об'єктивний результат перевірки, так як дозволив зменшити вплив різних похибок на проведення експериментів.

Головною задачею при виборі методів оброблення даних на цих етапах було виявлення й розпізнавання об'єктів в реальному часі, що надавали вибрані відео, із можливістю визначення вірогідності правильного виявлення об'єктів та обчислення вірогідності хибних тривоги.

Для проведення експериментальних досліджень була обрана відкрита бібліотека машинного навчання Keras, що є сумісною із TensorFlow. Для формування програм навчання нейронної мережі та її подальшого тестування використовувалася мова програмування Python. Для навчання нейронної мережі і тестування її ефективності роботи застосовувалися набори даних, що були завантажені з відкритих баз даних. Для складання цих наборів даних зображень з рухомими об'єктами із ресурсу YouTube було завантажено близько 40 різних відеороликів із рухомими людьми в різних ситуаціях Також ці відеофайли використані для створення наборів даних інших зображень із фрагментами дерев, що коливаються, та автомобілів. На етапі навчання нейронних мереж використовувалися 2000 зображень, а 500 зображень використовувалось для проведення тестування після закінчення етапу навчання.

Основними критеріями аналізу роботи програми були ймовірність правильного розпізнавання рухомого об'єкта та ймовірність помилкового виявлення об'єкта.

Для дослідження використовувалася нейронна мережа, що складаються з чотирьох згорткових шарів по 32, 64, 128 та 256 згорток у кожному із шарів, одного повнозв'язного проміжного шару із 512 нейронами й вихідного шару. До результатів виконання згорток застосовувалася функція MaxPool. Навчання нейронної мережі проводилося протягом 50 епох.

#### 4.2 Результати експериментальних досліджень

Перевірка працездатності програми виконувалася для двох аспектів пошук та виділення контуру об'єкта у залежності від відстані до нього та виділення об'єктів у залежності від їх кількості. При тестуванні відео зображень, на яких була присутня незначна кількість рухомих об'єктів (до 5) не було помітно ніяких проблем. Виявлення руху здійснювалась у всіх випадках, навіть якщо об'єкт перебував в полі зору камери не повністю. Підрахунок кількості об'єктів вівся без помилок. Проте програма показувала хибні результати при умові, якщо два об'єкти у значній мірі перекривали один одного. У такому випадку програма вважала, що на відео знаходиться один шуканий об'єкт та частина не розпізнаного об'єкту. Але як тільки цих два об'єкти розходились на таку відстань, що дозволяла провести між ними реальну візуальну границю або ж ступінь перекриття одного із об'єктів ставав менше деякого порогового значення, то програма виділяла вірну кількість об'єктів.

При перевірці роботи програми для групи відео, на яких перебувало середня кількість людей (до 10), результати роботи були подібними до попередньої групи експериментів. Однак із збільшенням кількості об'єктів на зображенні зросла і кількість таких об'єктів, які дотикаються або перекривають один одного в процесі руху. Це негативно вплинуло на підрахунок загальної кількості рухомих об'єктів у бік зменшення їх кількості. Проте виділення самого процесу руху відбувалась без відхилень.

При тестуванні відео файлів із кількістю об'єктів більше 10 виділення самого процесу руху відбувалась із високою точністю. Підрахунок рухомих об'єктів у зоні спостереження відбувався коректно, якщо між окремими

об'єктами можна було провести умовну лінію їх розмежування. Якщо ж траплялося дотикання деякої кількості об'єктів на зображенні, то формувалися контури різної форми, появлялася область значного розміру, яка ускладнювала процес підрахунку кількості рухомих об'єктів у відео, проте тільки ці об'єкти розходились, то програма продовжувала коректно підраховувати їх кількість.

Ефективність роботи згорткової мережі також була проаналізована на сукупності вибраних експериментальних даних. Результати цих досліджень показують, що правильне виявлення рухомих об'єктів досягало значення 0,97.

Для навченої нейронної мережі були проведені дослідження достовірності виявлення рухомого об'єкта у залежності від відстані до цього об'єкта. Для оцінювання достовірності роботи використали створені раніше набори даних в розмірах від 7x7 до 128x128 пікселів.

Роздільну здатність кожного із еталонних зображень перед подачею для оброблення на вхід нейронної мережі перераховувалося до формату розміром 32x32 пікселів. Для кожного із початкових розмірів еталонних форматів розраховувалося відповідне значення відстані виходячи із роздільної здатності камери відео спостереження у 1920x1080 пікселів. Результати цих досліджень приведені на рис. 4.1.

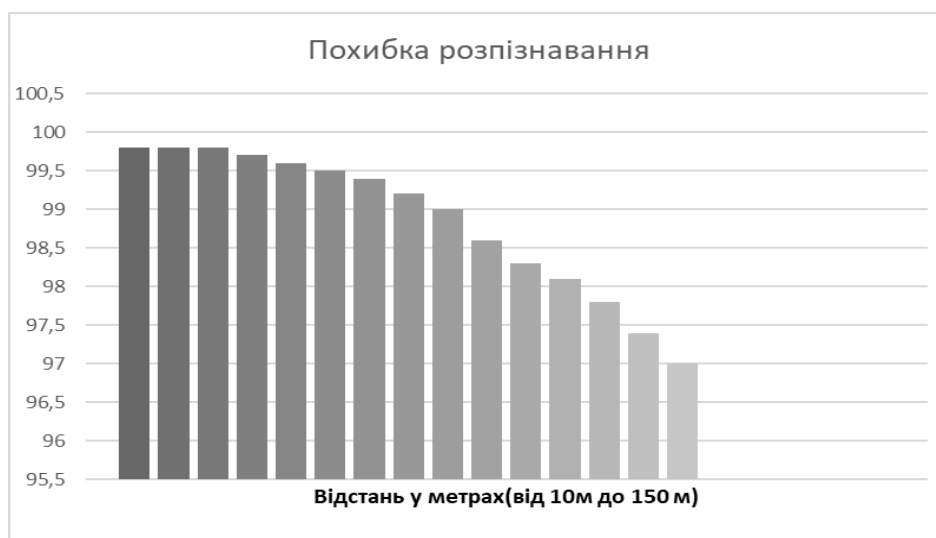


Рисунок 4.1 — Залежність точності виявлення об'єктів від відстані до них

Були проведені експериментальні дослідження обробки відеопотоку із використанням стаціонарної відеокамери, що дозволяє виконувати пошук та виявляти об'єкти на досить значних відстанях. Робота програми розділена на два основних етапи: виявлення всіх рухомих об'єктів і їх подальше розпізнавання після виділення контурів об'єктів.

Проведені експериментальні перевірки роботи програми довели її працездатність й достатню швидкість обробки відеопотоку (не нижче 8 кб/с) без застосування додаткових графічних прискорювачів, що аналогічні NVIDIA GeForce 1080.

До послідовності виявлення рухомих об'єктів висуваються такі вимоги, як нечутливість до плавної зміни фону зображення за рахунок зміни освітленості, коливання дерев, надійне виявлення руху об'єктів, що знаходяться на значній відстані, а також виявлення малорухомих об'єктів.

Досліджена послідовність виділення контурів рухомих об'єктів дозволяє виявляти й розпізнавати рухомі об'єкти із роздільною здатністю не нижче, ніж 7x7 пікселів, що відповідає відстаням до 150 м при роздільній здатності відео камери рівня FullHD й куті огляду у 60°.

У даному розділі магістерської роботи були вибрані апаратні засоби для перевірки працездатності розробленої програмної частини магістерської роботи по виділенню контурів рухомих об'єктів зображень із застосуванням мови програмування Python, використанням OpenCV та NumPy та виконано експериментальне дослідження перевірки її працездатності в умовах обробки даних, що надходять із камер відеоспостереження.

## **5 РОЗРАХУНОК ЕКОНОМІЧНОЇ ДОЦІЛЬНОСТІ СТВОРЕННЯ ПРОГРАМИ ВИДІЛЕННЯ КОНТУРІВ РУХОМИХ ОБ'ЄКТІВ**

Науково-технічна розробка має право на існування та впровадження, якщо вона відповідає вимогам часу, як в напрямку науково-технічного прогресу та і в плані економіки. Тому для науково-дослідної роботи необхідно оцінювати економічну ефективність результатів виконаної роботи.

Магістерська кваліфікаційна робота за темою «Біометрична технологія відстеження пересування людини у виділеній сцені спостереження» відноситься до науково-технічних робіт, які орієнтовані на виведення на ринок (або рішення про виведення науково-технічної розробки на ринок може бути прийнято у процесі проведення самої роботи), тобто коли відбувається так звана комерціалізація науково-технічної розробки. Цей напрямок є пріоритетним, оскільки результатами розробки можуть користуватися інші споживачі, отримуючи при цьому певний економічний ефект. Але для цього потрібно знайти потенційного інвестора, який би взявся за реалізацію цього проекту і переконати його в економічній доцільності такого кроку.

Для наведеного випадку нами мають бути виконані такі етапи робіт:

- проведено комерційний аудит науково-технічної розробки, тобто встановлення її науково-технічного рівня та комерційного потенціалу;
- розраховано витрати на здійснення науково-технічної розробки;
- розрахована економічна ефективність науково-технічної розробки у випадку її впровадження і комерціалізації потенційним інвестором і проведено обґрунтування економічної доцільності комерціалізації потенційним інвестором.

### **5.1 Проведення комерційного та технологічного аудиту науково-технічної розробки**

Метою проведення комерційного і технологічного аудиту дослідження за темою «Програмний засіб виділення контурів рухомих об'єктів в системах

комп'ютерного відеоспостереження» є оцінювання науково-технічного рівня та рівня комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності. Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням 5-ти бальної системи оцінювання за 12-ма критеріями, наведеними в табл. 5.1 [38].

Таблиця 5.1 — Оцінювання комерційного потенціалу розробки

Критерії оцінювання та бали (за 5-бальною шкалою)					
Критерій	0	1	2	3	4
Технічна здійсненність концепції:					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність продукту в реальних умовах
Ринкові переваги (недоліки):					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Багато аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів



Продовження таблиці 5.1

Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
Практична здійсненність					
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витратити значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні.	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років

## Закінчення таблиці 5.1

12	Необхідно регламентні документи та велика кількість дозвільних документів на виробництво продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту
----	---	---	---	--	---

Результати оцінювання науково-технічного рівня та комерційного потенціалу науково-технічної розробки потрібно звести до таблиці 5.2. Для оцінювання роботи були залучені експерти, викладачі кафедри ОТ експерт 1 доц. Савицька Л. А., експерт 2 доц. Войцеховська О. В. та експерт 3 доц. Ткаченко О. М.

Таблиця 5.2 — Результати оцінювання комерційного потенціалу розробки

Критерії	Експерт		
	1	2	3
	Бали:		
1. Технічна здійсненність концепції	5	5	5
2. Ринкові переваги (наявність аналогів)	3	3	4
3. Ринкові переваги (ціна продукту)	4	4	4
5. Ринкові переваги (технічні властивості)	3	4	4
5. Ринкові переваги (експлуатаційні витрати)	3	4	4
6. Ринкові перспективи (розмір ринку)	4	4	3
7. Ринкові перспективи (конкуренція)	3	3	4
8. Практична здійсненність (наявність фахівців)	3	3	3
9. Практична здійсненність (наявність фінансів)	3	4	4
10. Практична здійсненність (необхідність нових матеріалів)	4	4	4
11. Практична здійсненність (термін реалізації)	4	4	4
12. Практична здійсненність (розробка документів)	4	4	3
Сума балів	43	46	46

## Закінчення таблиці 5.2

Середньоарифметична сума балів $CB_c$	45,0
---------------------------------------	------

За результатами розрахунків, наведених в таблиці 5.2, зробимо висновок щодо науково-технічного рівня і рівня комерційного потенціалу розробки. При цьому використовуємо рекомендації, наведені в табл. 5.3 [38].

Таблиця 5.3 – Науково-технічні рівні та комерційні потенціали розробки

Середньоарифметична сума балів $CB_c$ , розрахована на основі висновків експертів	Науково-технічний рівень та комерційний потенціал розробки
41...48	Високий
31...40	Вище середнього
21...30	Середній
11...20	Нижче середнього
0...10	Низький

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Програмний засіб виділення контурів рухомих об'єктів в системах комп'ютерного відеоспостереження» становить 45,0 балів, що, відповідно до таблиці 5.3, свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки високий).

Аналіз даних ресурсу, на якому представлені отримані при тестуванні результати на запропонованих відео послідовність і метрика для оцінки ефективності алгоритмів супроводу множини об'єктів, показує, що найбільш ефективним є метод супроводу через виявлення. Цей підхід використовує ансамбль з детектора об'єктів і алгоритму для об'єднання результатів виявлень на двох кадрах. Ефективне вирішення проблеми об'єднання дозволяє коректно співвідносити результати виявлення різних об'єктів і формувати стійкі траєкторії руху для кожного з них.

В дослідженні використовувались методи теорії множин для формування множини ознак для відстеження пересування об'єктів, методи диференційного

числення для виділенню граничних контурів об'єктів, методи математичної статистики для виконання аналізу отриманих результатів розпізнавання. У роботі використано принципи об'єктно-орієнтованого програмування для реалізації запропонованого підходу.

Програмний продукт для відстеження переміщення об'єктів складається із основних модулів — модуль отримання кадрів із відео потоку, модуль виявлення руху у спостереженій сцені, модуль пошуку та виділення об'єктів, модуль відстеження руху та ідентифікації, модуль нормалізації зображення об'єктів, модуль виділення контурів рухомих об'єктів; модуль формування ознак із виявленням ключових точок бінарне зображення виділеного фрагменту руху, модуль ідентифікації, модуль обробки результатів ідентифікації. Модульність системи дозволить без особливих труднощів замінювати реалізацію того чи іншого алгоритму або зовсім замінити реалізацію одного алгоритму зовсім іншим без внесення істотних змін до структури системи в цілому. Така особливість системи виявлення людини та спостереження за її переміщенням за зображенням особи просто необхідна, оскільки всі системи, що виконують процес відслідковування осіб, часто застосовуються в різних умовах і для отримання найкращого результату, в поставлених умовах, може знадобитися замінити будь-який компонент системи найбільш підходящим.

## 5.2 Оцінювання рівня новизни розробки

Виводячи на ринок новинку виробник вважає, що тієї новизни, якою наділена нова розробка є достатньо для того, щоб вона була сприйнята споживачем як нова. Але це не завжди так, в силу того, що споживач і виробник неоднозначно визначають її рівень новизни. Тому доцільним є визначення рівня новизни розробки отриманої в результаті досліджень за темою «Біометрична технологія відстеження пересування людини у виділеній сцені спостереження».

Саме визначення рівня і ступеня інтегральної новизни є найбільш актуальним, оскільки її рівень визначає ступінь однакового позитивного сприйняття новизни розробки як виробником, так і споживачем, а отже і

ринком в цілому, а це, у свою чергу, є гарантією того, що новинка знайде своє місце на ринку, користуватиметься попитом у споживачів і забезпечить відшкодування витрат, зазнаних товаровиробником під час розроблення та виробництва технічної розробки [39].

Рівень новизни нової продукції розраховуємо експертним методом шляхом протиставлення нової продукції та її аналогів, що існують в даний час на ринку, за чинниками що визначають її значення, в системі «краще-гірше». Рівень новизни встановлюємо відносно рівня аналога (або продукту, що досить близький до аналога).

Для визначення *i*-го виду новизни, застосуємо чинники, які впливають на її рівень. Кожен чинник *i*-го виду новизни розраховуємо в балах. Більша кількість набраних балів свідчить про більший рівень новизни. Для оцінювання рівня новизни використаємо думки експертів, які встановлюють визначені бали відповідним чинникам. Бал відповідності проставляється в діапазоні від (-5 – значно гірше аналога до +5 – значно краще аналога). Результати попереднього оцінювання зведемо до відповідного листа оцінювання (таблиця 5.4).

Таблиця 5.4 — Лист оцінювання рівня новизни експертами

Види та чинники		Бали та експерти		
		Експерт 1	Експерт 2	Експерт 3
<i>I</i>		2	3	4
Споживча новизна	Питома вага 0,24	Максимальний бал $B_{iMAX}$		25
1. Зміна поведінкових звичок споживача		4	5	4
2. Ступінь задоволення потреб і запитів		5	4	4
3. Спосіб задоволення потреби		3	3	4
5. Формування нової потреби		1	1	2
5. Формування нового споживача		0	0	0
Середній бал експертів $B_{iomp}$		13		
Товарна новизна	Питома вага 0,202	Максимальний бал $B_{iMAX}$		30
1. Параметричні зміни показників продукції				
1.1. Якісні		3	4	3
1.2. Технічні		4	4	3
1.3. Економічні		3	3	3
1.5. Сервісні		4	4	4

Продовження таблиці 5.4

2. Якість продукції по відношенню до конкурентів	3	3	3
3. Функціональні зміни	3	3	3
Середній бал експертів $B_{iomp}$		20	
Виробнича новизна	Питома вага 0,042	Максимальний бал $B_{iMAX}$	25
1. Рівень унікальності товару для підприємства	5	5	5
2. Рівень унікальності для галузі	3	4	3
3. Рівень унікальності товару для країни	1	1	1
5. Зміна виробничої системи	4	4	4
5. Відносно існуючого асортименту	2	2	2
Середній бал експертів $B_{iomp}$		15	
Прогресивна новизна	Питома вага 0,2	Максимальний бал $B_{iMAX}$	25
1. Зміна технології виготовлення	4	4	4
2. Рівень застосування нових компонентів і матеріалів	1	2	1
3. Зміна технологічного принципу дії виробу	1	2	1
5. Зміна конструктивного виконання	3	2	3
5. Рівень застосування інновацій	2	2	2
Середній бал експертів $B_{iomp}$		11	
Ринкова новизна	Питома вага 0,1	Максимальний бал $B_{iMAX}$	20
1. Новий виріб на новому ринку	0	0	0
2. Новий виріб на відомому ринку	2	2	2
3. Модернізований виріб	2	2	2
5. Нова модель	1	2	2
Середній бал експертів $B_{iomp}$		6	
Екологічна новизна	Питома вага 0,035	Максимальний бал $B_{iMAX}$	20
1. Рівень екологічності чистоти технології виробництва	5	5	5
2. Рівень впровадження мало- та безвідходних технологій	5	5	5
3. Рівень екологічно небезпечних режимів експлуатації продукції	5	5	5
5. Рівень забруднення навколишнього середовища	5	5	5
Середній бал експертів $B_{iomp}$		20	
Соціальна новизна	Питома вага 0,036	Максимальний бал $B_{iMAX}$	20
1. Використання нового товару приводить до покращення стану здоров'я нації	0	0	0
2. Використання нового товару приводить до зростання доходів населення	0	0	0
3. Виробництво нового товару приводить до збільшення (зменшення) кількості робочих місць на підприємстві	4	5	4
5. Виробництво нового товару приводить до підвищення кваліфікації персоналу	3	3	3
Середній бал експертів $B_{iomp}$		7	
Маркетингова новизна	Питома вага 0,145	Максимальний бал $B_{iMAX}$	20
1. Нові методи маркетингових досліджень	0	0	0

## Закінчення таблиці 5.4

2. Вживання нових стратегій сегментації ринку	3	3	3
3. Вибір нової маркетингової стратегії обхвату і розвитку цільового сегмента	2	3	2
5. Побудова нових каналів збуту	0	1	1
Середній бал експертів $B_{iomp}$	6		

Значення і-го виду новизни розрахуємо за формулою (5.1) [39]:

$$I_i = \frac{B_{iomp}}{B_{iMAX}}, \quad (5.1)$$

де  $B_{iотр}$  — отримана кількість балів за шкалою оцінок чинників, що визначають і-й вид новизни;

$B_{iMAX}$  — максимальна кількість балів, що може бути отримана за і-м видом новизни.

Загальний рівень інтегральної новизни розраховуємо шляхом перемноження отриманого значення і-го виду новизни на її вагомість, причому вагомість і-го виду новизни визначаємо експертним методом, за формулою (5.2) [39]:

$$N_{инт} = \sum_i^n W_i \cdot I_i, \quad (5.2)$$

де  $N_{инт}$  — рівень інтегральної (сукупної) новизни;

$W_i$  — вагомість (питома вага) і-го виду новизни;

$n$  — загальна кількість видів новизни.

$$N_{инт} = (0,24 \cdot 13/25) + (0,202 \cdot 20/30) + (0,042 \cdot 15/25) + (0,2 \cdot 11/25) + (0,1 \cdot 6/20) + (0,035 \cdot 20/20) + (0,036 \cdot 7/20) + (0,145 \cdot 6/20) = 0,499.$$

Отримане значення інтегрального рівня новизни зіставляємо зі шкалою, що наведена в табл. 5.5 [38].

Таблиця 5.5 — Рівні новизни нового товару та їхня характеристика

Рівні новизни товару	Значення інтегральної новизни	Характеристика товару	Вид нового товару
Найвища	1,00	Абсолютно новий товар	Новий товар, що наділений ознаками інноваційності (інноваційний товар)
Висока	0,8...0,99	Товар, який не має аналогів	
Значуща	0,6...0,79	Принципова зміна споживчих властивостей товару	
Достатня	0,4...0,59	Принципова технологічна модифікація товару	
Незначна	0,2...0,39	Кардинальна зміна параметрів	Новий товар
Помилкова	0,00...0,19	Малоістотна модифікація	

Згідно таблиці 5.5 розробка відповідає рівню при значенні інтегральної новизни 0,499 — достатня новизна, за характеристикою — принципова технологічна модифікація програмного забезпечення; вид розробки — новий товар, що наділений ознаками інноваційності (інноваційний товар).

### 5.3 Розрахунок витрат на проведення науково-дослідної роботи

Витрати, пов'язані з проведенням науково-дослідної роботи на тему «Програмний засіб виділення контурів рухомих об'єктів в системах комп'ютерного відеоспостереження», під час планування, обліку і калькулювання собівартості науково-дослідної роботи групуємо за відповідними статтями.

До статті «Витрати на оплату праці» належать витрати на виплату основної та додаткової заробітної плати керівникам відділів, лабораторій, секторів і груп, науковим, інженерно-технічним працівникам, конструкторам, технологам, креслярам, копіювальникам, лаборантам, робітникам, студентам, аспірантам та іншим працівникам, безпосередньо зайнятим виконанням конкретної теми, обчисленої за посадовими окладами, відрядними розцінками, тарифними ставками згідно з чинними в організаціях системами оплати праці.

Витрати на основну заробітну плату дослідників ( $Z_0$ ) розраховуємо у відповідності до посадових окладів працівників, за формулою (5.3) [38]:



$$Z_o = \sum_{i=1}^k \frac{M_{ni} \cdot t_i}{T_p}, \quad (5.3)$$

де  $k$  — кількість посад дослідників залучених до процесу досліджень;

$M_{ni}$  — місячний посадовий оклад конкретного дослідника, грн;

$t_i$  — число днів роботи конкретного дослідника, дн.;

$T_p$  — середнє число робочих днів в місяці,  $T_p=24$  дні.

$$Z_o = 13280,00 \cdot 35 / 24 = 19366,66 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці 5.6.

Таблиця 5.6 — Витрати на заробітну плату дослідників

Найменування посади	Місячний посадовий оклад, грн	Оплата за робочий день, грн	Число днів роботи	Витрати на заробітну плату, грн
Керівник проекту	13280,00	553,33	35	19366,55
Інженер-розробник програмного забезпечення	12250,00	510,42	25	12760,5
Науковий співробітник з дослідження біометричних характеристик людини	12500,00	520,83	5	2604,15
Технік	7250,00	302,08	5	1510,42
Всього				36241,62

Витрати на основну заробітну плату робітників ( $Z_p$ ) за відповідними найменуваннями робіт НДР на тему «Програмний засіб виділення контурів рухомих об'єктів в системах комп'ютерного відеоспостереження» розраховуємо за формулою (5.4):

$$Z_p = \sum_{i=1}^n C_i \cdot t_i, \quad (5.4)$$

де  $C_i$  — погодинна тарифна ставка робітника відповідного розряду, за виконану відповідну роботу, грн/год;

$t_i$  — час роботи робітника при виконанні визначеної роботи, год.

Погодинну тарифну ставку робітника відповідного розряду  $C_i$  можна визначити за формулою (5.5):

$$C_i = \frac{M_M \cdot K_i \cdot K_c}{T_p \cdot t_{зм}}, \quad (5.5)$$

де  $M_M$  — розмір прожиткового мінімуму працездатної особи, або мінімальної місячної заробітної плати (в залежності від діючого законодавства), прийmemo  $M_M=2379,00$  грн;

$K_i$  — коефіцієнт міжкваліфікаційного співвідношення для встановлення тарифної ставки робітнику відповідного розряду (табл. Б.2, додаток Б) [38];

$K_c$  — мінімальний коефіцієнт співвідношень місячних тарифних ставок робітників першого розряду з нормальними умовами праці виробничих об'єднань і підприємств до законодавчо встановленого розміру мінімальної заробітної плати.

$K_i$  — коефіцієнт міжкваліфікаційного співвідношення для встановлення тарифної ставки робітнику відповідного розряду (табл. Б.2, додаток Б) [38];

$T_p$  — середнє число робочих днів в місяці, приблизно  $T_p = 24$  дн;

$t_{зм}$  — тривалість зміни, год.

$$C_1 = 2379,00 \cdot 1,10 \cdot 1,65 / (24 \cdot 8) = 22,49 \text{ грн.}$$

$$З_{р1} = 22,49 \cdot 8,20 = 184,41 \text{ грн.}$$

Розрахунки витрат на основну заробітну плату робітників у таблиці 5.7.

Таблиця 5.7 – Величина витрат на основну заробітну плату робітників

Найменування робіт	Тривалість роботи, год	Розряд роботи	Тарифний коефіцієнт	Погодинна тарифна ставка, грн	Величина оплати на робітника грн
Установка електронно-обчислювального обладнання	8,20	2	1,10	22,49	184,41

Закінчення таблиці 5.7

Підготовка робочого місця розробника програмного забезпечення	4,32	2	1,10	22,49	97,15
Інсталяція програмного забезпечення аналізу відеозображень	8,00	5	1,70	34,76	278,05
Формування дослідної бази даних відеозображень (сцени з відповідними характеристиками)	12,00	2	1,10	22,49	269,87
Налагодження програмних модулів	5,60	5	1,70	34,76	194,63
Монтаж серверного обладнання	11,00	5	1,70	34,76	382,31
Тренування нейронної мережі	20,00	4	1,50	30,67	613,34
Тестування системи	8,00	2	1,10	22,49	179,91
Всього					2199,67

Додаткова заробітна плата дослідників та робітників

Додаткову заробітну плату розраховуємо як 10 ... 12% від суми основної заробітної плати дослідників та робітників за формулою (5.6):

$$Z_{\text{дод}} = (Z_o + Z_p) \cdot \frac{H_{\text{дод}}}{100\%}, \quad (5.6)$$

де  $H_{\text{дод}}$  — норма нарахування додаткової заробітної плати, приймемо 10%.

$$Z_{\text{дод}} = (36241,62 + 2199,67) \cdot 10 / 100\% = 3844,12 \text{ грн.}$$

Нарахування на заробітну плату дослідників та робітників розраховуємо як 22% від суми основної та додаткової заробітної плати дослідників і робітників за формулою (5.7):

$$Z_n = (Z_o + Z_p + Z_{доо}) \cdot \frac{H_{зн}}{100\%} \quad (5.7)$$

де  $H_{зн}$  — норма нарахування на заробітну плату, приймаємо 22%.

$$Z_n = (36241,62 + 2199,67 + 4368,34) \cdot 22 / 100\% = 9302,79 \text{ грн.}$$

До статті «Сировина та матеріали» належать витрати на сировину, основні та допоміжні матеріали, інструменти, пристрої та інші засоби і предмети праці, які придбані у сторонніх підприємств, установ і організацій та витрачені на проведення досліджень за темою «Програмний засіб виділення контурів рухомих об'єктів в системах комп'ютерного відеоспостереження». Витрати на матеріали (М), у вартісному вираженні розраховуються окремо по кожному виду матеріалів за формулою (5.8):

$$M = \sum_{j=1}^n H_j \cdot C_j \cdot K_j - \sum_{j=1}^n B_j \cdot C_{вj}, \quad (5.8)$$

де  $H_j$  — норма витрат матеріалу  $j$ -го найменування, кг;

$n$  — кількість видів матеріалів;

$C_j$  — вартість матеріалу  $j$ -го найменування, грн/кг;

$K_j$  — коефіцієнт транспортних витрат, ( $K_j = 1,1 \dots 1,15$ );

$B_j$  — маса відходів  $j$ -го найменування, кг;

$n$  — кількість видів матеріалів;

$C_{вj}$  — вартість відходів  $j$ -го найменування, грн/кг.

$$M_1 = 3,00 \cdot 99,00 \cdot 1,1 - 0,000 \cdot 0,00 = 326,70 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці 5.8.

Таблиця 5.8– Витрати на матеріали

Найменування матеріалу, марка, тип, сорт	Ціна за 1 кг, грн	Норма витрат, кг	Величина відходів, кг	Ціна відходів, грн/кг	Вартість витраченого матеріалу,
--	-------------------	------------------	-----------------------	-----------------------	---------------------------------

Закінчення таблиці 5.8

Офісний папір 500-80	99,00	3,00	0,000	0,00	326,70
Папір для записів А5	49,00	4,00	0,000	0,00	215,60
Органайзер офісний Office	211,00	3,00	0,000	0,00	696,30
Канцелярське приладдя (набір офісного працівника)	178,00	5,00	0,000	0,00	979,00
Картридж для принтера Canon LBP	912,00	1,00	0,000	0,00	1003,20
Дископтический Optivisio CD-RW	11,10	3,00	0,000	0,00	36,63
Flesh-пам'ять DATA 16 GB	109,00	1,00	0,000	0,00	119,90
Тека для паперів	90,00	3,00	0,000	0,00	297,00
Всього					3674,33

Витрати на комплектуючі ( $K_v$ ), які використовують при проведенні НДР на тему «Біометрична технологія відстеження пересування людини у виділеній сцені спостереження» відсутні.

До статті «Спецустаткування для наукових (експериментальних) робіт» належать витрати на виготовлення та придбання спецустаткування необхідного для проведення досліджень, також витрати на їх проектування, виготовлення, транспортування, монтаж та встановлення.

Балансову вартість спецустаткування розраховуємо за формулою (5.9):

$$B_{\text{спец}} = \sum_{i=1}^k \Pi_i \cdot C_{\text{пр.і}} \cdot K_i, \quad (5.9)$$

де  $\Pi_i$  — ціна придбання одиниці спецустаткування даного виду, марки, грн;

$C_{np.i}$  — кількість одиниць устаткування відповідного найменування, які придбані для проведення досліджень, шт.;

$K_i$  — коефіцієнт, що враховує доставку, монтаж, налагодження устаткування тощо, ( $K_i = 1,10 \dots 1,12$ );

$\Pi_i$  — ціна придбання одиниці спецустаткування даного виду, марки, грн;

$k$  — кількість найменувань устаткування.

$$B_{\text{спец}} = 26394,00 \cdot 1 \cdot 1,11 = 29297,34 \text{ грн.}$$

Отримані результати зведемо до таблиці 5.9:

Таблиця 5.9 – Витрати на придбання спецустаткування по кожному виду

Найменування устаткування	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
Серверне обладнання на основі ПК EVEREST PC 13430U i5 9400F	1	26394,00	29297,34
Комплект IP-камер фіксації відеозображення	1	7605,00	8441,55
Всього			37738,89

До статті «Програмне забезпечення для наукових (експериментальних) робіт» належать витрати на розробку та придбання спеціальних програмних засобів і програмного забезпечення, (програм, алгоритмів, баз даних) необхідних для проведення досліджень, також витрати на їх проектування, формування та встановлення.

Балансову вартість програмного забезпечення, що використовується при створенні програмного продукту розраховуємо за формулою (5.10):

$$B_{\text{прог}} = \sum_{i=1}^k \Pi_{\text{прог}} \cdot C_{\text{прог}.i} \cdot K_i, \quad (5.10)$$

де  $\Pi_{\text{прог}}$  — ціна придбання одиниці програмного засобу даного виду, грн;

$C_{\text{прог}.i}$  — кількість одиниць програмного забезпечення відповідного найменування, які придбані для проведення досліджень, шт.;

$K_i$  — коефіцієнт, що враховує інсталяцію, налагодження програмного засобу тощо, ( $K_i = 1, 10 \dots 1, 12$ );

$k$  — кількість найменувань програмних засобів.

$$V_{\text{прг}} = 9840,00 \cdot 1 \cdot 1,11 = 10922,40 \text{ грн.}$$

Отримані результати зведемо до таблиці 5.10:

Таблиця 5.10 – Витрати на придбання програмних засобів по кожному виду

Найменування програмного засобу	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
Пакет імітаційного моделювання	1	9840,00	10922,40
Пакет прикладного інженерного математичного моделювання MatchLab 15 pro	1	7568,00	8400,48
Програмне забезпечення обробки відеозображень	1	6990,00	7758,90
Всього			27081,78

В спрощеному вигляді амортизаційні відрахування по кожному виду обладнання, приміщень та програмному забезпеченню тощо, розраховуємо з використанням прямолінійного методу амортизації за формулою (5.11):

$$A_{\text{обл}} = \frac{Ц_{\text{б}}}{T_{\text{в}}} \cdot \frac{t_{\text{вик}}}{12}, \quad (5.11)$$

де  $Ц_{\text{б}}$  — балансова вартість обладнання, програмних засобів, приміщень тощо, які використовувались для проведення досліджень, грн;

$t_{\text{вик}}$  — термін використання обладнання, програмних засобів, приміщень під час досліджень, місяців;

$T_{\text{в}}$  — строк корисного використання обладнання, програмних засобів, приміщень тощо, років.

$t_{\text{вик}}$  — термін використання обладнання, програмних засобів, приміщень під час досліджень, місяців;

$$A_{\text{обл}} = (31750,00 \cdot 2) / (3 \cdot 12) = 1763,89 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці 5.11.

Таблиця 5.11 – Амортизаційні відрахування по кожному виду обладнання

Найменування обладнання	Балансова вартість, грн	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн
Персональний комп'ютер	31750,00	3	2	1763,89
Робоче місце розробника програмного забезпечення	7370,00	5	2	245,67
Пристрої виводу інформації	6540,00	4	2	272,50
Оргтехніка	8390,00	4	2	349,58
Приміщення лабораторії	196500,00	25	2	1310,00
ОС Windows 10	5450,00	2	2	454,17
Прикладний пакет Microsoft Office 2016	3795,00	2	2	316,25
Всього				4712,06

Витрати на силову електроенергію ( $B_e$ ) розраховуємо за формулою (5.12):

$$B_e = \sum_{i=1}^n \frac{W_{yi} \cdot t_i \cdot \zeta_e \cdot K_{\text{ени}}}{\eta_i}, \quad (5.12)$$

де  $W_{yi}$  — встановлена потужність обладнання на визначеному етапі розробки, кВт;

$t_i$  — тривалість роботи обладнання на етапі дослідження, год;

$\zeta_e$  — вартість 1 кВт-години електроенергії, грн; (вартість електроенергії визначається за даними енергопостачальної компанії), прийmemo  $\zeta_e = 4,50$  грн;



$K_{\text{впі}}$  — коефіцієнт, що враховує використання потужності,  $K_{\text{впі}} < 1$ ;

$t_i$  — тривалість роботи обладнання на етапі дослідження, год;

$\eta_i$  — коефіцієнт корисної дії обладнання,  $\eta_i < 1$ .

$$B_e = 0,45 \cdot 300,0 \cdot 4,50 \cdot 0,95 / 0,97 = 607,50 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці 5.12.

Таблиця 5.12 — Витрати на електроенергію

Найменування обладнання	Встановлена потужність, кВт	Тривалість роботи, год	Сума, грн
Персональний комп'ютер	0,45	300,0	607,50
Робоче місце розробника програмного забезпечення	0,15	240,0	162,00
Пристрої виводу інформації	0,03	30,0	4,05
Оргтехніка	0,70	7,5	23,63
Серверне обладнання на основі ПК EVEREST PC 13430U i5 9400F	0,65	240,0	702,00
Комплект IP-камер фіксації відеозображення	0,05	240,0	54,00
Всього			1553,18

До статті «Службові відрядження» дослідної роботи на тему «Програмний засіб виділення контурів рухомих об'єктів в системах комп'ютерного відеоспостереження» належать витрати на відрядження штатних працівників, працівників організацій, які працюють за договорами цивільно-правового характеру, аспірантів, зайнятих розробленням досліджень, відрядження, пов'язані з проведенням випробувань машин та приладів, а також витрати на відрядження на наукові з'їзди, конференції, наради, пов'язані з виконанням конкретних досліджень.

Витрати за статтею «Службові відрядження» розраховуємо як 20...25% від суми основної заробітної плати дослідників та робітників за формулою (5.13):

$$B_{cs} = (Z_o + Z_p) \cdot \frac{H_{cs}}{100\%}, \quad (5.13)$$

де  $H_{cs}$  — норма нарахування за статтею «Службові відрядження», прийmemo  $H_{cs} = 20\%$ .

$$B_{cs} = (36241,62 + 2199,67) \cdot 20 / 100\% = 7688,25 \text{ грн.}$$

Витрати за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації» розраховуємо як 30...45% від суми основної заробітної плати дослідників та робітників за формулою (5.14):

$$B_{cn} = (Z_o + Z_p) \cdot \frac{H_{cn}}{100\%}, \quad (5.14)$$

де  $H_{cn}$  — норма нарахування за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації», прийmemo  $H_{cn} = 30\%$ .

$$B_{cn} = (36241,62 + 2199,67) \cdot 30 / 100\% = 11532,38 \text{ грн.}$$

До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками.

Витрати за статтею «Інші витрати» розраховуємо як 50...100% від суми основної заробітної плати дослідників та робітників за формулою (5.15):

$$I_s = (Z_o + Z_p) \cdot \frac{H_{is}}{100\%}, \quad (5.15)$$

де  $H_{is}$  — норма нарахування за статтею «Інші витрати», прийmemo  $H_{is} = 55\%$ .

$$I_s = (36241,62 + 2199,67) \cdot 55 / 100\% = 21142,7 \text{ грн.}$$

До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів;

витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов'язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін.

Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуємо як 100...150% від суми основної заробітної плати дослідників та робітників за формулою (5.16):

$$B_{нзв} = (Z_o + Z_p) \cdot \frac{H_{нзв}}{100\%}, \quad (5.16)$$

де  $H_{нзв}$  — норма нарахування за статтею «Накладні (загальновиробничі) витрати», прийmemo  $H_{нзв} = 115\%$ .

$$B_{нзв} = (36241,62 + 2199,67) \cdot 115 / 100\% = 44207,48 \text{ грн.}$$

Витрати на проведення науково-дослідної роботи на тему «Біометрична технологія відстеження пересування людини у виділеній сцені спостереження» розраховуємо як суму всіх попередніх статей витрат за формулою (5.17):

$$B_{заг} = Z_o + Z_p + Z_{доо} + Z_n + M + K_{г} + B_{снец} + B_{прз} + A_{обл} + B_e + B_{св} + B_{сн} + I_{г} + B_{нзв}. \quad (5.17)$$

$$B_{заг} = 36241,62 + 2199,67 + 3844,12 + 9302,79 + 3674,33 + 0,00 + 37738,89 + 27081,78 + 4712,06 + 1553,18 + 7688,25 + 11532,38 + 21142,7 + 44207,48 = 210919,25 \text{ грн.}$$

Загальні витрати ЗВ на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховується за формулою (5.18):

$$ЗВ = \frac{B_{заг}}{\eta}, \quad (5.18)$$

де  $\eta$  - коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи, прийmemo  $\eta=0,9$ .

$$ЗВ = 210919,25 / 0,9 = 234354,72 \text{ грн.}$$

#### 5.4 Розрахунок економічної ефективності науково-технічної розробки при її можливій комерціалізації потенційним інвестором

В ринкових умовах узагальнюючим позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів цієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку.

Результати дослідження проведені за темою «Програмний засіб виділення контурів рухомих об'єктів в системах комп'ютерного відеоспостереження» передбачають комерціалізацію протягом 4-х років реалізації на ринку.

В цьому випадку майбутній економічний ефект буде формуватися на основі таких даних.

$\Delta N$  — збільшення кількості споживачів продукту, у періоди часу, що аналізуються, від покращення його певних характеристик (таблиця 5.13).

Таблиця 5.13 — Зростання кількості споживачів

Показник	1-й рік	2-й рік	3-й рік	4-й рік
Збільшення кількості споживачів, осіб	1000	2500	3500	1000

$N$  — кількість споживачів які використовували аналогічний продукту році до впровадження результатів нової науково-технічної розробки, прийmemo 8000 осіб;

$C_o$  — вартість програмного продукту у році до впровадження результатів розробки, прийmemo 6000,00 грн;

$\pm \Delta C_o$  — зміна вартості програмного продукту від впровадження результатів науково-технічної розробки, прийmemo 500,00 грн.

$N$  — кількість споживачів які використовували аналогічний продукту році до впровадження результатів нової науково-технічної розробки, прийmemo 8000 осіб;

Можливе збільшення чистого прибутку у потенційного інвестора  $\Delta \Pi_i$  для кожного із 4-х років, протягом яких очікується отримання позитивних

результатів від можливого впровадження та комерціалізації науково-технічної розробки, розраховуємо за формулою (5.19) [38]:

$$\Delta\Pi_i = (\pm\Delta C_o \cdot N + C_o \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\vartheta}{100}\right), \quad (5.19)$$

де  $\lambda$  — коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість, у 2022 році ставка податку на додану вартість складає 20%, а коефіцієнт  $\lambda = 0,8333$ ;

$\rho$  — коефіцієнт, який враховує рентабельність інноваційного продукту).

Прийmemo  $\rho = 25\%$ ;

$\vartheta$  — ставка податку на прибуток, який має сплачувати потенційний інвестор, у 2022 році  $\vartheta = 18\%$ .

Збільшення чистого прибутку 1-го року:

$$\Delta\Pi_1 = (500,00 \cdot 8000,00 + 6500,00 \cdot 1000) \cdot 0,83 \cdot 0,25 \cdot (1 - 0,18/100\%) = 1786575,00 \text{ грн.}$$

Збільшення чистого прибутку 2-го року:

$$\Delta\Pi_2 = (500,00 \cdot 8000,00 + 6500,00 \cdot 3500) \cdot 0,83 \cdot 0,25 \cdot (1 - 0,18/100\%) = 4551512,50 \text{ грн.}$$

Збільшення чистого прибутку 3-го року:

$$\Delta\Pi_3 = (500,00 \cdot 8000,00 + 6500,00 \cdot 7000) \cdot 0,83 \cdot 0,25 \cdot (1 - 0,18/100\%) = 8422425,00 \text{ грн.}$$

Збільшення чистого прибутку 4-го року:

$$\Delta\Pi_4 = (500,00 \cdot 8000,00 + 6500,00 \cdot 8000) \cdot 0,83 \cdot 0,25 \cdot (1 - 0,18/100\%) = 9528400,00 \text{ грн.}$$

Приведена вартість збільшення всіх чистих прибутків ПП, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки (формула 5.20):

$$ПП = \sum_{i=1}^T \frac{\Delta\Pi_i}{(1 + \tau)^t}, \quad (5.20)$$

де  $\Delta\Pi_i$  — збільшення чистого прибутку у кожному з років, протягом яких виявляються результати впровадження науково-технічної розробки, грн;

$T$  — період часу, протягом якого очікується отримання позитивних результатів від впровадження та комерціалізації науково-технічної розробки, роки;

$\tau$  — ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні,  $\tau=0,11$ ;

$T$  — період часу, протягом якого очікується отримання позитивних результатів від впровадження та комерціалізації науково-технічної розробки, роки;

$t$  — період часу (в роках) від моменту початку впровадження науково-технічної розробки до моменту отримання потенційним інвестором додаткових чистих прибутків у цьому році.

$$\begin{aligned} \text{ПП} &= 1786575,00/(1+0,11)^1 + 4551512,50/(1+0,11)^2 + 8422425,00/(1+0,11)^3 + \\ &+ 9528400,00/(1+0,11)^4 = 1609527,03 + 3694109,65 + 6158404,57 + 6276652,21 = 17738693,46 \text{ грн.} \end{aligned}$$

Величина початкових інвестицій  $PV$ , які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки розраховуються за формулою (5.21):

$$PV = k_{инв} \cdot 3B, \quad (5.21)$$

де  $k_{инв}$  — коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію, приймаємо  $k_{инв}=1,5$ ;

$3B$  — загальні витрати на проведення науково-технічної розробки та оформлення її результатів, приймаємо 234354,72 грн.

$$PV = k_{инв} \cdot 3B = 1,5 * 234354,72 = 351532,08 \text{ грн.}$$

Абсолютний економічний ефект  $E_{abc}$  для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме формула (5.22):

$$E_{abc} = III - PV \quad (5.22)$$

де  $III$  — приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково-технічної розробки, 17738693,46 грн;

$PV$  — теперішня вартість початкових інвестицій, 351532,08 грн.

$$E_{abc} = III - PV = 17738693,46 - 351532,08 = 17381161,3 \text{ грн.}$$

Внутрішня економічна дохідність інвестицій  $E_g$ , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки (формула 5.23):

$$E_g = T_{ж} \sqrt[4]{1 + \frac{E_{abc}}{PV}} - 1, \quad (5.23)$$

де  $E_{abc}$  — абсолютний економічний ефект вкладених інвестицій, 17381161,3 грн;

$PV$  — теперішня вартість початкових інвестицій, 351532,08 грн;

$T_{ж}$  — життєвий цикл науково-технічної розробки, тобто час від початку її розробки до закінчення отримання позитивних результатів від її впровадження, 4 роки.

$$E_g = T_{ж} \sqrt[4]{1 + \frac{E_{abc}}{PV}} - 1 = (1 + 17381161,3 / 351532,08)^{1/4} - 1 = 1,66.$$

Мінімальну внутрішню економічну дохідність вкладених інвестицій  $\tau_{min}$  обчислюємо за формулою (5.24):

$$\tau_{min} = d + f, \quad (5.24)$$

де  $d$  — середньозважена ставка за депозитними операціями в комерційних банках; в 2022 році в Україні  $d=0,25$ ;

$f$  — показник, що характеризує ризикованість вкладення інвестицій, прийmemo 0,27.

$\tau_{\min} = 0,25+0,27 = 0,54 < 1,66$  свідчить про те, що внутрішня економічна дохідність інвестицій  $E_g$ , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки вища мінімальної внутрішньої дохідності. Тобто інвестувати в науково-дослідну роботу за темою «Програмний засіб виділення контурів рухомих об'єктів в системах комп'ютерного відеоспостереження» доцільно.

Період окупності інвестицій  $T_{ок}$  які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки обчислюємо за формулою (5.25):

$$T_{ок} = \frac{1}{E_g}, \quad (5.25)$$

де  $E_g$  — внутрішня економічна дохідність вкладених інвестицій.

$$T_{ок} = 1 / 1,66 = 0,60 \text{ р.}$$

$T_{ок} < 3$ -х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Програмний засіб виділення контурів рухомих об'єктів в системах комп'ютерного відеоспостереження» становить 45,0 бали, що, свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки високий).



Абсолютний економічний ефект для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки протягом чотирьох років становить 17381161,3 грн.

Термін окупності становить 0,6 року, що менше 3-х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

Отже можна зробити висновок про доцільність проведення науково-дослідної роботи за темою «Програмний засіб виділення контурів рухомих об'єктів в системах комп'ютерного відеоспостереження».

## ВИСНОВКИ

Актуальною задачею сьогодення є створення систем виділення й розпізнавання рухомих об'єктів із застосуванням засобів комп'ютерного зору. Рішення у цій області інформаційних технологій знаходять широке примінення в самих різних сферах та галузях діяльності людства: забезпечення захисту від несанкціонованого проникнення зловмисників на об'єкти під охороною, відстеження появи автомобіля на парковці, розпізнавання руху людини за її ходою та інше. У даній магістерській кваліфікаційній роботі розроблені підхід, алгоритм та засоби пошуку й виділення контурів рухомих об'єктів у отриманих відео зображеннях у заданій зоні спостереження.

У першому розділі магістерської роботи був проведений огляд особливості роботи систем комп'ютерного зору, виконаний аналіз методів виділення руху, таких як визначення зміщення прямокутних блоків у зображенні та переміщення областей пікселів між сусідніми кадрами, а також методу оптичного потоку, проведений аналіз методів виділення контурів об'єктів та формування етапів виділення об'єктів, використання нейромережі глибокого навчання для розпізнавання рухомих об'єктів у відео послідовності.

У другому розділі магістерської роботи вдосконалено метод пошуку рухомих об'єктів у відео послідовності та виділення їх контурів. Основним підходом до пошуку й виділення рухомих об'єктів використано пошук області руху із використанням міжкадрової різниці, виділення контуру об'єкта із використанням переведення зображення у бінарне та виділення контуру об'єктів шляхом послідовного аналізу кожної із трьох суміжних різнобарвних точок аналізованого зображення у вигляді прямокутного трикутника, розроблено алгоритм виділення контуру та застосування нейронної мережі глибокого навчання для розпізнавання рухомих об'єктів у отриманому відео зображенні.

У третьому розділі магістерської роботи виконано вибір мови та середовища розробки програмного продукту, розроблено програму для

реалізації запропонованого підходу по виділенню контурів рухомих об'єктів при обробці відео зображення для пошуку рухомих об'єктів у вибраній сцені.

У четвертому розділі магістерської роботи описано перевірку працездатності розробленого програмного продукту та проведено експериментальне дослідження його роботи.

У п'ятому розділі магістерської роботи виконані економічні розрахунки із обґрунтування доцільності виконання нової роботи по виділенню контурів рухомих об'єктів у відео зображеннях, обчислені фінансові затрати на створення програмного продукту та визначено економічні переваги від впровадження запропонованого наукового рішення у вигляді завершеного програмного продукту.

Створений програмний продукт може використовуватися у комп'ютерних системах відеоспостереження для виділення контурів рухомих об'єктів.

**ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ**

1. Шапиро Л. Компьютерное зрение / Л. Шапиро, Дж. Штокман — М.: Бином, 2009. — 763с.
2. Волковицкий В.Д. Системы контроля и управления доступом./ В. Д. Волковицкий, В. В. Волхонский — М.: Экополис и культура, 2017. — 376 с.
3. Ворона В.А. Системы контроля и управления доступом./ В.А. Ворона, В.А.Тихонов — М.: Горячая линия Телеком, 2010. — 272 с.
4. Визильтер Ю. В. Обработка и анализ изображения в задачах машинного зрения. / Ю.В.Визильтер, С. Ю. Желтов — М.: Физматкнига, 2010. — 672 с.
5. Лавренюк В. С. Виділення контурів рухомих об'єктів у системах відеоспостереження. / В. С. Лавренюк, М. А. Очкуров // Всеукраїнська науково-практична інтернет-конференція «Молодь в науці: дослідження, проблеми, перспективи (МН-2022)». [Електронний ресурс] — Режим доступу: <https://conferences.vntu.edu.ua/index.php/mn/mn2022/paper/viewFile/15629/13139>.
6. Желтов С. Ю. Обработка и анализ изображений в задачах машинного зрения / С. Ю. Желтов. — М.: Физматкнига, 2010. — 672 с.
7. Фомин Я. А. Распознавание образов: теория и применение. / Я. А. Фомин. — М.: ФАЗИС, 2012.— 429 с.
8. Заяць В. М. Методи розпізнавання образів. Навч. посібник. / В. М. Заяць, Р. М. Камінський. — Львів, видав. Національного університету «Львівська політехніка», 2004. — 176 с.
9. Гонсалес Р. Цифровая обработка изображений. / Р. Гонсалес, Р. Вудс. — Москва: Техносфера, 2005. — 1072с.
10. Обухова Н. А. Обнаружение и сопровождение движущихся объектов методом сопоставления блоков // Информационно-управляющие системы. — 2004. — № 1. — С. 30 — 37.

11. Лукьяница А. А. Цифровая обработка видеоизображений / А. А. Лукьяница, А. Г. Шишкин — Москва: «Ай-Эс-Эс Пресс», 2009. — 145с.
12. Бабарика А. О. Обгрунтування показника вибору оптимального алгоритму виділення фону у відеопослідовностях з камер відеоспостереження відомчих систем відео спостереження. / А. О. Бабарика // Сучасні інформаційні технології у сфері безпеки та оборони .— 2019. — № 3. — с. 97—102.
13. Chan T. Active contours without edges / T. Chan, L. Vese // IEEE transactions on image processing. — V10, № 3 — P. 266—277.
14. Tissainayagam P. Object Tracking in Image Sequences Using Point Features / P. Tissainayagam, D. Suter // Pattern Recognition .— Vol. 38. — 2005. — p.105—113.
15. Дослідження і розробка алгоритмів визначення меж об'єктів на зашумлених зображеннях. [Електронний ресурс]. — Режим доступу: <http://masters.donntu.org/2018/fknt/frolov/diss/indexu.htm>.
16. Обзор основных методов контурного анализа для выделения контуров движущихся объектов [Електронний ресурс]. — Режим доступу: <http://www.engjournal.ru/articles/1280/1280.pdf>.
17. Медведева Е. В. Метод выделения движущихся объектов в видеопотоке на основе оценки их границ/ Е. В. Медведева, К. А. Карлушин, Е. Е. Курбатова // Машинное обучение и анализ данных. — 2015. Т. 1, № 12. — С. 1696 — 1705.
18. Алпатов Б. А. Подходы к обнаружению и оценке параметров движущихся объектов на видеопоследовательности применительно к транспортной аналитике / Б. А. Алпатов, П. В. Бабаян, М. Д. Ершов // Компьютерная оптика. — 2020. — Т. 44, № 5. — С. 746-756.
19. Хайкин С. Б. Нейронные сети: полный курс. — 2-е изд. Пер. с англ. — М.: Издательский дом «Вильямс», 2006. — 1104 с.
20. Павлов В. А. Применение свёрточной нейронной сети для сопровождения объектов на последовательности аэрофотоснимков /

С. Б. Макаров, А. И. Бобровский, В. А. Павлов, А. К. Безбородов // Вопросы радиоэлектроники. Серия Техника телевидения. — 2020. — № 2. — С. 3 — 18.

21. Гетьманець О. І. Алгоритм перетворення відеопотоку для прискорення та оптимізації пошуку переміщення об'єкту / О. І. Гетьманець, О. О. Гагарін. // Штучний інтелект. — 2017. — №2. — С. 38—44.

22. Синєглазов В. А. Глибокі нейронні мережі для вирішення завдань розпізнавання і класифікації зображення. — Україна: Національний авіаційний університет, 2020. — 276 с.

23. Лінійна нейронна мережа [Електронний ресурс]. Режим доступу: <https://uk.discografie.org/369813-linear-vs-nonlinear-neural-network-PUTODO>.

24. Способ подчеркивания контуров изображения движущихся объектов. Патент RU №2337501, МПК G06T7/20, опубл. 27.10.2008.

25. System and method of processing information (images) aimed at tracking a person through multiple cameras. Патент US №9390332 B2, МПК G06T 7/00, опубл. 12.07.2016.

26. Способ выделения контуров движущихся объектов. Патент RU № 2466456 C2, МПК G06K 9/50.

27. Lucas B.D. An Iterative Image Registration Technique with an Application to Stereo Vision / B.D. Lucas, T. Kanade // Proceedings of the 7th international joint conference on Artificial intelligence. — 1981. — Vol. 2. — P. 674—679.

28. Redmon, J. You only look once: Unified, real-time object detection / J. Redmon, S.K. Divvala, R.B. Girshick, A. Farhadi // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). — 2016. — P. 779 — 788.

29. Бьерн Страуструп. Язык программирования C++. Специальное издание. — М.: ДМК Пресс, 2011. — 1136 с.

30. Коэльё Л. Построение систем машинного обучения на языке Python. / Л. Коэльё, В. Ричерт — Перевод с английского. — М.: ДМК Пресс, 2015. — 364 с.

31. Маккинли У. Python и анализ данных. — Перевод с английского. — М.: ДМК Пресс, 2015. — 482 с.
32. OpenCV — библиотека компьютерного зрения с открытым исходным кодом [Электронный ресурс] — Режим доступа: <http://software.intel.com/en-us/articles/>.
33. Середовище PyCharm. [Электронный ресурс] — Режим доступа: <https://uk.wikipedia.org/wiki/PyCharm>.
34. Мировой рынок камер видеонаблюдения (CCTV). [Электронный ресурс] — Режим доступа: <https://mediasat.info/2020/09/07/global-market-videonabludenie-2027/>.
35. Аналітичні огляди Форекс. [Электронный ресурс] — Режим доступа: [https://www.instaforex.com/ua/video\\_analysis/311142](https://www.instaforex.com/ua/video_analysis/311142).
36. Засоби фірми Hikvision [Электронный ресурс] — Режим доступа: <https://hikvision.org.ua/ua/node/1009>.
37. Камери відеоспостереження. [Электронный ресурс] — Режим доступа: [https://відеокамери.com.ua/videocamera/external\\_cameras](https://відеокамери.com.ua/videocamera/external_cameras).
38. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт / Уклад.: В. О. Козловський, О. Й. Лесько, В. В. Кавецький. — Вінниця: ВНТУ, 2021. — 42 с.
39. Кавецький В. В. Економічне обґрунтування інноваційних рішень: практикум / В. В. Кавецький, В. О. Козловський, І. В. Причепа — Вінниця: ВНТУ, 2016. — 113 с.
40. Тарифи на електроенергію [Электронный ресурс]. Режим доступа: <http://index.minfin.com.ua/tarif/electric.php>.

## ДОДАТОК А

Міністерство освіти та науки України  
Вінницький національний технічний університет  
Факультет інформаційних технологій та комп'ютерної інженерії

### ЗАТВЕРДЖУЮ

Завідувач кафедри ОТ ВНТУ

д.т.н., проф.

\_\_\_\_\_ О. Д. Азаров

“ \_\_\_ ” \_\_\_\_\_ 2022 р.

### ТЕХНІЧНЕ ЗАВДАННЯ

на виконання магістерської кваліфікаційної роботи  
«Програмний засіб виділення контурів рухомих об'єктів в системах  
комп'ютерного відеоспостереження»

08-23.МКР.003.00.000 ТЗ

Науковий керівник к.т.н., доц. каф. ОТ  
\_\_\_\_\_ Колесник І. С.

Студент групи КІ-20мз  
\_\_\_\_\_ Лавренюк В. С.

Вінниця 2022



## 1 Підстава для виконання магістерської кваліфікаційної роботи (МКР)

1.1 Актуальність даного дослідження визначається необхідністю вирішення основних проблем існуючих засобів виділення контурів рухомих об'єктів та відстеження її пересування у зоні відео спостереження для комп'ютерних систем. Особливо актуальним дане дослідження є у зв'язку з тенденціями до впровадження таких засобів у різноманітних засобах для пошуку та виділення контурів рухомих об'єктів.

1.2 Наказ про затвердження теми магістерської кваліфікаційної роботи.

## 2 Мета і призначення МКР

2.1 Мета роботи полягає у підвищенні ефективності виділення контурів рухомих об'єктів у відео потоці та відстеження її пересування. Підвищення ефективності способу виділення контурів рухомих об'єктів заключається в більш точній перевірці відповідності знайдених об'єктів на зображенні заданим.

2.2 Призначення розробки — виконання магістерської кваліфікаційної роботи із подальшим впровадженням та розвитком продукту.

## 3 Вихідні дані для виконання МКР

— запропонувати нові підходи для реалізації методу виділення контурів рухомих об'єктів;

— кольорове цифрове зображення з розширенням не менше за 1024 пікселів по горизонталі та 768 пікселів по вертикалі, розміром не більше 2Мб;

— частота надходження відеокадрів не менше 25 кадрів/сек.

## 4 Вимоги до виконання МКР

Запропонувати нові підходи для реалізації методу пошуку та виділення контурів рухомих об'єктів у отриманій сцені відео спостереження.

Розробити послідовність для пошуку та виділення контурів рухомих об'єктів у отриманій сцені відео спостереження.

Виконати розробку програмного забезпечення для виділення контурів рухомих об'єктів, провести його тестування. Схеми послідовності виділення контурів рухомих об'єктів та лістинги програми представити в додатках до роботи.

5 Етапи МКР та очікувані результати приведені у таблиці А.1.

Таблиця А.1 — Етапи виконання роботи

№ етапу	Назва етапу	Термін виконання		Очікувані результати
		початок	кінець	
1	Аналіз завдання. Вступ	25.03.22	09.03.22	Вступ
2	Аналіз способів виділення контурів рухомих об'єктів	10.03.22	28.03.22	розділ 1
3	Розробка структури комп'ютерної системи та способу виділення контурів рухомих об'єктів	29.03.22	21.04.22	Розділ 2, розробка способу
4	Розробка програми, проектування комп'ютерної системи	22.04.22	10.05.22	Розділ 3, розробка програми
5	Практична реалізація, результати	11.05.22	16.05.22	Розділ 4
6	Розробка економічної частини	17.05.22	31.05.22	Розділ 5
7	Оформлення пояснювальної записки	01.06.22	09.06.22	ПЗ, презентація

## 6 Матеріали, що подаються до захисту МКР

Пояснювальна записка МКР, графічні і ілюстративні матеріали, протокол попереднього захисту МКР на кафедрі, відзив наукового керівника, відзив рецензента, протоколи складання державних екзаменів, анотації до МКР українською та іноземною мовами.

## 7 Порядок контролю виконання та захисту МКР

Виконання етапів розрахункової та графічної документації МКР контролюється науковим керівником згідно зі встановленими термінами. Захист МКР відбувається на засіданні Державної екзаменаційної комісії, затвердженою наказом ректора.

## 8 Вимоги до оформлення МКР

Вимоги викладені в «Положенні про порядок підготовки магістрів у Вінницькому національному технічному університеті» з урахуванням змін, що подані у бюлетені ВАК України № 9-10, 2011р., «Положення про кваліфікаційні роботи на другому (магістерському) рівні вищої освіти, Вінниця, 2021» а також в МЕТОДИЧНИХ ВКАЗІВКАХ до дипломного проектування, ДСТУ 3008:2015, ДСТУ 3974-2000 «Правила виконання дослідно-конструкторських робіт. Загальні положення» та діючого ГОСТ 2.114-95 ЄСКД.

9 Вимоги щодо технічного захисту інформації в МКР з обмеженим доступом відсутні.

Технічне завдання до виконання отримав \_\_\_\_\_ Лавренюк В. С.

## ДОДАТОК Б

## Лістинг програми

```

import numpy as np
import cv2
import VideoStream
import yolov3.config
import yolov3.utils
import dlib
import datetime
from yolov3.model import YOLO3
from pathlib import Path
video = cv2.VideoCapture("vtest.avi")
ret, frame1 = video.read()
ret, frame2 = video.read()
while video.isOpened():
    difference = cv2.absdiff(frame1, frame2)
    rgb_image = frame[:, :, :-1]
    gray = cv2.cvtColor(difference, cv2.COLOR_BGR2GRAY)
    blur = cv2.GaussianBlur(gray, (5,5), 0)
    threshold = cv2.threshold(blur, 20, 255, cv2.THRESH_BINARY)
    python convert.py = yolov3.cfg yolov3.weights model_data/yolo.h5
    model = YOLO3 (mode="inference",
    model_dir=MODEL_DIR,
    config= YOLO3Config())
    model.load_weights(YOLO3_MODEL_PATH, by_name=True)
    results = model.detect([rgb_image], verbose=0)
        height, width, depth = image_to_process.shape
        blob = cv2.dnn.blobFromImage(image_to_process, 1 / 255, (608, 608), (0, 0,
0), swapRB=True, crop=False)

```

```

net.setInput(blob)
outs = net.forward(out_layers)
class_indexes, class_scores, boxes = ([[] for i in range(3)])
objects_count = 0
for out in outs:
    for obj in out:
        scores = obj[5:]
        class_index = np.argmax(scores)
        class_score = scores[class_index]
        if class_score > 0:
            center_x = int(obj[0] * width)
            center_y = int(obj[1] * height)
            obj_width = int(obj[2] * width)
            obj_height = int(obj[3] * height)
            box = [center_x - obj_width // 2, center_y - obj_height // 2,
obj_width, obj_height]
            boxes.append(box)
            class_indexes.append(class_index)
            class_scores.append(float(class_score))
chosen_boxes = cv2.dnn.NMSBoxes(boxes, class_scores, 0.0, 0.4)
for box_index in chosen_boxes:
    box_index = box_index[0]
    box = boxes[box_index]
    class_index = class_indexes[box_index]
    if classes[class_index] in classes_to_look_for:
        objects_count += 1
        image_to_process = draw_object_bounding_box(image_to_process,
class_index, box)
    final_image = draw_object_count(image_to_process, objects_count)
return final_image

```

```

cv2.namedWindow(«YOLO3_Deep_SORT», 0);
cv2.resizeWindow('YOLO3_Deep_SORT', 1024, 768);
cv2.imshow('YOLO3_Deep_SORT', frame)
def draw_object_bounding_box(image_to_process, index, box):
    x, y, w, h = box
    start = (x, y)
    end = (x + w, y + h)
    color = (0, 255, 0)
    width = 2
    final_image = cv2.rectangle(image_to_process, start, end, color, width)
    start = (x, y - 10)
    font_size = 1
    font = cv2.FONT_HERSHEY_SIMPLEX
    width = 2
    text = classes[index]
    final_image = cv2.putText(final_image, text, start, font, font_size, color,
width, cv2.LINE_AA)
    return final_image
video_capture = cv2.VideoCapture(0)
known_object_encodings, known_object_names = Load Objects()
while True:
    ret, frame = video_capture.read()
    rgb_frame = frame[:, :, ::-1]
    object_locations = object_recognition. object_locations(rgb_frame)
    object_encodings = object_recognition. object_encodings(rgb_frame,
object_locations)
    for (top, right, bottom, left object_encoding in zip(object_locations, object
_encodings):
        matches = object_recognition.compare_ object_s(known_ object
_encodings, object_encoding)

```

```

name = "Unknown"
object_distances = object_recognition.object_distance(known_object
_encodings, object_encoding)
best_match_index = np.argmin(object_distances)
if matches[best_match_index]:
    name = known_object_names[best_match_index]
cv2.rectangle(frame, (left, top), (right, bottom), (0, 0, 255), 1)
cv2.rectangle(frame, (left, bottom - 25), (right, bottom), (0, 0, 255),
cv2.FILLED)
font = cv2.FONT_HERSHEY_COMPLEX_SMALL
object_boxes = get_object_boxes(r['rois'], r['class_ids'])
print("Object found in frame of video:")
for box in object_boxes:
    print("Object:", box)
    y1, x1, y2, x2 = box
    cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 255, 0), 1)
    cv2.imshow('Video', frame)

```

## ДОДАТОК В

### Послідовність виділення рухомих об'єктів



Рисунок В.1 — Послідовність виділення рухомих об'єктів



## ДОДАТОК Г

## Методи виділення контурів об'єктів

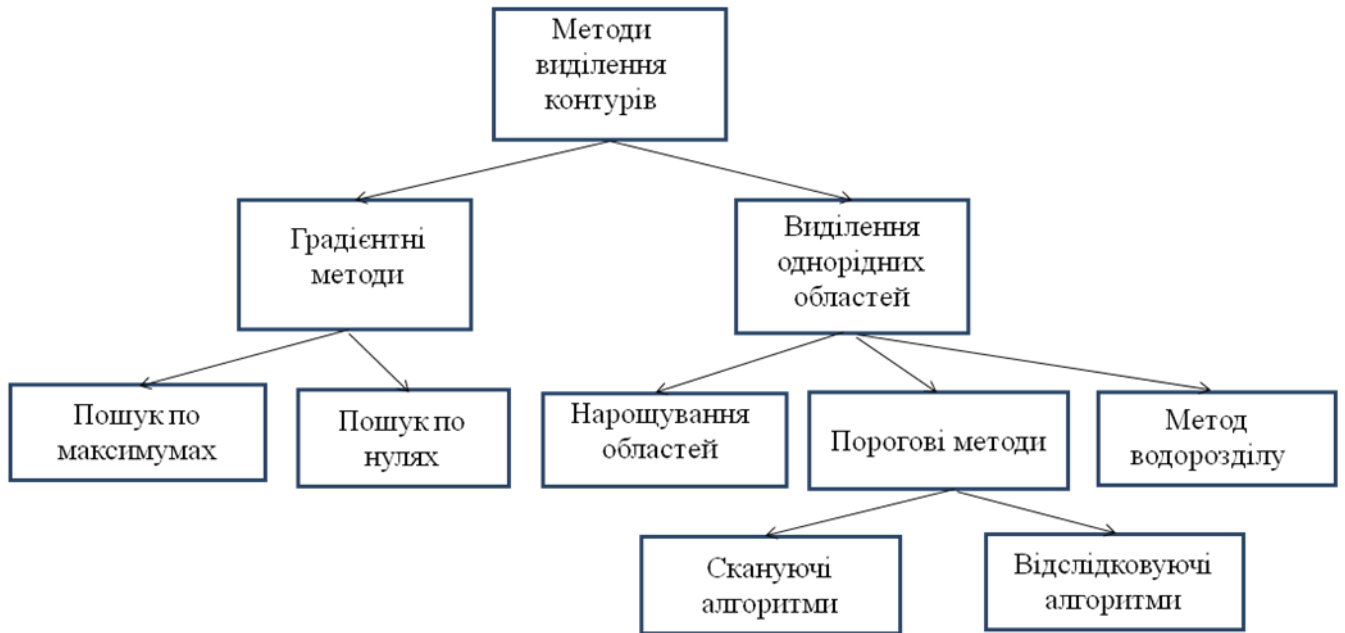


Рисунок Г.1 — Методи виділення контурів об'єктів

## ДОДАТОК Д

### Методи виділення руху

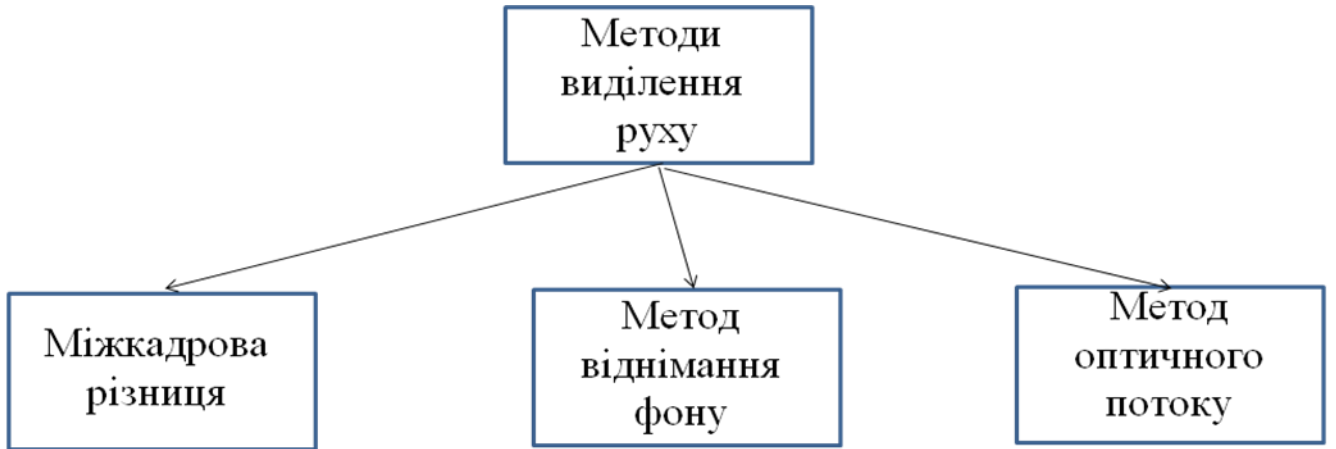


Рисунок Д.1 — Методи виділення руху

## ДОДАТОК Е

Структурна схема системи комп'ютерного відеоспостереження

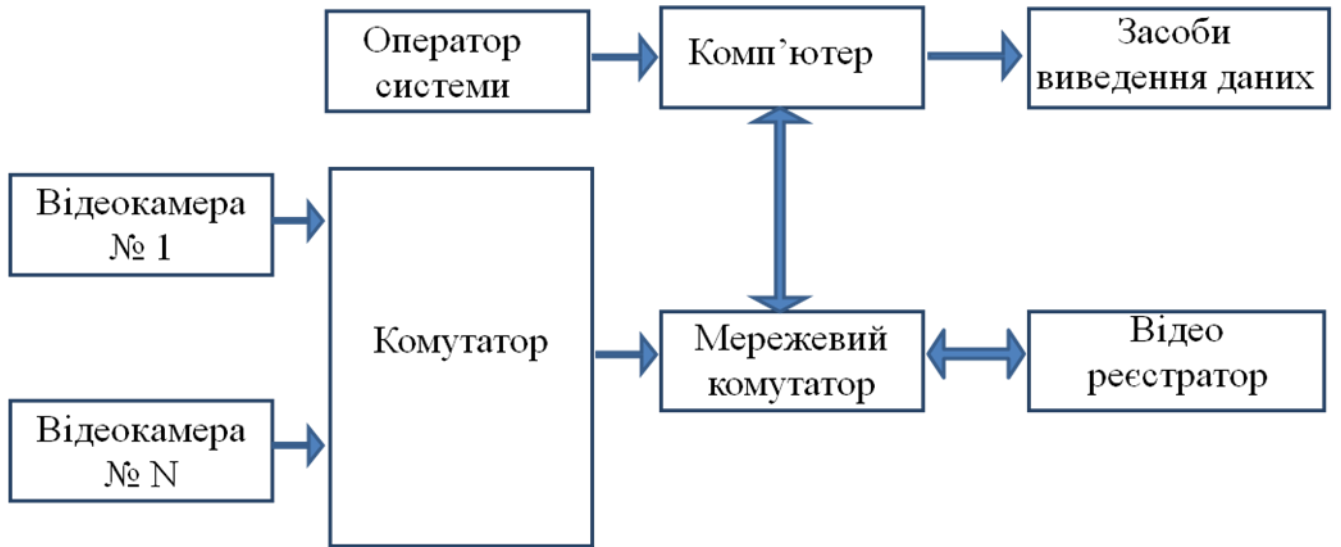


Рисунок Е.1 — Структурна схема системи комп'ютерного відеоспостереження

**ДОДАТОК Ж**

Структурна схема програми виділення контурів об'єктів

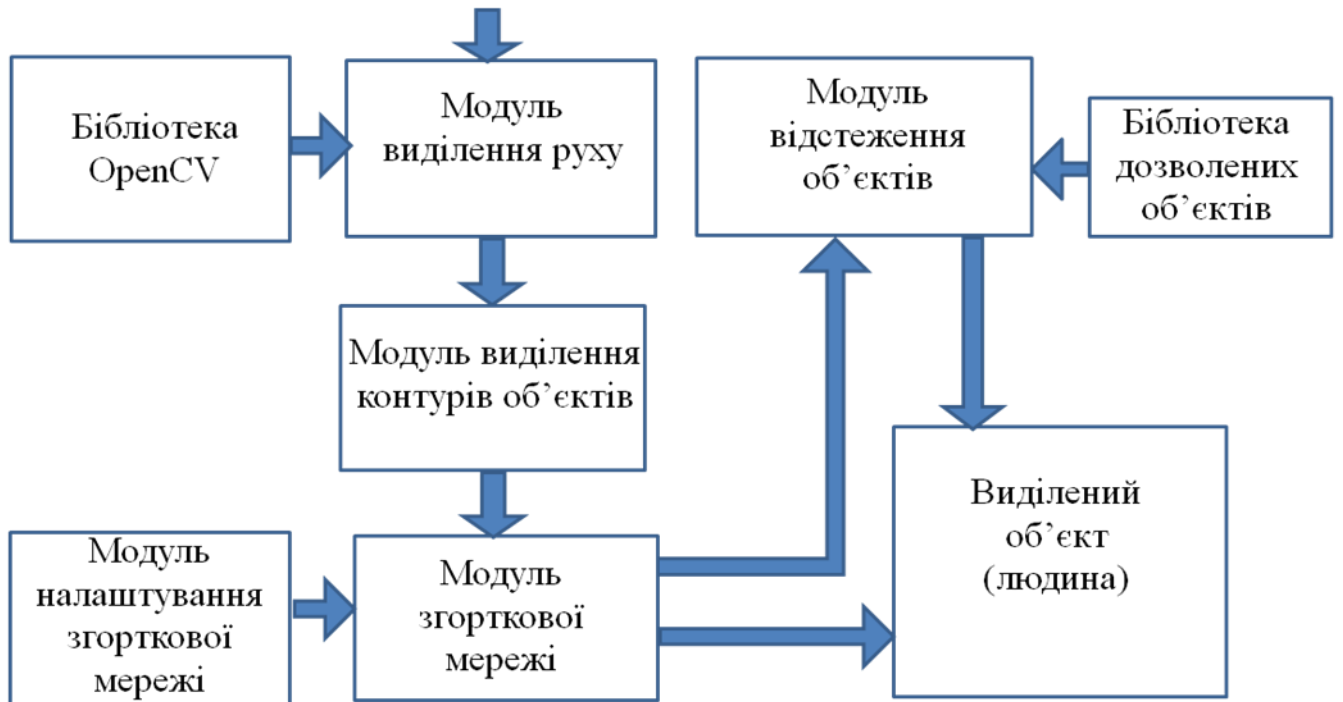


Рисунок Ж.1 — Структурна схема програми виділення контурів об'єктів

**ДОДАТОК И**  
**ПРОТОКОЛ**  
**ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ НА НАЯВНІСТЬ**  
**ТЕКСТОВИХ ЗАПОЗИЧЕНЬ**

Назва роботи: Програмний засіб виділення контурів рухомих об'єктів в системах комп'ютерного відеоспостереження

Тип роботи: магістерська кваліфікаційна робота  
(БДР,МКР)

Підрозділ кафедра обчислювальної техніки  
(кафедра,факультет)

Показники звіту подібності Unicheck

Оригінальність 94,6 Схожість 5,4.

Аналіз звіту подібності (відмітити потрібне):

Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.

Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її виконання автором. Роботу направити на розгляд експертної комісії кафедри.

Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Особа, відповідальна за перевірку \_\_\_\_\_ Захарченко С.М.  
(підпис) (прізвище,ініціали)

Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck щодо роботи.

Автор роботи \_\_\_\_\_ Лавренюк В. С.  
(підпис) (прізвище,ініціали)

Керівник роботи \_\_\_\_\_ Колесник І. С.  
(підпис) (прізвище,ініціали)