

Вінницький національний технічний університет
(повне найменування вищого навчального закладу)

Факультет інформаційних технологій та комп'ютерної інженерії
(повне найменування інституту)

Кафедра обчислювальної техніки
(повна назва кафедри)

Пояснювальна записка

до магістерської кваліфікаційної роботи

магістра

(освітньо-кваліфікаційний рівень)

на тему: «Метод виділення рухомих об'єктів цифрових зображень шляхом пошуку точок інтересу»

Виконав: студент 2 курсу, групи 1КІ-18м

напряму підготовки (спеціальності)

123 – «Комп'ютерна інженерія»

(шифр і назва напряму підготовки, спеціальності)

Шевчук О. В.

(прізвище та ініціали)

Керівник к.т.н., доц. Черняк О. І.

(прізвище та ініціали)

Рецензент к.т.н., доц. Поплавський А. В.

(прізвище та ініціали)

м. Вінниця - 2019року

АНОТАЦІЯ

Дана магістерська кваліфікаційна робота присвячена темі дослідження методів виділення рухомих об'єктів цифрових зображень шляхом пошуку точок інтересу.

В магістерській кваліфікаційній роботі зроблено аналіз методів виділення рухомих об'єктів зображення та виконано огляд способів побудови систем з використанням цих методів, вдосконалено метод виділення рухомих об'єктів зображення із використанням пошуку точок інтересу розроблено алгоритм по обробці цифрового зображення, а також розроблена програма виділення рухомих об'єктів зображення.

В магістерській роботі також розглянуті питання економічної доцільності розробки.

ANNOTATION

This master's thesis deals with the topic of exploring the methods of moving digital objects by searching for points of interest.

In the master's qualification work the methods of selection of moving image objects were analyzed and the methods of constructing systems using these methods were reviewed, the method of allocation of moving objects of objects using search of points of interest was developed, the algorithm for processing of digital images was developed, and the program of selection of moving ones was developed image objects.

The master's thesis also addresses the economic feasibility of development.

ЗМІСТ

ВСТУП.....	8
1 АНАЛІЗ МЕТОДІВ ТА ЗАСОБІВ ВИДІЛЕННЯ РУХОМИХ ОБ'ЄКТІВ.....	12
1.1 Комп'ютерний зір як засіб виділення об'єктів	12
1.2 Траєкторія об'єкта в відеопотоці	16
1.3 Способи пошуку та виділення об'єктів	18
1.4 Етапи формування пошуку об'єктів	25
1.5 Виділення рухомих об'єктів у відеоряді	27
1.6 Нейромережа глибокого навчання	34
1.7 Висновок	40
2 РОЗРОБКА МЕТОДУ І АЛГОРИТМУ ВИДІЛЕННЯ РУХОМИХ ОБ'ЄКТІВ.....	41
2.1 Відстеження об'єктів.....	41
2.2 Механізм створення траєкторії.....	42
2.3 Використання нейронної мережі для виділення об'єктів.....	44
2.4 Алгоритм виявлення об'єктів	47
2.5 Виділення об'єктів нейромережею MobileNet.....	50
2.6 Висновок	55
3 РОЗРОБКА ПРОГРАМНИХ ЗАСОБІВ ВИДІЛЕННЯ РУХОМИХ ОБ'ЄКТІВ.....	56
3.1 Вибір інструментальних засобів для розробки програмного продукту	56
3.2 Вибір середовища розробки.....	58
3.3 Розробка програмного продукту	59
3.4 Перевірка працездатності.....	66
3.5 Висновок	66

					08 - 23.МКР.017.00.000 ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Шевчук О. В.			Метод виділення рухомих об'єктів цифрових зображень шляхом пошуку точок інтересу. Пояснювальна записка	Літ.	Арк.	Акрушів
Перевір.		Черняк О. І.					6	
Реценз.		Поплавський А. В.				ВНТУ, 1КІ – 18М		
Н. Контр.		Швець С. І.						
Затверд.		Маргинюк Т. Б.						

4 РОЗРАХУНОК ЕКОНОМІЧНОЇ ДОЦІЛЬНОСТІ СТВОРЕННЯ ПРОГРАМИ ВИДІЛЕННЯ РУХОМИХ ОБ'ЄКТІВ ЦИФРОВОГО ЗОБРАЖЕННЯ.....	67
4.1 Технологічний аудит розробки.....	68
4.2 Прогнозування витрат на виконання та впровадження результатів наукової роботи.....	72
4.3 Прогнозування комерційних ефектів від реалізації результатів розробки.....	77
4.4 Розрахунок ефективності вкладених інвестицій та періоду їх окупності.....	78
4.5 Висновок.....	81
ВИСНОВКИ.....	83
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	85
ДОДАТКИ.....	89

ВСТУП

Побудова систем пошуку, виділення та розпізнавання об'єктів із використанням засобів комп'ютерного зору – актуальна задача сьогодення. Рішення у цій області знаходять широке застосування у найрізноманітніших сферах життя людини: відстеження появи машин на стоянці, пошук забутого багажу в приміщеннях аеропортів та вокзалів, забезпечення захисту від несанкціонованого проникнення на певні об'єкти, розпізнавання обличч людей, автомобільних номерів, написів, тощо. Етап розпізнавання об'єктів є одним із найважливіших у загальній послідовності вирішення задач по ідентифікації об'єктів. Ця задача, із якою людина легко вправляється кожного дня, є досить складною для обчислювальної техніки. Знаходження відповідності між двома зображеннями деякої сцени чи об'єкту є складовою частиною багатьох систем обробки та аналізу візуальної інформації. Впровадження методів комп'ютерного зору у виконання цих дій дозволяє автоматизувати процеси, що пов'язані із обробкою візуальної інформації.

Одним з важливих та актуальних завдань комп'ютерного зору є локалізація візуальних об'єктів у цифрових зображеннях. Завданням локалізації є визначення області у цифровому зображенні, у якій розташований шуканий візуальний об'єкт. Виконання операції локалізації шуканих об'єктів у цифрових зображеннях дозволяє вирішити ряд проблем, серед яких автоматизація пошуку і сортування різних деталей в промисловості, розпізнавання осіб, візуальна навігація, удосконалення систем автоматичного водіння та автопілотів літальних апаратів, а також системи відеоспостереження.

Пошук та виявлення об'єктів, що переміщуються у просторі, є важливою задачею у сфері комп'ютерного зору. Цей процес включає в себе отримання цифрового зображення, його попередню обробку, аналіз та подальше розпізнавання об'єктів у зображеннях, використовуючи статистичні методи та моделі, що створені за допомогою геометричних, фізичних та статистичних особливостей об'єктів, що аналізуються [1-3]. Широке поширення потужних

комп'ютерів, доступність високоякісних камер за невеликою ціною та збільшення потреб для автоматизованого аналізу відео зумовило значний інтерес у області алгоритмів пошуку, виявлення та супроводження рухомих об'єктів.

У аналізі відеопослідовностей існують три основні етапи: виділення та розпізнавання об'єкта, що переміщається, відстеження виділеного об'єкта кадр за кадром, та аналіз виділених об'єктів для визначення та прогнозування їх поведінки у часі [4]. Використання алгоритму стеження за об'єктом використовується у багатьох задачах, таких як: розпізнавання руху людини за ходою, автоматизоване спостереження, що виявляє підозрілу активність; навігація машин, що пов'язана із плануванням маршруту та уникненням перешкод; взаємодія людина-комп'ютер для розпізнавання жестів, слідкування за поглядом людини для введення даних тощо.

У найпростішій формі відстеження може бути визначено як проблема оцінки траєкторії переміщення об'єкта у площині зображення, що рухається навколо спостережуваної сцени. Для цього у засобах відстеження об'єкт, що рухається, помічаються постійним ярликом. Залежно від області, у якій використовується процес відстеження, засоби відстеження руху можуть надавати додаткову інформацію щодо об'єкта: орієнтацію, розмір, площу, форму. Процес пошуку, виділення і відстеження переміщення об'єктів є досить складною задачею через вплив таких факторів: завади у зображеннях, нестача інформації через проекцію трьохвимірного зображення на двовимірну площину, складні рухи об'єктів, зміна освітлення сцени, потреба у визначенні об'єктів у масштабі реального часу [5, 6]. При вирішенні задачі розпізнавання об'єкту однією із основних проблем є знаходження компромісу між якістю розпізнавання та швидкістю його виконання. Тому задача пошуку та виявлення рухомих об'єктів у виділеній сцені потребує подальшого вдосконалення.

Отже, виходячи із розглянутого, задачі подальшого розроблення та вдосконалення методів пошуку, виявлення та виділення рухомих об'єктів у

цифрових зображеннях, що є кращими за критеріями достовірність та швидкість, на теперішній час є **актуальними**.

Метою дослідження є вдосконалення методів пошуку та виділення рухомих об'єктів у зображеннях із використанням знаходження характерних областей об'єктів у вигляді точок інтересу.

Задачі дослідження: 1) виконати аналіз існуючих методів обробки цифрових зображень для пошуку та виділення рухомих об'єктів; 2) запропонувати поліпшений метод обробки цифрових зображень для пошуку та виділення рухомих об'єктів; 3) створити алгоритм та розробити програму обробки цифрового зображення для виділення рухомих об'єктів шляхом пошуку точок інтересу; 4) здійснити обґрунтування доцільності здійснення нового наукового рішення, виконати розрахунок економічних затрат для створення програмних засобів виділення рухомих об'єктів цифрових зображень та визначити економічні переваги від впровадження нового розробленого програмного продукту.

Об'єкт дослідження – процес обробки даних про об'єкти зображення шляхом пошуку та виділення точок інтересу у аналізованій сцені для знаходження рухомого об'єкта.

Предмет дослідження – методи обробки цифрових зображень для пошуку та виділення рухомих об'єктів.

Методи дослідження: використовувались методи диференційного числення для знаходження екстремумів функції, методи дискретної математики для обробки матриць, що описують зображення, методи математичної статистики для оцінки точності розпізнавання об'єктів, методи теорії множин для опису процесу виділення рухомих об'єктів. Використано засади об'єктно-орієнтованого програмування для реалізації запропонованого підходу по пошуку та виділенню рухомих об'єктів у цифрових зображеннях.

Наукова новизна одержаних результатів полягає у тому, що:

– удосконалено метод обробки цифрових зображень для виділення рухомих об'єктів, який відрізняється від існуючих виконанням пошуку та

виділення об'єктів у вибраній сцені цифрового зображення із використанням характерних особливостей шуканих об'єктів у вигляді точок інтересу, що дозволяє із більшою ефективністю проводити виділення рухомих об'єктів цифрових зображень.

Практичне значення одержаних результатів:

– створено алгоритм обробки цифрових зображень для виділення рухомих об'єктів шляхом знаходження точок інтересу;

– розроблено програму обробки цифрових зображень для виділення рухомих об'єктів із урахуванням характерних особливостей об'єктів та знаходження точок інтересу.

Апробація результатів магістерської роботи: зроблено доповідь на молодіжній науково-практичній інтернет-конференції студентів, аспірантів та молодих науковців «*Молодь в науці: дослідження, проблеми, перспективи (МН-2019)*».

Публікації [7]:Шевчук О. В. Виділення рухомих об'єктів цифрових зображень шляхом пошуку точок інтересу.*Матеріали конференції «Молодь в науці: дослідження, проблеми, перспективи (МН-2019)»*. Збірник матеріалів. - Вінниця, ВНТУ, 2019. - с. 15-16. - [Електронний ресурс]. Режим доступу https://conferences.vntu.edu.ua/public/files/mn/mn-2019_netpub.pdf. Дата звернення: листопад, 2019.

1 АНАЛІЗ МЕТОДІВ ТА ЗАСОБІВ ВИДІЛЕННЯ РУХОМИХ ОБ'ЄКТІВ

Виділення рухомих об'єктів у межах сцени отриманого зображення ґрунтується на пошуку окремих пікселів або груп пікселів, які за вибраними параметрами відрізняються на сусідніх відеокадрах. Розгляду та аналізу існуючих методів виділення рухомих об'єктів у відео послідовності присвячений даний розділ.

1.1 Комп'ютерний зір як засіб виділення об'єктів

Комп'ютерний зір – це низка наукових та технічних складових, за допомогою яких створюються комп'ютерні системи для виявлення, класифікації та стеження об'єктів. Комп'ютерний зір відноситься до теорії та технології побудови штучних систем, які отримують дані у вигляді зображення, що можуть бути представлені у формі відеопослідовності, зображення із різних камер спостереження або веб-камер. Однією із основних сфер застосування комп'ютерного зору є машинний зір.

Областю використання машинного зору як напрямку комп'ютерного зору є цифрові пристрої введення/виведення даних, що призначені для контролю за виробничим процесом, а також спостереження через камери відеонагляду. Системи машинного зору використовують камери відеоспостереження, а також відповідне програмне забезпечення для обробки зображення. Як правило, системи машинного зору побудовані на виконання вузькоспеціалізованих задач, таких як виявлення та підрахунок об'єктів, виявлення переміщення об'єктів у просторі, зчитування серійних номерів або виявлення дефектів деталей. Головна користь від систем машинного зору полягає у можливості безперервної та високої швидкості роботи та необхідної точності вимірювань.

Машинний зір — це процес використання комп'ютерного зору у промисловому виробництві, який складається із апаратних та програмних компонентів, що часто використовуються. Типова побудова системи машинного зору складається із таких компонентів:

- 1) одна або декілька цифрових або аналогових відеокамер (кольорові або чорно-білі) із відповідним оптичним забезпеченням для отримання зображення;
- 2) програмне забезпечення для оброблення отриманих зображень. Для аналогових камер це ще і завдання переведення зображення у цифрову форму;
- 3) процесор (сучасний персональний комп'ютер із багатоядерним процесором або вбудований процесор);
- 4) спеціальне програмне забезпечення засобів машинного зору, яке надає інструменти для створення окремих застосунків програмних продуктів;
- 5) датчик для синхронізації роботи по виявленню, захопленню та обробці зображення (часто це оптичний або магнітний датчик).

У підсумку отримуємо машинний зір як частину комп'ютерного зору у складі комп'ютерної системи.

Комп'ютерна система – це інформаційно-технічний комплекс, метою якого є оброблення, збереження та введення-виведення інформації. До складу комп'ютерної системи входять комп'ютери, пристрої введення-виведення інформації, сервери із відповідним програмним забезпеченням. Комп'ютерні системи можуть об'єднуватися у комп'ютерну мережу. За допомогою локальної або глобальної системи передачі даних через комп'ютерну мережу здійснюється обмін інформацією. Для опису таких систем використовують технічні, документальні, організаційні, алгоритмічні, функціональні, програмні та інформаційні структури. Комп'ютерну систему можна інтегрувати із іншими інженерними технологіями для розширення можливостей та створювати єдине середовище для управління різними процесами завдяки різноманіттю та значній уніфікації комп'ютерного устаткування.

Комп'ютерна система складається із таких компонентів:

- 1) апаратура комп'ютера (hardware), основними частиною якої є центральний процесор (CPU - CentralProcessorUnit), що виконує команди комп'ютера; пам'ять комп'ютера (memory), що зберігає програми та дані, та пристрої введення-виведення або так звані зовнішні пристрої (I / O devices, input-output devices), що забезпечують введення даних у комп'ютер та виведення

результатів роботи програм комп'ютера у формі, яка сприймається користувачем (людиною або іншими програмами);

2) операційна система (operatingsystem) - основний засіб для управління, та системне програмне забезпечення, що розподіляє керування процесом використання апаратури комп'ютера користувачами та різними програмами;

3) прикладне програмне забезпечення (applicationssoftware) – це програми, що призначені для вирішення різних типів задач. До них належать компілятори; графічні бібліотеки, системи управління базами даних; ігрові програми, різні офісні програми. Порівняно із операційною системою прикладне програмне забезпечення створює наступний більш високий рівень та дозволяє вирішувати на комп'ютері різні прикладні завдання.

Поставлене завдання пошуку та виявлення руху об'єктів у відео послідовності і буде одним із таких прикладних програмних продуктів. Створення такого продукту є однією із задач створення системи розпізнавання образів.

Системи розпізнавання образів за методами та засобами пошуку об'єктів поділяють на:

1) Детерміновані системи, у яких для побудови алгоритмів розпізнавання образів використовують «геометричні» міри близькості. У таких системах відбувається вимірювання відстаней між кількісними ознаками об'єктів та класових еталонів.

2) Ймовірнісні системи, у яких для побудови алгоритмів розпізнавання образів використовують ймовірнісні методи у межах застосування теорії статистики. Для функціонування такої системи розпізнавання образів потрібно мати ймовірнісні залежності між вибраними ознаками об'єктів та їх класами.

3) Логічні системи, що ґрунтуються на дискретному аналізі та обчисленні логічних висловлювань. У таких системах будуються на вирішенні булевих рівнянь, для цього використовуються логічні ознаки об'єктів та необхідно знайти класи, до яких слід віднести такі об'єкти.

4) Нейроноподібні системи, що використовують для розпізнавання образів у своїй побудові моделі нейронів.

5) Структурні або лінгвістичні системи розпізнавання образів призначаються для побудови таких алгоритмів розпізнавання, що використовують при цьому речення. Кожне із таких утворених речень описує структуру об'єкта через певні непохідні елементи. Із таких речень формується свого роду спеціальна мова. У такій системі класифікація об'єктів здійснюється шляхом порівняння речень невідомих об'єктів із уже створеними еталонними реченнями у класах.

6) Експертні системи розпізнавання образів. Такі системи використовують у тих випадках, коли існують евристичні або інтуїтивні способи та засоби вирішення поставлених задач, але відсутні точні алгоритми чи розрахунки. Експертна система формується із сукупності комп'ютерних програм, які містять накопичені знання значної кількості експертів у заданій предметній області та здатна у межах такої області виконувати класифікацію об'єктів та формувати відповіді, поради або рекомендації, за необхідністю запитуючи про аналізовані об'єкти додаткову інформацію.

По взаємозв'язку до навчання систем розпізнавання образів можна поділити на такі типи [4, 10]:

1) системи без навчання, щонають достатню кількість інформації для побудови апріорного словника та алфавіту ознак класів, а також для створення опису всіх класів мовою цих ознак. Тому достатньо у першому наближенні визначити відстані або кінцеві границі між цими класами ознак та сформулювати вирішальні правила, які дозволяють поділити утворену множину ознак на відповідні класи. Для побудови такої системи розпізнавання потрібно мати повну початкову апріорну інформацію про об'єкти;

2) системи із навчанням. У таких системах розпізнавання образів початкової інформації достатньо тільки для створення апріорного класового алфавіту та апріорного словника ознак, але зовсім недостатньо для опису класів мовою ознак. Наявна початкова інформація дозволяє розділити конкретні

об'єкти до різних класів. «Вчитель» декілька разів представляє системі розпізнавання навчальні образи уже створеного відомого класу. Якщо така система розпізнавання образів формує невірну відповідь, то у системі за відомим алгоритмом змінюються усі коефіцієнти підсилення виділених сигналів до отримання вірної відповіді. Після такої перевірки та навчання система розпізнавання починає працювати самостійно;

3) системи із самонавчанням. Початкової апріорної інформації у цих системах вистачає всього лише на формування словника ознак, але зовсім недостатньо для будь-якої іншої класифікації заданих об'єктів. На початковій стадії формування системі показують навчальний набір об'єктів, які задані значеннями своїх ознак. Система не має змогу отримати при цьому вказівок через недостатній об'єм такої початкової інформації про те, до якого саме класу належать об'єкти цього початкового набору. На стадії самонавчання вказівки замінюються конкретним рядом правил і тоді система розпізнавання образів сама починає створювати класифікацію, якою і керується у подальшому. Метою такого самонавчання є напрацювання деякої кількості інформації, якої було б цілком достатньо, щоб система розпізнавання функціонувала у повному обсязі. Для отримання такого результату потрібно мати зворотній зв'язок, щоб виконувати коригування і уточнення опрацьованої апріорної інформації.

1.2 Траєкторія об'єкта в відеопотоці

Траєкторія руху матеріальної точки - це така лінія у просторі, по якій рухається об'єкт, що представляє собою деяку множину точок, в яких знаходилася, у даний момент перебуває або буде знаходитися вибрана матеріальна точка при своєму переміщенні у просторі відносно обраної системи відліку координат.

Відеопотік - це послідовність кадрів певного формату у часі.

У більшості випадків спостережуваний у відеопотоці об'єкт не є матеріальною точкою. Для знаходження його координат необхідно виділити точку, що найбільш точно відтворює координати об'єкта у даний момент часу.

Координати точки траєкторії руху об'єкта - це деякі лінійні величини, що описують положення об'єкта відносно вибраного початку координат. По відношенню до двомірної траєкторії руху - це координати X , Y та час.

Область об'єкта – це мінімальна частина простору, що цілком охоплює виявлений об'єкт. У тому разі, коли для виділення об'єкта застосовується прямокутна область, то її центром, а також і центром виділеного об'єкта, є точка перетину діагоналей виділеної прямокутної області. Виявленим у сцені об'єктам присвоюється унікальний ідентифікатор. Ця процедура необхідно для вилучення повторного виявлення об'єкта, так як обов'язково необхідно, щоб певному одному об'єкту у відповідність однозначно ставилася одна траєкторія. Координати такого виявленого об'єкту необхідно зберігати в базі даних для подальшого оброблення. У базі даних для зберігання координат виявленого об'єкта потрібні такі стовпці: присвоєний унікальний ідентифікатор знайденого об'єкта, його координати по осях абсцис і ординат та час.

Частота дискретизації (частота семпловання, англ. Samplerate) - частота отримання відліків неперервного у часі сигналу при його дискретизації. В системі обробки траєкторій руху об'єктів необхідно передбачити можливість використання різних частот дискретизації. Вибір частоти дискретизації залежить від властивостей відеопотоку та швидкості переміщення спостережуваних об'єктів. Встановлена частота дискретизації кадрів у відеопотоці повинна забезпечувати безперервність траєкторії руху об'єктів та її необхідної точності. Фактично лінія траєкторії руху не повинна бути занадто сильно «згладженою», так як це може призвести до істотних втрат певних елементів траєкторії руху. Також надмірно висока частота дискретизації відео потоку призводить до суттєвого збільшення використання ресурсів обчислювальної системи та збільшує навантаження на мережу передачі даних. Вказані фактори вкрай негативно позначаються на швидкості роботи засобів обчислювальної системи.

1.3 Способи пошуку та виділення об'єктів

Цифрове зображення може бути описане за допомогою глобального дескриптора, який по суті є вектором ознак цього зображення. При цьому кожна точка цього зображення робить деякий певний внесок у значення таким чином сформованого дескриптора.

Під детекторами особливих або ключових точок в обробці відео розуміються алгоритми пошуку та виділення точок, що є симетричними до різного типу перетворень у зображенні. Такими особливими точками або точками інтересу можуть служити кути, границі або інші особливості аналізованих об'єктів у зображенні.

Особлива точка сцени або ключова точка (pointfeature) це така точка, зображення якої відрізняється від зображення всіх сусідніх із нею точок. Для опису такої точки використовують певну околицю. Тоді особлива точка сцени M це така точка зображення, околиця якої $O(m)$ відрізняється від околиці будь-якої іншої точки зображення $O(n)$ в деякій іншій околиці особливої точки $O_2(m)$. Із цього випливає, що між особливістю сцени та його зображенням є відповідність – точковій особливості сцени має відповідати точкова особливість зображення. Однак обернене твердження не є справедливим: існують такі точки зображення, яким не відповідають ніякі інші особливі точки сцени. Такі точки називається неправильними особливостями сцени (falsefeaturepoint). Головною проблемою при аналізі руху об'єктів стоїть задача відділення таких неправильних точок від дійсних точок сцени при відстеженні руху об'єктів. Ця ситуація ускладнюється тим, що надійно вирішити її при роботі із одним зображенням неможливо. Необхідно знайти відповідність між особливостями декількох зображень, а потім побудувати таку модель, якій ці особливості будуть відповідати найкращим чином. Особливі точки, відповідність яких не буде задовольняти такій моделі, будуть вважатися неправильними або викидами (outlier).

Точка інтересу – це така точка сцени, зображення околиці якої можна відрізнити від зображень околиць інших точок сцени, що розглядається. Головною властивістю точки інтересу аналізованого об'єкта є її інваріантність до змін точки спостереження камери або освітленості. Для кожного об'єкта у

зображенні можна відшукати його точки інтересу і описати ними сам об'єкт. Сформувавши опис об'єкта з деякого тренувального зображення його можна у подальшому використовувати для пошуку такого об'єкту у інших зображеннях. Для кращої надійності роботи такого алгоритму розпізнавання важливою умовою є те, що точки інтересу об'єкта мають знаходитися незалежно від зміни яскравості, масштабу зображення та впливу шумів. При цьому відносне розташування між вибраними точками інтересу об'єкта має залишатися незмінним. Тому такі точки як правило знаходять на межах об'єктів.

При стеженні за переміщеннями об'єктів у відеопотоці із застосуванням точок інтересу необхідно спочатку знайти точки інтересу у першому кадрі відео ряду. Потім визначається якість отриманих точок та відбираються лише ті, якість яких задовольняє певному визначеному порозу. Після цього для кожного наступного кадру відео послідовності знаходиться нове положення точок інтересу, оновлюються дані про їх якість та здійснюється перевірка на виконання умови задоволення заданому порозу якості. Якщо у підсумку такої роботи кількість точок інтересу стає меншою за певний заданий поріг, то шукаються нові точки з поточного кадру та додаються до знайдених раніше точок.

Для знаходження особливих точок у цифровому зображенні, які мають виразні відмінності у їх місцях локалізації, використовується детектор точок. Для цих особливих точок бажано мати постійну якість до зміни інтенсивності навколишнього світла та позиції камери.

За останні десятиріччя було запропоновано величезну кількість різних типів детекторів для пошуку точкових особливостей цифрових зображень. Запропоновані детектори використовують різні підходи до формування функції оцінювання пікселів для пошуку особливостей об'єктів. У нашому випадку у системах стеження за переміщенням об'єктів визначальним параметром якості для вибору детектора буде якість подальшого відстеження об'єкта у зображенні.

Одним з перших детекторів особливих точок був запропонований оператор Моравека [14]. Цей детектор визначає зміни інтенсивності зображення у блоках розміром 4x4 пікселі у горизонтальному, вертикальному, та двох діагональних напрямках та вибирає найменше з чотирьох отриманих значень як характерне значення для даного вікна. Ця точка позначається як точка інтересу, коли відхилення інтенсивності яскравості для неї є локальним максимумом у блоці розміром 12x12 пікселів. Оператор Моравека є одним із найстаріших і дозволяє знаходити кути об'єктів у зображенні. Алгоритм Моравека - це один із перших алгоритмів для пошуку та виявлення кутів. Алгоритм здійснює перевірку кожного пікселя у зображенні, щоб визначити чи є цей піксель тим кутом, аналізуючи околицю у області цього пікселя. Подібність знаходиться шляхом аналізу суми квадратів різниць яскравості між двома ділянками. Менше значення вказує на більшу схожість.

Розглядається три випадки:

– якщо піксель знаходиться у області рівномірної інтенсивності яскравості, то прилеглі ділянки зображення будуть виглядати приблизно однаковими;

– якщо піксель знаходиться на краю можливого кута, то тоді сусідні ділянки у перпендикулярному до краю напрямку будуть виглядати зовсім різними, а сусідні ділянки в паралельному до краю напрямку будуть змінюватися незначно;

– якщо піксель знаходиться у точці, де є зміни в усіх напрямках, то жодна із прилеглих ділянок буде різною та не буде виглядати приблизно так само.

Але у більшості випадків об'єкти, за якими відбувається спостереження, є досить складними, і простої операції знаходження кутів об'єктів не є достатнім для опису вибраного образу та здійснення його подальшої ідентифікації. Крім того, один з недоліків цього методу пов'язаний з тим, що він не є ізотопним: якщо кут не спрямований в бік сусіднього кута, то він не буде виявлений, як особлива точка.

Інший детектор Харріса [15] розглядає цифрове зображення як функцію двох змінних $I(x,y)$. Основна ідея роботи детектора полягає в тому, що знаходиться градієнт зображення для кожного пікселя, наприклад, за допомогою використання операції гауссовського згладжування. Для кожного пікселя цього зображення вираховується значення функції відгуку кута (cornerresponsefunction), яка оцінює ступінь подібності вибраної околиці точки зображення до кута. Для цього на першому етапі розраховується матриця Харріса за такою формулою:

$$M = \begin{vmatrix} \left(\frac{\partial I}{\partial x}\right)^2 & \left(\frac{\partial I}{\partial x}\right) \left(\frac{\partial I}{\partial y}\right) & \left(\frac{\partial I}{\partial y}\right)^2 \\ \left(\frac{\partial I}{\partial x}\right) \left(\frac{\partial I}{\partial y}\right) & \left(\frac{\partial I}{\partial x}\right)^2 & \left(\frac{\partial I}{\partial y}\right)^2 \end{vmatrix} \quad (1.1)$$

де $I(x,y)$ – яскравість зображення у точці (x,y) . Потім розраховується функція відгуку кута за заданою формулою

$$R = \det M - k (trM)^2 \quad (1.2)$$

де k – деяка емпірична константа, яка у багатьох випадках приймається рівною 0,04. Вона була запропонована, так як визначення власних чисел майже неможливо.

Точками інтересу вважаються такі точки зображення, що відповідають локальним максимумам функції відгуку кута. Для кутових точок значення $R > 0$. У значній кількості випадків детектор Харріса знаходить занадто велику кількість кутів, що призводить до ускладнення у їх подальшому відстеженні. Для зменшення цієї проблеми вводять обмеження на мінімальну відстань між знайденими особливими точками і всі решта точок відкидаються за вибраним пороговим значенням. Шукаються локальні максимуми в околиці заданого радіусу, які потім вибираються в якості особливих точок (точок інтересу).

У детекторі Shi-Tomasi [16] використовується та ж сама матриця M , що у детекторі Харріса, але знаходяться мінімальні власні числа матриці напряму. Використовується диференціальний спосіб виявлення точок для обчислення кривизни попередньо масштабованої кривої і знаходяться позитивні максимуми і негативні мінімуми цього аналізованого диференціального виразу у деякому

масштабі t та у масштабі простору L представлення вихідного зображення. Головною проблемою цього способу є те, що він надто чутливий до завад та до вибору масштабного рівня аналізу зображення. У підсумку при великих значеннях масштабу аналізовані кути будуть пов'язані із представленням їх у вигляді закруглених кутів великої просторової протяжності, а при менших значеннях масштабу вони будуть представлені як гострі кути з невеликою просторовою протяжністю. Такий підхід є першим використанням детектора кутів із автоматичним вибором масштабу і він був використаний для виділення кутів при великих зміни масштабу в області аналізованого зображення.

Ще один детектор Blob [17] був створений на основі ЛапласіанГауссіан. У області комп'ютерного зору детектор Blob відноситься до візуальних модулів, які спрямовані на пошук областей, точок або піків зображення, що є більш яскраво вираженими, ніж навколишні точки. Метод лапласіанГауссіан ґрунтується на виконанні одноразової операції смугової фільтрації просторової функції цифрового зображення з застосуванням оператора лапласіангауссіана.

Цей метод створений завдяки властивості комутативності лінійних операторів Гауса і Лапласа із використанням оператора згортки та можливості об'єднання їх в один оператор шляхом зміни послідовності виконання операцій диференціювання та згладжування. Створений таким чином об'єднаний оператор отримав назву оператора ЛапласіангауссіанаLoG (LoG - LaplacianofGaussian) та представляє собою згортку операторів Гауса та Лапласа.

Перевагою такого методу є одноразова обробка цифрового зображення, так як згортка операторів виконується завчасно. Недоліком даного підходу є те, що оператори Гауса та Лапласа є ненаправленими. Метод досить чутливий до зміни яскравості зображення у паралельному напрямку, що значно зменшує співвідношення сигнал/завада.

Для вирішення таких проблем у пошуку та розпізнаванні особливих точок застосовуються дескриптори. Дескриптори у загальному вигляді представляють собою згортку характеристик і подання ключових точок у форматі, який доступний для перевірки на співпадіння.

Поширений дескриптор SIFT (Scale-invariant feature transform), що був запропонований Девідом Лоу [15], дозволяє ефективно виділяти та ідентифікувати об'єкти навіть на сильно зашумлених цифрових зображеннях та при частковому перекритті об'єктів, тому що дескриптор SIFT є інваріантним до зміни положення у просторі та рівномірного масштабування, а також частково інваріантний до зміни освітленості та операції афінного скосу. Дескриптор SIFT дозволяє відшукати точки інтересу та сформувати дескриптори околиць цих знайдених точок. На першому кроці роботи алгоритму SIFT змінюється масштабний простір шляхом використання фільтра Гауса для зображення у різних масштабах. Із отриманих зображень знаходять різницю між гаусіанами. У точках максимуму та мінімуму зображень обираються точки-кандидати із різницями гаусіан.

На наступному етапі положення кожної точки-кандидата оновлюється шляхом інтреполяції кольору по сусіднім пікселям. Потім відкидаються такі точки-кандидати, що мають недостатню контрастність або знаходяться на краях зображення. Для тих точок, що залишилися, знаходиться напрям за градієнтом у невеликому околі. Метод SIFT знаходить значно більшу кількість точок інтересу у порівнянні із іншими відомими детекторами, і розробники цього алгоритму показали емпірично, що він є стійкішим до змін у зображенні.

Одним із найбільших недоліків методу SIFT є те, що він працює лише із зображеннями у лише градаціях сірого, відкидаючи при цьому важливу інформацію про аналізоване зображення, яку несе у собі колір зображення. Також була запропонована модифікація методу SIFT, яка дозволяє враховувати кольорові особливості цифрового зображення. На першому етапі роботи модифікованого алгоритму CGSIFT здійснюється квантування кольорового зображення з метою подання його із невеликою кількістю кольорів. А вже на другому етапі застосовується детектор SIFT для знаходження особливостей у такому квантованому зображенні.

Інший дескриптор SURF (SpeededUpRobustFeature) [16] схожий на SIFT, але у декілька разів працює швидше за нього. Також цей дескриптор

інваріантний до повороту об'єкта та масштабу. Пошук та виділення точок інтересу у алгоритмі SURF ґрунтується на обчисленні гессіан.

Матриця Гессе – це матриця, що формується із других часткових похідних функції та задається такою формулою $H(f(x,y))$:

$$H(f(x,y)) = \begin{bmatrix} \frac{d^2 f}{dx^2} & \frac{d^2 f}{dx dy} \\ \frac{d^2 f}{dx dy} & \frac{d^2 f}{dy^2} \end{bmatrix} \quad (1.3)$$

$$\det(H) = \frac{d^2 f}{dx^2} \frac{d^2 f}{dy^2} - \left(\frac{d^2 f}{dx dy} \right)^2 \quad (1.4)$$

Детермінант $\det(H)$ цієї матриці називається Гессіаном. Якщо розглядати зображення як деяку функцію, то у точках, де відбувається максимальна зміна яскравості зображення, Гессіан цієї функції досягає своїх екстремумів. Після виділення ключових точок зображення із використанням гессіана у алгоритмі SURF формуються дескриптори точок. Кожен дескриптор фактично є набором чисел, що відображає зміну градієнту функції біля знайденої ключової (особливої) точки. Для обчислення матриці Гессе застосовується метод бінаризованої апроксимації лапласіанугаусіан.

Кожні методи відстеження переміщення об'єктів потребують певного механізму виділення об'єкта на отриманих кадрах або коли об'єкт вперше появляється на відео. Загальний підхід у виділенні об'єктів – це використання інформації із одиничного кадру. Але деякі існуючі алгоритми можуть використовувати часову інформацію, яка була виділена із послідовності кадрів.

У алгоритмі SUSAN кути визначаються шляхом сегментації кругових околиць у схожі та несхожі ділянки. Кути знаходяться у тих ділянках, де знайдена відносна площа схожих ділянок (similar USAN) знаходиться на локальному мінімумі, який нижчий заданого певного порогу.

Для кожного пікселя зображення розглядається кругова область певного фіксованого радіуса. Центр такого пікселя називається ядром, його значення інтенсивності запам'ятовується. Усі інші пікселі зображення розділяються на дві категорії: у схожі та несхожі ділянки у залежності від того, схоже чи не схоже значення інтенсивності ядра. У тих місцях, де у присутній ділянці

зображення під круговою областю схожі ділянки займають майже всю площу кути будуть відсутні, там де є грані, це відношення зменшується до 50%, та на кутах воно досягає приблизно до 25%. У підсумку такого аналізу кути фіксуються там, де відносна площа схожих ділянок досягає деякого локального мінімуму, що нижчий певного порогу. Для підвищення якості роботи алгоритму найближчим до ядру пікселям надаються вищі вагові коефіцієнти.

Пошук та виділення об'єктів можна досягнути за допомогою побудови деякого представлення сцени, яка ще називається Backgroundsubtraction або backgroundmodel. Для кожного нового кадру шукаються відхилення від заданої моделі. Будь-які суттєві зміни у зоні зображення порівняно із моделлю означають, що виділений об'єкт рухається. Значних поліпшень в побудові моделей є застосування статистичних мультимодальних моделей для опису кольору кожного із пікселів. Для моделювання кольору пікселя також використовують змішування по Гаусу. У цьому методі піксель у даному кадрі перевіряється порівняно із моделлю фону та порівнюється із кожним Гаусіаном у моделі до тих пір, поки потрібний не буде знайдено. Якщо такий знайдено, то значення Гаусіана оновлюється, в іншому випадку додається новий Гаусіан із знайденим значенням кольору поточного пікселя.

1.4 Етапи формування пошуку об'єктів

Відомі стандартні методи пошуку та виділення об'єктів включають: виділення за значенням яскравості; виділення за ознаками текстур; виділення за кольором або ручне виділення.

Найбільш поширеним та простим методом виділення об'єктів, що досить часто використовується, є виділення об'єктів за значенням яскравості. При застосуванні цього методу виділення об'єктів або окремих фаз кадрів здійснюється шляхом відбору тих точок зображення, що потрапляють у певний інтервал значень яскравості. Такий відбір може здійснюватися за гистограмою яскравості для даного цифрового зображення. Цифровим зображенням є набір точок, яскравість яких у 8-ми бітному представленні зображення знаходиться у

інтервалі від 0 (відповідає абсолютно чорному кольору) до 255 (відповідає абсолютно білому кольору). Гістограма яскравості зображення відтворює процентний вміст точок за певним значенням яскравості. Виділення об'єктів за значенням яскравості як правило здійснюється методом встановлення граничних значень яскравості, і всі ті точки зображення, яскравість яких знаходиться у межах заданих значень, будуть виділені у зображенні як маски об'єктів, які потім будуть оброблятися.

Пошук та виділення об'єктів за кольором здійснюється із використанням колірних моделей, цей процес за характером є складнішим, ніж операція простого виділення за яскравістю, але дозволяє отримати кращі результати у тих випадках, коли об'єкти із однаковими характеристиками по яскравості відрізняються по кольору. У сучасних системах аналізу зображень виділення об'єктів зображення за кольором здійснюється із використанням методу навчання, коли користувач системи вказує ті ділянки цифрового зображення, які слід виділяти. Підготовлена таким чином система на наступному цифровому зображенні автоматично виділятиме ті об'єкти, які відповідають за кольором вказаним ділянкам. Колір містить дуже важливу інформацію із точки зору подібності об'єктів, хоча і приймати рішення лише на підставі тільки кольору зазвичай досить важко. Зображення об'єктів реального світу містять багато різних відтінків кольорів, значна частина яких є дуже близькими та подібними, аналізувати всі наявні відтінки кольору немає сенсу, так як пікселі, що належать навіть єдиному об'єкту, дуже часто відрізняються за значеннями кольору, хоча це для людського ока є непомітним.

Пошук та виділення об'єктів за особливостями їх текстур найближче підходить до механізмів ідентифікації об'єктів, які використовує мозок людини при їх розпізнаванні. Цей підхід полягає у розпізнаванні об'єкту або якоїсь фази сцени не за яскравістю або ж колірними характеристиками, а за певними зовнішніми ознаками або особливостями текстури. Це стосується таких ситуацій, коли необхідно виділити певний шар або ділянку із орієнтованою структурою всередині однорідної структури об'єкта, яка не відрізняється від

аналізованого шару ні за параметрами яскравості, ні за параметрами кольору, але відрізняється виключно на вигляд. Такий спосіб виділення об'єктів найбільш складний для своєї реалізації, і не всі існуючі аналізатори цифрових зображень придатні вирішувати таку задачу виділення за ознаками текстур.

Ручний спосіб пошуку та виділення об'єктів або окремих фаз сцени застосовується у тих випадках, коли об'єкти не можна виділити жодним із перерахованих вище підходом. При використанні ручного способу виділення користувач сам вказує системі, що необхідно вважати об'єктом, а що - ні.

Для отримання кількісної інформації про об'єкти зображення за допомогою застосування аналізатора, необхідною початковою умовою є виділення об'єктів щодо фону або сторонніх елементів. Виділення та подальше розпізнавання об'єктів зображення є важливими етапами обробки зображення.

Уже тільки після того, як об'єкти, що представляють цікавість, будуть виділені, тобто тільки після того, коли аналізатор зображення має інформацію, що слід вважати об'єктами, наступають наступні етапи по їх обробці.

1.5 Виділення рухомих об'єктів у відеоряді

Для виділення рухомих об'єктів існує значна кількість різних типів алгоритмів. Здатність цих алгоритмів функціонувати та видавати вірний результат залежить від різних значень параметрів вхідних даних. У зв'язку із такою постановкою задачі виникає необхідність застосовувати не один із алгоритмів для вирішення цього завдання. Тому на поточному етапі роботи по створенню засобів виділення рухомих об'єктів стоїть завдання знайти декілька найбільш ефективних та коректних алгоритмів із ряду існуючих, запропонувати нові підходи та шляхом комбінування із кількох алгоритмів досягти бажаного результату.

Розрізняють такі основні підходи до пошуку та виділення рухомих об'єктів в кадрі.

1)Визначення переміщення областей пікселів між сусідніми кадрами [3, 4]. У цьому випадку виділення рухомих об'єктів ґрунтується на сегментації

поточного та попереднього кадрів, пошуку відповідних областей у зображенні та оцінці у зміні їх розташування або форми. Такий підхід використовується для вирішення завдання супроводу рухомих об'єктів та заснований на апріорному допущенні, що об'єкт, який рухається, є гомогеним за кольором або текстурою та є у достатній мірі контрастним по відношенню до фону зображення. Головними недоліками даного підходу є досить низька стійкість в умовах малої контрастності рухомих об'єктів щодо нерухомого фону та високі обчислювальні затрати при виконанні операції сегментації зображення.

2) Визначення зміщення прямокутних блоків у зображенні [6, 7]. Пошук та виділення рухомих об'єктів у кадрі ґрунтується на знаходженні векторів зсуву виділених прямокутних блоків для пари кадрів. Блоки із ненульовою довжиною векторів зсуву вважаються такими, що містять рухомі об'єкти. Недоліками такого підходу є нестабільність результату роботи для об'єктів із малим відношеннями площі до периметру та для однотонних об'єктів при наявності завад.

Для виконання завдання контролю території із низькою інтенсивністю руху та відносно стабільними умовами відеозйомки ефективним є використання попиксельного визначення зміни кадрів, оскільки такий підхід вимагає малих обчислювальних витрат, не має обмежень на кількість одночасно контрольованих об'єктів, та має відносно малу чутливість до використання різних засобів маскування рухомих об'єктів.

Розрізняють два основні підходи до попиксельного визначення змін у кадрі:

1) Знаходження безпосередньої зміни яскравості пікселів поточного та попереднього кадрів, що аналізуються, де MT – значення порогової величини, яка визначає чутливість цього методу до міжкадрових відмінностей.

Головним недоліком даного способу є неможливість виявлення об'єктів, що повільно рухаються, та швидкість яких у проекції є нижчою, ніж один піксель за час проходження інтервалу між двома сусідніми кадрами. Для вирішення цієї проблеми використовується аналіз декількох попередніх кадрів.

Однак це тягне за собою додаткові затрати пам'яті на зберігання даних та витратити процесорного часу на аналіз таких додаткових кадрів, а також появу у матриці прапорів відмінностей виділеного «сліду» такого руху об'єкта.

2. Знаходження зміни яскравості між поточним кадром та сформованим фоном. Після виконання операції формування матриці прапорів відмінностей на кожному наступному кроці відбувається переформування фону.

При $AF = 1$ або $DF = 255$ процес зводиться до визначення безпосередньої зміни яскравості пікселів між поточним та попереднім кадрами.

Недоліком даного підходу є значний час на відновлення коректного фону після випадкового перекриття поля спостереження камери стороннім предметом, котрий перекрив значну частину області кадру (як приклад, птах, що летить поруч або сусідній рухливий об'єкт), або зміна умов освітленості сцени (наприклад, включення світла у приміщенні). Дана ситуація вирішується за допомогою операції примусового переформатування фону, якщо кількість знайдених відмін Mt перевищує значення деякої порогової величини MDF :

Слід відзначити, що замість значення яскравості пікселя може використовуватися також будь-яка інша норма колірному простору зображення або здійснюватися оцінка руху по кожній із колірних координат RGB із логічним підсумовуванням результату [8]:

Під рухомим об'єктом у відео розуміється такий об'єкт, положення якого змінюються у кожному відеокадрі по відношенню до інших об'єктів сцени. Рухомою ділянкою на відео є частина відеокадру, що охоплює деякий рухомий об'єкт. Відомі методи класифікації рухомих об'єктів у відео послідовності ґрунтуються на методах розпізнавання об'єктів у цифровому зображенні, так як відеодані мають вигляд набору послідовних кадрів відеопослідовності у растровому представленні. Однією з базових вимог до таких методів виділення рухомих об'єктів є швидкодія, оскільки класифікація об'єктів повинна виконуватись в режимі реального часу. Основна вимога виконання таких операцій полягає у тому, щоб отримати бажану простоту реалізації. Під

простотою реалізації розуміється низька обчислювальна складність вибраного методу по відношенню до реалізації у вибраному програмному забезпеченні.

У сучасних системах відеомоніторингу, в яких застосовуються алгоритми виділення, розпізнавання та аналізу об'єктів, важливим пунктом є операція правильного виділення рухомого об'єкта [18]. При роботі такої системи у реальних умовах існуючі методи пошуку та виділення об'єктів повинні враховувати наявність завад, а також враховувати зміну рівня цих завад у просторі та часі; можливе часте адаптування до зміни стану заднього плану або ж освітленості, що із великою ймовірністю впливає на кінцевий результат роботи, який дуже часто стає некоректним. Отже, необхідно також вирішувати завдання дослідження відомих методів пошуку та виявлення рухомих об'єктів для їх подальшого вдосконалення та розроблення нових підходів та методів.

Виділення рухомих об'єктів складаються із двох етапів вирішення: перший – пошуку та виділення усіх пікселів переднього плану, другий – подальше об'єднання виділених пікселів у окремі об'єкти.

Для пошуку та розпізнавання рухомих об'єктів вирішуються такі основні задачі:

- пошук та виявлення об'єкта у кадрі;
- визначення параметрів об'єкта за його характеристиками;
- визначення напрямку та швидкості руху об'єкта;
- визначення у полі обраних для ідентифікації параметрів об'єкта відстані від спостерігача до вибраного образу об'єкта;
- формування висновку щодо доцільності його подальшого супроводження наявними засобами.

При розпізнаванні образів типовими є такі задачі:

- задачі віднесення виділеного об'єкта за його характеристиками до одного із класів;
- вибір інформативних ознак об'єкта;
- виконання класифікації для множини об'єктів, явищ, ситуацій по їх характеристиках;

- приведення вихідних даних до виду, зручного для подальшого розпізнавання;

- прогнозування для прийняття рішення на майбутнє.

При виникненні необхідності відеозахоплення рухомого об'єкта необхідно забезпечити узгодження швидкості та напрямку переміщення об'єкта із динамічними параметрами наявної системи відеоспостереження.

При проектуванні систем пошуку та виділення рухомих об'єктів слід враховувати обмеженість граничних швидкостей пересування відеокамери та потенційних граничних швидкостей об'єктів. Подібні дослідження є актуальними при розробці засобів візуального спостереження як стаціонарних (контролю дорожнього руху, охоронних систем тощо), так і мобільних, наприклад, оброблення відеоданих із безпілотних повітряних апаратів.

Для визначення напрямку руху об'єкта кадр співставляється із умовною системою координат та здійснюється порівняння двох кадрів відеопотоку для виявлення областей, які відрізняються між собою. Таким підходом фіксується факт появи об'єкта у кадрі. За певними критеріями, у тому числі за формою або кольоровими ознаками визначається належність об'єкта, який підлягає відстеженню, до заданого класу. Вектор напрямку руху об'єкта визначається шляхом порівняння кадрів. Знаходження швидкості руху об'єкту здійснюється шляхом визначення часу між моментами фіксації двох кадрів та визначенням відношення між двома положеннями виділеного об'єкту за визначеним часом.

Для кодування ознак рухомих об'єктів використовується метод центральних моментів, який полягає у тому, що для кадру на площині із об'єктом визначаються координати точок на цій площині. Ці координати знаходяться відносно середніх значень всієї множини точок об'єкта, що розглядаються.

У роботі мають бути виконані такі вимоги до алгоритмів:

- робота у масштабі реального часу і невелика обчислювальна складність;

- надійне виявлення об'єкта в різний час доби;

- надійна та безвідмовна робота в будь-яку пору року.

Загальний алгоритм методу знаходження міжкадрової різниці для двох кадрів виглядає так:

- на вхід для аналізу надходять два відеокадри із відеопотоку;
- аналізуються дві послідовності байтів у форматі RGB;
- попіксельно обчислюються міжкадрові різниці;
- для кожного із пікселів обчислюється середнє значення;
- виконується порівняння отриманих значень із заданим порогом.

У результаті отримуємо двійкову маску, де одному елементу якої відповідають три складових компонентів кольору пікселя двох початкових кадрів [19]. У масці одиниці розміщуються там, де, можливо, відбувається рух, однак на цьому етапі формування маски можуть бути і невірні опрацьовування окремих елементів маски, які помилково встановлені у одиницю. У якості вхідних кадрів як правило використовують послідовні кадри із відеопотоку, а також можливе використання кадрів із значним інтервалом, як приклад, використання від одного до трьох кадрів. Цей підхід доволі простий для його реалізації. Суттєвим недоліком такого підходу є фіксація завад апаратури під час реєстрації даних. Для точної локалізації і виявлення рухомих об'єктів розглядається метод різниці поточного та базового кадрів, який є кадром фону. На якість виявлення руху впливає спосіб формування базового кадру, так як він повинен мати певні властивості:

- мінімальний рівень завад;
- якщо аналізований кадр являє собою кадр реального зображення із відеопотоку, то він має за часом мінімально відставати від поточного кадру;
- якщо базовий кадр формується штучно, то він повинен мати мінімальну кількість рухомих елементів, в іншому випадку неминучі помилкові спрацьовування на такі об'єкти, які на поточному кадрі вже відсутні, проте у базовому кадрі містяться деякі їх елементи.

Перед процесом оновленням кадру необхідно здійснити фільтрацію. У процесі формування фонового кадру основною задачею є відрізання ділянок, на

яких є рух. Перевагою даного методу є менша чутливість до завад. Між методами міжкадрової різниці та виділення фону існує така відмінність:

- для отримання оцінки фону аналізується декілька десятків кадрів;
- на попередньому кадрі об'єкти розміщені майже так само як і на наступному;
- проведення операції оцінювання фону вимагає обробки більшої кількості кадрів.

Процес ідентифікації об'єктів на основі застосування методу розпізнавання за допомогою шаблонів базується на розпізнанні образів та полягає у тому, що отримане зображення невідомого об'єкта послідовно порівнюється із зображеннями відомих об'єктів, що зберігаються у базі даних. Процес ідентифікації об'єктів можна розбити на три основні етапи: реєстрацію та нормалізацію отриманих зображень; розподіл та виділення їх особливостей; класифікацію об'єктів.

Відомий метод морфологічної обробки представляє собою процес аналізу зображення відносно форми об'єктів. Математичну морфологію використовують у різних типах систем, у яких виконується оброблення зображення, для досягнення різних цілей та на різних етапах роботи:

- виявлення контурів об'єктів зображення;
- покращення візуальних характеристик цифрового зображення (контрастності, яскравості тощо);
- відновлення пошкоджених зображень, наприклад, реставрація фотознімків;
- зниження рівня завад.

Метод кореляції у основному використовують для знаходження ступеня подібності виділених ділянок на різних кадрах. У результаті функція кореляції двох зображень формує максимум лише у тому випадку, коли два зображення однакові. Максимальним допустимим значенням функції кореляції буде одиниця. Основним недоліком цього методу є досить висока обчислювальна складність.

Один із алгоритмів, який відстежує переміщення деякого об'єкта у кадрі є алгоритм по відстеженню за центром мас об'єкта. Ідея цього алгоритму – знаходити центр мас об'єкта та слідкувати за його переміщенням. Суть алгоритму заключається у тому, що кольорові компоненти усіх пікселів зображення послідовно порівнюються із тим пікселем, на якому раніше був зафіксований рух.

Першим кроком у процесі ідентифікації об'єкта є виділення цього об'єкта у отриманій відеопослідовності за його деякими характеристиками, для чого використовуються математичні процедури квантування, фільтрації, нормалізації, диференціювання, фрагментизації та об'єднання ознак. Для деяких типів зображень попередня обробка заключається у соляризації або бінарізації, також можливе виділення об'єктів для систем реального часу на основі використання вертикальних зрізів.

1.5 Нейромережа глибокого навчання

Операцію виділення та розпізнавання рухомих об'єктів у відео послідовності можна реалізувати за допомогою нейромережі глибокого навчання. Операція глибокого навчання є частиною більш загальної сім'ї методів машинного навчання, яка ґрунтується на представленні даних навчання, на відміну від методів та алгоритмів конкретного завдання. Машинне навчання може бути без нагляду, під частковим наглядом або під наглядом.

Машинні методи глибокого навчання характеризуються як клас алгоритмів навчання, який:

- формує у процесі навчання шари на кількох рівнях представлення, які співвідносяться до різних рівнів абстракції; шари утворюють певну ієрархію понять;

- для вибору ознак із перетворенням використовує багатозначну систему нелінійних фільтрів. Кожен наступний шар системи отримує на вході вихідні дані попереднього шару. Система глибокого навчання може працювати із

алгоритмами навчання із учителем та без учителя, при цьому аналіз являє собою навчання без учителя, а класифікація - навчання із учителем;

- є частиною більш широкої сфери машинного навчання для вивчення та представлених даних;

- має декілька шарів, які дозволяють показувати або визначати дані (навчання без учителів). При цьому ознаки організовані ієрархічно та визначають більш високий рівень виробництва.

Алгоритми глибокого навчання відрізняються від алгоритмів неглибокого навчання по кількості параметризованих перетворень, за якими стабілізований сигнал, що поширюється від вхідного шару до вихідного шару, де параметризованими перетвореннями вважається такий блок оброблення даних, у якого присутні параметри навчання такі як ваги чи пороги.

Згортова нейронна мережа (англ. CNN, convolutionalneuralnetwork) – це спеціальна архітектура штучних нейронних мереж, що налаштована на ефективне розпізнавання зображень та відноситься до складу засобів та технологій глибокого навчання.

Основою побудови згорткових нейронних мереж є чергування згорткових шарів (convolutionlayers) і субдискретизуючих шарів (subsamplinglayers або poolinglayers, шарів підвибірки). Структура згорткої нейронної мережі - односпрямована (без зворотних зв'язків), та є принципово багат шаровою. Для навчання мережі використовуються відомі стандартні методи, найчастіше всього це метод зворотного поширення помилки.

У основі роботи згорткової нейронної мережі є інтерпретація переходу від конкретних особливостей зображення до більш загальних та абстрактних деталей, та на наступних етапах до ще більш абстрактних деталей аж до формування та виділення понять самого високого рівня. При цьому нейронна мережа самоналаштовується та формує сама необхідну ієрархію абстрактних ознак, відкидаючи незначні деталі та виділяючи істотне.

У звичайному перцептроні, який являє собою повнозв'язану нейронну мережу, кожен наступний нейрон з'єднаний із усіма нейронами попереднього

шару, причому кожний такий зв'язок має свій персональний ваговий коефіцієнт. В згортковій нейронній мережі у виконанні операції згортки використовується тільки обмежена матриця ваг незначного розміру, яку переміщують по усьому оброблюваному шару (на самому початку роботи - безпосередньо по вхідному зображенню), виробляючи після кожного зсуву відповідний сигнал активації для кожного нейрона наступного шару на аналогічній позиції. Таким чином для різних нейронів вихідного шару мережі використовуються одна і та ж сама матриця ваг, яку ще називають ядром згортки. Цю матрицю інтерпретують як графічне кодування певної ознаки, наприклад, наявності лінії під певним кутом. Тоді у наступному шарі, що вийшов в результаті виконаної операції згортки такою матрицею ваг, вказує на наявність даної ознаки у оброблюваному шарі та її координати, формуючи у підсумку так звану карту ознак (Featuremap). У згортковій нейронній мережі такий набір ваг не один, а їх ціла гама, що кодують елементи зображення (наприклад дуги та лінії під різними кутами). Такі ядра згортки не закладаються дослідником завчасно, а формуються самостійно у мережі шляхом її навчання відомим класичним методом зворотного поширення помилки. Кожний прохід таким набором ваг створює свій власний примірник карт ознак, перетворюючи нейронну мережу на багатоканальну (у одному шарі багато незалежних карт ознак). Також слід відмітити, що при обробленні шару матрицею ваг її пересувають не на повний крок (на розмір цієї матриці), а на деяку невелику відстань. Так, наприклад, при розмірі матриці ваг 5×5 її переміщують на один або два нейрони (пікселі) замість п'яти, щоб не перескочити шукані ознаки.

Наступна операція субдискретизації (Subsampling або ще Pooling, що перекладається як «операція підвибірки» або об'єднання) здійснює зменшення розмірності сформованих карт ознак. В даній архітектурі нейронної мережі вважається, що дані про факт наявності шуканої ознаки є важливішим за точне знання її координат, тому з декількох сусідніх нейронів карти ознак відбирається максимальне значення ознаки та приймається за один нейрон ущільненої карти ознак меншого розміру. За рахунок виконання цієї операції,

крім прискорення у подальших обчисленнях, нейронна мережа стає більш інваріантною до масштабу вхідного зображення.

Типова структура згорткової нейронної мережі складається з великої кількості шарів. Після надходження вхідного зображення у початковий шар сигнал проходить послідовність згорткових шарів, в яких по чергово виконується власне згортка та субдискретизація (пулінг). Чергування таких шарів дозволяє формувати «карти ознак» із карт ознак на кожному шарі, де на кожному наступному шарі така карта зменшується у розмірі, але збільшується кількість каналів для обробки. На практиці це створює можливість розпізнавання складних ієрархій ознак. Як правило після проходження декількох таких шарів карта ознак перетворюється у вектор або даже у скаляр, але таких карт ознак виникають сотні. На виході згорткових шарів нейронної мережі додатково утворюють кілька шарів повнозв'язної нейронної мережі (у вигляді перцептрона), на вхід якого передаються сформовані кінцеві карти ознак.

Особливості оброблення даних у згорткових нейронних мережах полягає у такому:

- зменшення розмірності: за рахунок шарів субдискретизації просторова розмірність зображення поступово зменшується;
- спільні ваги: існує невеликий набір ваг для великої кількості зв'язків;
- локальне сприйняття: на вхід кожного окремого нейрона подається не все зображення (виходи попереднього шару), а тільки певна його частина. Цей підхід дозволяє не змінювати та залишати топологію зображення від одного шару до наступного шару.

Розпізнавання об'єктів у такій мережі виконується незалежно від масштабу зображення. У цьому випадку сам факт знаходження ознаки є важливішим за знання місця точного його розташування на зображенні. На рисунку 1.1 приведена типова архітектура згорткової нейронної мережі.



Рисунок 1.1 – Структура згорткової нейронної мережі

В згортковому шарі мережі над зображеннями, які подаються на вхід, здійснюється операція згортки або пошук зваженої суми вибраних фрагментів зображення навколо ядра згортки:

$$(f * g)[m,n] = \sum_{k,l} (f[m - k, n - l] \times g[k,l]), \quad (1.5)$$

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.

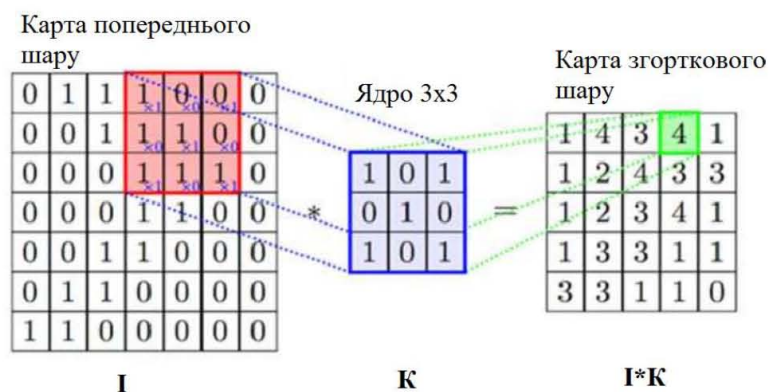


Рисунок 1.2 – Операція згортки

Ядро згортки є фактично набором заданих вагових коефіцієнтів. У результаті виконання даної операції отримуємо також деяке зображення, яке тепер буде вже називатися картою ознак. Залежно від коефіцієнтів обраного

ядра згортки карта ознак буде виділятиме ті чи інші характеристики початкового вхідного зображення.

Для найбільш повного та якісного виділення характеристик початкового вхідного зображення застосовуються кілька різних ядер згортки і тому на виході згорткового шару формуються кілька карт ознак. Ядро згортки представляє собою фільтр або ж вікно, яке переміщається по всій попередній карті та знаходить певні визначені ознаки об'єктів зображення.

Отриманий скалярний результат виконання операції згортки потрапляє на функцію активації. Для цього раніше традиційно використовувалися функції гіперболічного тангенса або ж сигмоїд. Зараз частіше застосовують функцію ReLU, яка дозволяє прискорити процес навчання мережі та спростити виконання обчислень за рахунок своєї простоти.

Наступний шар субдискретизації представляє собою нелінійну операцію ущільнення карти ознак, та при цьому група пікселів (як правило розміром 2×2) ущільнюється до одного пікселя після проходження нелінійного перетворення. Найчастіше для цього використовується функція знаходження максимуму. У перетворенні беруть участь непересічні прямокутники або ж квадрати, кожен із яких переводиться у один піксель, і при цьому вибирається той піксель, який має максимальне значення. Виконання операції субдискретизації дозволяє суттєво зменшити обсяг зображення у просторі. Пулінг трактується так: якщо на попередній операції згортки були вже виявлені деякі певні ознаки, то для подальшої обробки таке докладне зображення вже не потрібно, і тому воно ущільнюється до менш детального. Фільтрація вже непотрібних деталей зображення допомагає мережі не перенавчатися. Як правило шар субдискретизації поміщається після шару згортки перед наступним шаром для виконання згортки.

Після кількох операцій проходження згортки зображення і подальшого ущільнення за допомогою операції субдискретизації система переформатовується від конкретної сітки початкових пікселів із високою роздільною здатністю до утворення більш абстрактних карт ознак. На кожному

наступному шарі мережі збільшується число каналів та зменшується розмірність аналізованого зображення у кожному каналі. У підсумку залишається великий набір каналів, що зберігають незначну кількість даних або навіть один параметр, які трактуються як самі абстрактні поняття, що виявлені у початковому вихідному зображенні.

Потім ці дані об'єднуються та передаються на повнозв'язану нейронну мережу, яка теж може мати декілька шарів. У цьому випадку повнозв'язані шари мережі вже втрачають початкову просторову структуру пікселів та мають незначну розмірність по відношенню до кількості пікселів початкового вихідного зображення.

1.5 Висновки

У розділі був проведений аналіз сучасного стану сфери машинного зору по виділенню рухомих об'єктів. Був виконаний аналіз таких методів, як визначення переміщення областей пікселів між сусідніми кадрами, метод визначення зміщення прямокутних блоків у зображенні, метод кореляції, метод використання шаблонів, метод відстеження переміщення центру мас, розпізнавання рухомих об'єктів у відео послідовності за допомогою нейромережі глибокого навчання, а також використання детекторів та дескрипторів для виділення об'єктів руху. Розглянуто основні підходи до виділення особливих точок зображення, визначено основні їх переваги та недоліки. Обрано методи, які зможуть більш вдало реалізувати завдання виділення рухомих об'єктів.

2 РОЗРОБКА МЕТОДУ І АЛГОРИТМУ ВИДІЛЕННЯ РУХОМИХ ОБ'ЄКТІВ

Розробці методу та алгоритму пошуку рухомих об'єктів у отриманій відео послідовності із використанням підходу по знаходженню точок інтересу у зображенні та використанням нейронної мережі для виділення рухомих об'єктів присвячений даний розділ.

2.1 Відстеження об'єктів

Пошук та відстеження переміщення об'єктів є двома різними процесами. При пошуку та виявленню об'єкта виконується операція визначення місця знаходження об'єкта у кадрі. Детектор для пошуку об'єкта, як правило, є більш обчислювально складним і, отже, працює більш повільно, ніж алгоритм для подальшого відстеження переміщення об'єктів. Приклади алгоритмів для пошуку та виявлення об'єктів включають до своєї роботи каскади Хаара, HOG + Linear SVM та детектори на базі глибокого навчання, такі як YOLO, Faster R-CNNs і SingleShotDetectors (SSD) або інші. Приклади алгоритмів відстеження переміщення об'єктів можуть включати для своєї роботи такі підходи як: MOSSE, MedianFlow, GOTURN, дискримінаційні кореляційні фільтри, ядерні кореляційні фільтри.[21-25]

Виявлення та відстеження переміщення об'єктів включає в себе такі етапи:

- початковий пошук та виявлення об'єктів та формування набору вхідних даних із координатами, що обмежують розміри блоків у вигляді об'єктів;
- присвоєння унікального ідентифікатора для кожного виявленого об'єкту;
- подальше відстеження кожного виявленого об'єкта при переміщенні у кадрі з обов'язковим збереженням присвоєних їм унікальних ідентифікаторів.

Для визначення траєкторії переміщення об'єкта відстеження об'єктів має першочергове значення.

До алгоритму відстеження об'єктів у рамках даної роботи були виявлені та повинні бути виконані наступні вимоги:

- етап виявлення об'єкта потрібно виконувати один раз на початку роботи (тобто об'єкт спочатку повинен бути виявлений);
- швидкість роботи алгоритму по відслідковуванню переміщення об'єкта повинна бути вище, ніж у детектора по пошуку об'єкта;
- алгоритм має бути стійким до оклозії (приховування, перекриття);
- можливість повторної обробки, коли відстежуваний об'єкт «зникає» або переміщується за існуючі межі рамки відео;
- можливість відновлення стеження за об'єктами, які були із тих чи інших причин «втрачені» між кадрами.

2.2 Механізм створення траєкторії

Розроблений алгоритми створення траєкторії руху об'єкта повинен об'єднувати в собі єдиний механізм концепції пошуку, виявлення та відстеження переміщення об'єктів, який як правило розділений на дві окремі фази: виявлення та відстеження.

Перша фаза роботи алгоритму – процес виявлення. В ході виконання фази виявлення використовується більш складний у обчислювальному плані алгоритм виявлення об'єктів у кадрі, щоб:

- визначити, чи є нові об'єкти у отриманому кадрі;
- визначити, чи виявлені об'єкти, які були із різних причин «втрачені» під час виконання фази відстеження.

Для кожного виявленого у кадрі об'єкту створюється або оновлюється відповідна мітка відстеження із новими координатами. Так як детектор пошуку об'єктів як правило має більш високу обчислювальну складність, його дія виконується один раз при надходженні N кадрів.

Друга фаза роботи алгоритму – процес відстеження. Вона виконується без схильності до частого повернення до виконання процесу виявлення. Для кожного із знайдених у ході виконання фази виявлення об'єктів створена мітка

для подальшого відстеження об'єкта при його переміщенні по послідовності отриманих кадрів. Механізм відстеження об'єкта повинен діяти швидше та ефективніше, ніж попередній етап роботи використанням детектора пошуку об'єкта. Процес відстеження триває аж до досягнення заданого N-го кадру, потім знову вступає у дію детектор пошуку об'єктів. Надалі весь процес роботи алгоритму повторюється.

Перевагою даного гібридного підходу є те, що можна використовувати високоточні методи пошуку та виявлення об'єктів без значного обчислювального навантаження на весь процес відслідковування рухомих об'єктів.

Відомі способи автоматичного аналізу цифрових зображень [20], які дозволяють знаходити та виділяти рухомі об'єкти у полі зору відеокамери та використовувати для цього процесу порівняно прості обчислювальні пристрої. У даних способах, як правило, формується модель нерухокої сцени, потім порівнюють кожний новий кадр із відеопотоку з побудованою моделлю сцени і шукають відмінності нового кадру та нерухокої сцени. При цьому, як правило, знаходять та виділяють не тільки рухомі об'єкти, але й завади, обумовлені впливом різного типу перешкод: зміщенням деяких об'єктів сцени під дією вітру, зміною у освітленості сцени внаслідок дії зовнішніх рухомих освітлювачів або ж мінливій хмарності. У багатьох випадків до перешкод відносять також рух птахів, тварин, комах у безпосередній близькості від установлених відеокамер. Тому швидкодіючі різницеві методи пошуку та визначення рухомих об'єктів у послідовності відео кадрів мають певні проблеми при розрізненні перешкод у зображенні.

Відомий алгоритм OnlineBoostingTracker [26] – один із перших алгоритмів відстеження об'єктів, який має досить високу точність. У цьому алгоритмі використовується версія алгоритму AdaBoost, та каскади Хаара. Подальшим розвитком є алгоритм MILTracker (MultipleInstanceLearningTracker) [27] де використовується навчання із учителем. У цьому алгоритмі шукається місце знаходження об'єкта та інші області, де ще може бути такий об'єкт.

Алгоритм KCFTracker (KernelizedCorrelationFiltersTracker) [27] використовує ідеї двох попередніх алгоритмів та має більшу швидкість.

Останнім часом для аналізу цифрових зображень використовують апарат загорткових нейронних мереж [22, 23], за допомогою яких можна відшукати рухомі об'єкти при дії перешкод. Достовірні результати аналізу цифрових зображень на основі таких нейронних мереж досягаються за допомогою обчислювальних пристроїв з досить високою обчислювальною потужністю, а для адаптації загорткових нейронних мереж до виконання операції фільтрації перешкод потрібен значний набір зображень, на яких така нейромережа буде навчатися. Це створює певні додаткові умови для широкого використання нейронних мереж.

2.3 Використання нейронної мережі для виділення об'єктів

Повнозгорткові нейронні мережі застосовуються для задач, в яких необхідно визначити форму і місце знаходження шуканого об'єкта або декількох об'єктів. Подібні типу завдання досить проблематично вирішувати із використанням простих загорткових нейронних мереж. Для визначення, у яких випадках та чому краще використовувати повно згорткові мережі замість звичайних згорткових мереж, необхідно виконати порівняння зазначених типів нейронних мереж.

Найбільш очевидною відмінністю повнозгорткових мереж від інших типів нейронних мереж є кінцевий результат роботи цих мереж. Прості згорткові мережі використовуються для класифікації, визначаючи до якого із класів належить аналізоване зображення, та для локалізації об'єкта у зображенні. У результаті використання будь-якого із цих методів кінцевими даними є числа або масиви чисел. Таким чином можна отримати досить обмежену інформацію про зображення, але не можемо перетворити її у необхідний нам вигляд.

Повнозгорткові нейронні мережі на виході видають сегментоване зображення, що відповідає по розмірності вхідному зображенню. Тому інша

назва таких мереж - сегментаційні нейронні мережі. Сегментація - це процес об'єднання об'єктів у групи за певними загальними ознаками. У підсумку отримуємо від такої мережі набагато більше корисної інформації, а досягнутий кінцевий результат можемо обробити досить простими евристичними методами.

Для кращого розуміння принципу роботи повнозгорткових мереж та з'ясування, для вирішення яких задач їх можна використовувати, важливо ознайомитися із їх загальною архітектурою. При проектуванні загорткових мереж у архітектуру можуть вноситися різноманітні засоби для підвищення точності розпізнавання (*localresponse normalization layer*, *dropout layer* і т.п.). Розглянемо базову архітектуру, яка практично є незмінною та визначає роботу повнозгорткової мережі.

Основа роботи повнозгорткових мереж – це згортка цифрового зображення. Ключовими шарами є згорткові шари (*convolution layer*). В загортковому шарі фіксується кількість виходів із шару, ядро згортки та його крок, розміри ядра, відступ.

При виконанні операції згортки ядро проходить по усьому зображенню, в результаті чого отримуємо величину відгуку на дію ядра згортки у кожній точці зображення. Кількість ядер кожного шару згортки дорівнює добутку кількості вхідних картинок на кількість виходів із шару. Отримані результати проходять обробку через наступний шар згортки, формуючи значення вже для інших ядер. До кожного загорткового шару можна додавати шари нормалізації або регуляризації (у залежності від бажання та вибору розробника). Після проходження зображень через багато загорткових шарів отримуємо значну кількість різноманітних можливих інтерпретацій початкового зображення (рис 2.1).

Після проходження через необхідну кількість шарів згортки оброблене зображення потрапляє до шару пулінг (*pooling layer*). У цьому шарі відбувається зменшення розміру вхідних зображень без зміни їх кількості. Цей шар має ядро, яке рухається аналогічно ядру згортки, та обчислює єдине значення для кожної

такої області зображення. Зменшення початкового зображення сприяє більш швидкій обробці загортковою мережею більшої кількості отриманих даних. Це дозволяє додавати у наступних загорткових шарах більшу кількість виходів та підвищує також точність отриманих результатів. Річ у тому, що на зменшеному таким чином зображенні ядра згортки того ж самого розміру тепер здатні захоплювати значну більшу область шуканого об'єкта.

Послідовність дій у вигляді роботи шарів: convolution /.../ convolution / pooling (де розробником визначається кількість шарів convolutionlayers) може здійснюватися декілька разів, аж до тих пір, поки не буде досягнутий бажаний мінімальний розмір зображення. Цей розмір зображення визначається експериментальним шляхом.

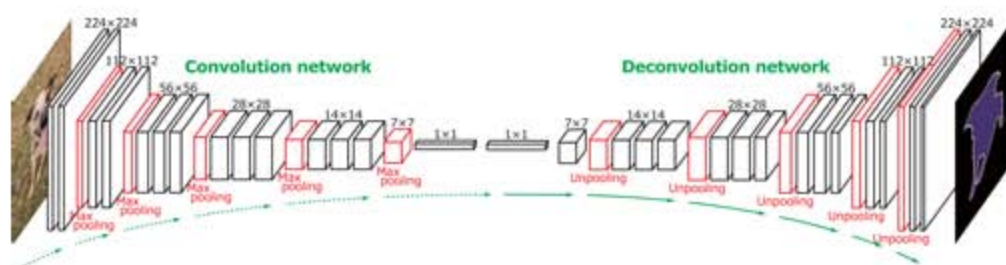


Рисунок 2.1 – Повнозгорткова нейронна мережа

Для того щоб знайдені об'єкти у кінцевому підсумку відповідали початковому оригінальному масштабу, отримане зменшене зображення необхідно повернути до значення початкового розміру. Цю задачу виконує шар `upsample` (`upsamplelayer`) та збільшує оброблене зображення. На кожний вихід для поновлення розмірів є два вхідних зображення: перше - це оброблене зображення у вигляді певної картинки із попереднього шару (це можуть бути шари `convolution` або `pooling`), та друге - це відповідна картинка із `poolinglayer`, кількість виходів якої відповідають кількості входів відповідного `upsample`, а також розміри вихідної картинки `pooling` дорівнюють розміру вхідної картинки `upsample`.

Так чином отримуємо симетричну архітектуру мережі щодо останнього шару `pooling` та першого шару `upsample`. Між шарами для збільшення розмірів

зображення також поміщаються згорткові пари, але кількість виходів із них поступово зменшується.

Послідовність шарів *upsample / convolution / ... / convolution* необхідна для того, щоб привести оброблене зображення до початкових розмірів, при цьому відбувається скорочення кількості можливих інтерпретацій цифрового зображення до кількості шуканих груп об'єктів.

2.4 Алгоритм виявлення об'єктів

Відеосигнал, який формується цифровою відеокамерою, представляє собою послідовність відеокадрів, кожен із яких описується прямокутною матрицею у вигляді растра, яка складається із N стовпців та M рядків. Елементи матриці $I_t[m, n]$ відповідного отриманого відеокадру називаються пікселями зображення та є скалярними величини у випадку монохромного зображення.

За відсутності руху у сцені спостереження відеокамера системи перетворює зображення нерухомої сцени, яку називають фоном $BG[m, n]$. Якщо відкинути вплив на отримане цифрове зображення відеокадру завад активних елементів системи формування цифрового зображення (завади матриці фотоелектронного перетворювача, аналого-цифрового перетворювача, підсилювачів та тощо), а також припустити, що за час отримання певної послідовності відеокадрів умови знімання та освітлення сцени є незмінними, то при відсутності руху у зоні спостереження, яка відповідає місцезнаходженню пікселю із координатами $[m, n]$, можна вважати його значення його параметрів постійними. Таким чином постійне значення яскравості деякого пікселя $I[m, n]$ у межах певної послідовності отриманих відеокадрів можна розглядати як критерій відсутності руху у відповідній області спостережуваної сцени.

Але на практиці під час створення зображень відеокадрів наявність випадкових завад неминуча. Головними джерелами таких завад є власні завади апаратури формування зображення, похибка квантування сигналу у разі перетворення аналогового сигналу у цифрову форму, а також похибки корекції отриманого цифрового значення кожного пікселя зображення (як приклад, це

може бути похибка заокруглення даних під час виконання різних математичних операцій).

Використовуючи адитивну модель сигналу у загальному випадку відеосигнал $I_t[m,n]$ можна представити як суміш істинного значення сигналу $I_t^*[m,n]$ та значення деякої випадкової завади $\varepsilon_t[m,n]$:

$$I_t [m,n] = I_t^* [m,n] + \varepsilon_t [m,n], \quad (2.1)$$

де $\varepsilon_t [m,n]$ – значення реалізації випадкового процесу дії завади у пікселі t-кадру із координатами $[m,n]$.

У тому випадку, коли відеокамера фіксує нерухому сцену, істинне значення отриманого відеосигналу відповідає фону $BG[m,n]$ і тоді:

$$I_t[m,n] = BG[m,n] + \varepsilon_t [m,n]. \quad (2.2)$$

Беручи до уваги наявність дії незалежних випадкових факторів, які позначаються на флуктуаціях вихідного сигналу із відеокамери, можна із достатньою точністю розглядати значення завади ε як цілком випадкової величини, яка описується нормальним законом розподілу та характеризується функцією густини розподілу ймовірностей із математичним очікуванням μ та із середнім квадратичним відхиленням σ .

Рух у отриманій послідовності відеосигналів проявляється як різка зміна яскравості деяких окремих пікселів, за координатами яких фіксується переміщення спостережуваного об'єкта у просторі. Інші зміни яскравості здійснюються досить плавно, наприклад, зміна природного освітлення протягом дня.

Допустимо, що під час настроювання системи моніторингу руху об'єктів, який відбувається протягом перших K кадрів, відеокамера фіксує нерухому сцену, у якій містяться лише статичні об'єкти фону. Ця умова дає змогу

достатньо точно визначити оцінку істинного значення яскравості фону $\mu[m,n]$ як середнього значення отриманих зображень та вирахувати величину дисперсії завади $\sigma^2 [m,n]$ відеокамери.

Після визначення параметрів сцени система переходить в режим виділення рухомих об'єктів. Критерієм знаходження рухомих об'єктів у межах пікселя із координатами $[m,n]$ можна скористатися такою умовою:

$$|\mu[m,n] - I_t[m,n]| \geq T, \quad (2.3)$$

де T – деяке задане порогове значення.

На вибір значення порогу T впливають різні фактори (значення завад, контраст рухомого об'єкта по відношенню до фону та інше). Відповідно до однієї із властивості нормального закону розподілу, можна допустити, що всі значення яскравості пікселів фону із урахуванням завад повинні потрапляти у інтервал від $T_1 = I_t [m,n] + 3\sigma [m,n]$ до $T_2 = I_t [m,n] - 3\sigma [m,n]$.

Із урахуванням цієї властивості нормального закону розподілу критерій виявлення руху об'єктів можна записати таким чином:

$$|\mu[m,n] - I_t[m,n]| > 3\sigma[m,n] \quad (2.4)$$

де $\sigma [m,n]$ – значення середнього квадратичного відхилення, що було попередньо визначено у процесі настроювання системи моніторингу руху об'єктів для кожного пікселя із координатами $[m,n]$. Для того, щоб алгоритм виділення руху був адаптивний до зміни фону під час спостереження для кожного наступного кадру для кожного пікселя цього кадру знаходиться поточне значення середньої яскравості як істинне значення яскравості пікселя:

$$\mu_{t+1}[m,n] = \alpha\mu_t[m,n] + (1 - \alpha)I_t[m,n], \quad (2.5)$$

де α – коефіцієнт швидкості навчання для оновлення фону.

Слід відзначити, що оновлення параметрів фону зображення необхідно здійснювати лише в тих областях кадру, де на даний момент часу рух відсутній. Відмічена особливість цього алгоритму зумовлена тим, що за значної частоти утворення кадрів відеокамери та досить повільному переміщенню об'єкта по сцені цей об'єкт поступово може бути за допомогою операції оновлення буде перенесений у фон зображення.

Для адаптації описаного алгоритму виявлення рухомих об'єктів до динамічного фону сцени паралельно із описаним вище основним підходом використовується також метод міжкадрової різниці.

2.5 Виділення об'єктів нейромережею MobileNet

MobileNet – сучасна архітектура згорткової нейронної мережі, що дозволяє знаходити та розпізнавати об'єкти у зображенні із досить високою точністю та низьким рівнем використання програмних ресурсів. Мережу розроблено компанією Google. Архітектура MobileNet знаходиться у відкритому доступі, що робить можливим її використання у даній роботі. Проект цієї мережі підтримується та активно розвивається. На даний час випущена версія MobileNetV2 [22].

До програмного забезпечення, де може використовуватися дана архітектура, ставляться низькі системні вимоги і тому вона може виконуватися на пристроях з обмеженими ресурсами, такими як одноплатні комп'ютери або смартфони. Архітектура мережі MobileNets відрізняється від традиційних згорткових нейронних мереж підходом до використання глибинної віддільної згортки.

Загальна ідея глибинної сепарабельної згортки полягає у тому, щоб розбити процес виконання згортки на дві стадії: на першому етапі виконується глибинна згортка 3×3 ; потім слід за нею виконується потокова згортка 1×1 . Таке рішення дозволяє зменшити фактичну кількість параметрів у нейронній мережі. На малюнку 1 зліва представлений блок звичайної загорткової мережі, а праворуч приведено базовий блок MobileNet.

Згорткова частина мережі MobileNet складається із одного звичайного згорткового шару із згорткою 3×3 на початку та тринадцяти блоків, що зображені справа на малюнку, із поступово наростаючою кількістю фільтрів та зменшенням просторової розмірності тензора. Характерною особливістю цієї архітектури є відсутність maxpooling-шарів. Для зниження просторової розмірності замість них використовується згортка із параметром stride, що рівний 2. Двома гіпер параметрами архітектури мережі MobileNet є α (множник ширини) та ρ (множник глибини або множник дозволу). Множник ширини α відповідає за кількість каналів у кожному шарі. Наприклад, $\alpha = 1$ дає надає нам архітектуру, що описана вище, а значення $\alpha = 0.25$ формує архітектуру із зменшенням у чотири рази числом каналів на виході кожного блоку мережі. Множник дозволу формує просторові розміри вхідних тензорів. Наприклад, коли $\rho = 0.5$, то це означає, що ширина та висота featuremap, яка поступає на вхід кожного шару, буде зменшена у два рази. Приведених два параметри дозволяють змінювати розміри нейронної мережі. При зменшенні параметрів α та ρ знижується точність розпізнавання зображення, але у той же час збільшується швидкість роботи мережі та зменшується обсяг використовуваної пам'яті.

Мережа MobileNet — це клас нейромережевих моделей, що орієнтовані на використання у області мобільних та вбудованих додатків у галузі комп'ютерного зору. Така архітектура ґрунтується на використанні роздільної згортки у глибину (Depthwise separable convolution). У відмінності від багатьох інших відомих моделей, що орієнтуються на ті ж класи пристроїв, для яких первинною метою при розробці є зменшення розміру нейронної мережі, головною метою при розробці мережі MobileNet є зменшення тривалості затримки при роботі мережі, а додатковим ефектом є зменшення розміру.

Спільна колективна згортка у глибину дозволяє суттєво зменшити обсяг обчислень, які виконуються у перших декількох шарах, і тим самим вирішує головну мету даної моделі.

Перший шар мережі MobileNet виконує стандартну операцію згортки над усіма каналами вхідного зображення. Ядро згортки проходить по зображенню, виконуючи обчислення на кожному кроці зваженої суми вхідних пікселів, які охоплені ядром, по усіх вхідних каналах. Особливо важливим є те, що операція згортки об'єднує значення усіх вхідних каналів. Якщо зображення має 3 початкових вхідних канали, то запуск одного ядра згортки із цього зображення формує вихідне зображення із одним каналом на піксель.

Тому для кожного вхідного пікселя зображення, незалежно від того, скільки попередніх каналів він має, згортка формує новий вихідний піксель тільки із одним каналом (на практиці використовують набір ядер згортки на початковому вхідному зображенні, і потім кожне із них отримує свій окремий власний канал на виході зображення).

У моделі MobileNet використовується стандартна згортка, але всього тільки один раз на першому початковому рівні. Інші шари мережі виконують так звану роздільну згортку у глибину. Ця згортка представляє із себе комбінацію двох різних варіантів виконання згортки: згортки у глибину та поточечну згортку.

На відміну від звичайної згортки дана згортка не об'єднує вхідні канали, а здійснює згортку по кожному каналу окремо. Для зображення із трьома каналами згортка у глибину утворює вихідне зображення, яке також має три канали. Кожен із каналів отримує свій власний набір ваг. Завданням згортки у глибину є фільтрація вхідних каналів. Цей процес використовується у завданнях по виявленню меж, колірній фільтрації і т.д.

Згортка у глибину також може мати певний множник каналу. Якщо цей множник дорівнює 2, то для кожного вхідного каналу мережі згортка утворює 2 вихідних канали та формує 2 різних наборів ваг. Але у мережі MobileNet цей параметр не використовується.

Після виконання операції згортки у глибину здійснюється операція поточної згортки, що являє собою операцію згортки із ядром розміром 1×1 .

Описана операція просто згортає усі канали, обраховуючи їх зважену суму. Як і для стандартної згортки, як правило використовується набір таких поточкових ядер, який утворює вихідне зображення із багатьох каналів. Метою поточної згортки є завдання об'єднання вихідних каналів згортки у глибину для формування нових ознак.

У підсумку, коли ці два підходи об'єднуються - згортка у глибину та послідовна поточкова згортка, отримуємо колективну згортку у глибину. На відміну від стандартної операції згортки, яка виконує як операцію фільтрації, так і операцію об'єднання за один прохід, у роздільній згортці у глибину ці дві операції здійснюються як окремі кроки.

У кінцевому підсумку результати обох цих підходів досить подібні - вони і фільтрують дані і формують нові ознаки, але виконання звичайної згортки вимагає значно більшого обсягу обчислень, щоб у підсумку отримати такий же результат та також потрібно більше ваг. Внаслідок цього, хоча ці підходи виконують приблизно одне і теж саме, роздільна згортка у глибину працює значно швидше.

Мережа MobileNet використовує до 13 таких подільних згорток у глибину підряд. Повна мережа MobileNet має 30 шарів. Топологія побудови мережі при цьому досить проста: 1) згортковий шар із кроком два; 2) шар згортки у глибину; 3) поточковий згортковий шар, який подвоює кількість каналів; 4) тепер шар згортки у глибину із кроком два; 5) знову поточковий згортковий шар, який подвоює кількість каналів; 6) тепер шар згортки у глибину; 7) тепер поточковий згортковий шар; 8) наступний шар згортки у глибину із кроком два; 9) поточковий згортковий шар, який подвоює кількість каналів.

Після першого шару як стандартної операції згортки шари у глибину та поточковий шари чергуються. На певних позиціях шар згортки у глибину приймає розмір кроку, рівний двом, щоб зменшити розмірність оброблених даних у міру проходження їх по мережі. Час від часу поточковий шар подвоює кількість каналів даних. За усіма загортковими шарами виконується функція активації ReLU.

Цей процес продовжується до тих пір, поки початкове вихідне зображення розміром 224×224 не буде зменшене до розміру 7×7 пікселів, але із 1024 каналами. Після цього оброблене зображення надходить у шар із агрегацією на основі визначення середнього, який працює по усьому зображенні, та на виході із якого отримуємо зображення розміром $1 \times 1 \times 1024$, яке фактично є вектором із 1024 елементів. Коли мережа MobileNet використовується у якості основи для побудови класифікатора, як, наприклад, у ImageNet, у якого 1000 можливих класів, то останній шар мережі є повнозв'язним шаром із softmax активацією та із 1000 виходами. Якщо ж мережа MobileNet використовується для іншого набору даних або ж як мережу, яка формує ознаки, а не як класифікатор, то для розглянутого прикладу можна використовувати інший шар у якості кінцевого.

У мережі MobileNet також можливо використовувати зменшену версію її архітектури. У мережі є три гіперпараметра, які можна налаштувати, щоб визначити розмір мережі:

- множник ширини (α), який визначає кількість каналів. Якщо множник ширини дорівнює 1, то мережа починається із 32 каналів та закінчується 1024;
- множник дозволу (ρ), що відповідає за розміри вхідного зображення для обробки (визначає просторові розміри вступних тензорів). Розмір введення зображення по замовчуванню становить 224×224 пікселів;
- зменшення глибини: у повній мережі є група із п'яти шарів у середині, яку можна не використовувати без істотної втрати точності для роботи мережі.

Ці зміни параметрів можуть використовуватися, щоб утворити мережу меншого розміру, що збільшує швидкість обробки. Однак такі зміни параметрів впливають на точність роботи мережі.

2.6 Висновки

В даному розділі виконано опис послідовності роботи по пошуку та виділенню рухомих об'єктів із використанням нейромережових технологій. Основним підходом до пошуку та виділення рухомих об'єктів у відеопотоці

використано пошук точок інтересу у зображенні для знаходження рухомих об'єктів. Розроблено алгоритм роботи програми для виділення точок інтересу.

Для виділення ознак рухомих об'єктів використано детектори для знаходження цих об'єктів у аналізованій сцені та наступного застосування нейронної мережі глибокого навчання для виділення рухомих об'єктів. У розділі розроблено послідовність обробки цифрового зображення для пошуку рухомих об'єктів у вибраній сцені. Запропонований у роботі підхід по використанню детекторів для виділення точок інтересу та застосування нейронної мережі глибокого навчання дозволяє виконати процес обробки комп'ютерних зображень із метою знаходження рухомих об'єктів.

3 РОЗРОБКА ПРОГРАМНИХ ЗАСОБІВ ВИДІЛЕННЯ РУХОМИХ ОБ'ЄКТІВ

Розробці програмних засобів пошуку рухомих об'єктів у отриманій відео послідовності із використанням нейронної мережі для виділення рухомих об'єктів та перевірки їх працездатності присвячений даний розділ.

3.1 Вибір інструментальних засобів для розробки програмного продукту

В теперішній час існує велика кількість різновидів програмування, а також мов програмування, тому для того, щоб вибрати мову та середовище необхідно висунути деякі критерії, наприклад: кросплатформенність, а також підтримка бібліотеки OpenCV.

Значна частина мов програмування підтримує бібліотеку OpenCV, тому і вибір буде іти між основними мовами програмування такими як: Java, C# та Python.

Хоча мова Python спочатку була в тіні своїх конкурентів, але при цьому коли були проаналізовані деякі переваги та недоліки, можна зараз зробити висновок, що на нинішньому етапі розвитку вона є досить універсальною та гнучкою, отже було прийнято рішення, що за основу буде взято мову Python.

Python — це мова програмування високого рівня, інтерпретована та об'єктно-орієнтована зі строгою динамічною типізацією. Вона є привабливою для стрімкої розробки програм, а також як засіб поєднання наявних компонентів. Python підтримує пакети модулів та модулі, і саме це сприяє повторному використанню модульності та коду. Стандартні бібліотеки та інтерпретатор Python доступні як у вихідній, так і у скомпільованій формі на всіх основних платформах. В мові програмування Python підтримується кілька парадигм програмування, зокрема: процедурна, об'єктно-орієнтована, аспектно-орієнтована та функціональна.

Мова Python підтримує динамічну типізацію, а саме тип змінної визначається лише під час виконання. Через це замість «присвоювання значення змінної» потрібно говорити про «зв'язуванні значення з деякими іменами». В мові Python є вбудовані типи: Unicode-рядок, булевий рядок, число з плаваючою комою, ціле число довільної точності, комплексне число і деякі інші. З колекцій в Python вбудовані: кортеж (незмінний список), список, словник, множина і інші. Усі значення, такі як функції, методи, модулі, класи є об'єктами.

Новий тип можливо додати написавши клас (class), або визначивши тип в модулі розширення (наприклад, написаному на мові C). Система класів підтримує метапрограмування і спадкування (одиначне і множинне). Можливе успадкування від більшості вбудованих типів і типів розширень.

Усі об'єкти поділяються на атомарні і посилальні. До атомарних відносяться int, long, complex і деякі інші. При присвоєнні атомарних об'єктів копіюється їх значення, в той час як для посилальних копіюється тільки покажчик на об'єкт, таким чином, обидві змінні після присвоювання використовують одне і те ж значення. Довідкові об'єкти бувають змінювані і незмінні. Наприклад, рядки і кортежі є незмінними, а списки, словники і багато інших об'єктів - змінними. Кортеж в Python, по суті, є незмінним списком. У багатьох випадках кортежі працюють швидше списків.

Переваги Python:

- досить не складний у вивченні, особливо на початковому етапі;
- відкрита розробка;
- надає ресурси динамічної семантики і швидкого прототипування;
- велика кількість корисних бібліотек і розширень мови можна використовувати у своїх проектах завдяки програмним інтерфейсами та гранично уніфікованого механізму імпорту;
- абсолютно усе у Python є об'єктом у сенсі ООП, хоча при цьому об'єктний підхід не нав'язується розробнику;

– механізми модульності дуже добре продумані і також можуть легко використовуватись.

Недоліки Python:

– на Python створено не дуже велика кількість якісних програмних проектів в порівнянні з деякими іншими універсальними мовами програмування, такими як Java;

– не дуже вдала підтримка багатопоточності;

– відсутність комерційної підтримки засобів розробки (однак ця ситуація з часом змінюється);

– бенчмарки показують на багато меншу продуктивність Python у порівнянні із основними Java VM, що створює цій мові репутацію повільного середовища.

– початкова обмеженість ресурсів для роботи із базами даних;

Як було стверджено раніше, у програмі використовується бібліотека OpenCV, для виконання цілого ряду задач які поставленні програмному продукту.

OpenCV (у перекладі з англ. OpenSourceComputerVisionLibrary, бібліотека комп'ютерного зору із відкритим вихідним кодом) - бібліотека алгоритмів комп'ютерного зору, чисельних алгоритмів загального призначення із відкритим кодом та обробки зображень[27].

3.2 Вибір середовища розробки

Інтегроване середовище розробки — це комплекс програмних цілей для розробки програмного забезпечення. Інтегроване середовище розробки складається із інструментів для автоматизації складання та відлагоджування програм, а також редактора початкового коду. Велика кількість сучасних середовищ розробки, зазвичай, мають можливість автодоповнення коду.

Деякі середовища розробки містять інтерпретатор, компілятор або ж обидва (наприклад Eclipse та NetBeans), інші не містять жодного з них (Lazarus та SharpDevelop). Деякі інтегровані середовища розробки містять інструменти

для полегшення розробки інтерфейсу користувача (XCode, EmbarcaderoDelphi) (GUI) або систему керування версіями. Багато сучасних IDE містять інспектор об'єктів, інспектор класів, схему ієрархії класів для легкої розробки об'єктно-орієнтованої програмного забезпечення.

Інтегровані середовища розробки створювалися з метою максимізувати продуктивність розробника, при цьому йому надаються пов'язані інструменти розробки із схожими інтерфейсами як одну програму, у якій відбуватиметься весь процес розробки і яка надає необхідні функції для компілювання, модифікації, розгортання та налагодження програмного забезпечення.

Одним з завдань IDE є зменшення кількості часу, необхідного на конфігурацію різних інструментів розробки, при цьому пропонуючи цей самий набір, як єдине ціле. Це може збільшити продуктивність розробника, тільки у випадку, коли навчання працює як інтегроване середовище розробки та є швидшим, ніж освоєння усіх інструментів. Крім цього, більша інтеграція між вбудованими інструментами може сприяти додатковому збільшенню продуктивності. Наприклад, синтаксичний аналіз коду може проходити під час його редагування, та при цьому будуть виявлятися помилки ще до трансляції коду.

Деякі інтегровані середовища розробки використовують для певної мови програмування (або двох та більше споріднених мов), надають набір можливостей, що більше підходять до парадигми програмування деякої відповідної мови. Такими IDE є, наприклад Xcode, PhpStorm, Xojo та Delphi.

З другої сторони, існує багато більш універсальних IDE, що є багатомовними, наприклад ActiveStateKomodo, Eclipse, IntelliJ IDEA, MyEclipse, NetBeans, OracleJDeveloper, Codenvy та Microsoft VisualStudio.

Тому проаналізувавши усі вищезгадані середовища розробки, було вирішено використовувати PyCharm, середовище розробки, що було розроблено для мови програмування Python.

3.3 Розробка програмного продукту

Задачею магістерської кваліфікаційної роботи є розробка програмного продукту виділення рухомих об'єктів у відеопотоці. Для цього знадобиться встановити мову програмування Python на персональному комп'ютері, а також скачати та встановити бібліотеку Opencv. Після того як це буде зроблено потрібно встановити саме середовище програмування (PyCharm). Після цього потрібно впевнитись у тому, що всі попередні налаштування правильні і тільки після цього можна починати писати програмний код.

Першим кроком є підключення потрібних бібліотек :

```
fromimutils.videoimportVideoStream
fromimutils.videoimportFPS
importnumpyasnp
importargparse
importimutils
importtime
importcv2
```

Далі аналізується аргументи командної стрічки:

```
ap = argparse.ArgumentParser()
ap.add_argument("-p", "--prototxt", required=True,
                help="pathtoCaffe 'deploy' prototxtfile")
ap.add_argument("-m", "--model", required=True,
                help="pathtoCaffepre-trainedmodel")
ap.add_argument("-c", "--confidence", type=float, default=0.2,
                help="minimumprobabilitytofilterweakdetections")
args = vars(ap.parse_args())
```

У цьому коді атрибути використані в наступному контексті:

- prototxt: шлях до prototxtCaffe файлу;

- Model: шлях до передчасно заготовленої моделі;
- Confidence: мінімальний поріг валідності для розпізнавання об'єкта (значення за замовчуванням приймається рівним 20%)

Після цього додаються основні об'єкти. Ініціюється список класів та набір кольорів:

```
CLASSES = ["background", "aeroplane", "bird", "bicycle", "boat",
           "bottle", "bus", "cat", "car", "chair", "cow", "diningtable",
           "horse", "dog", "motorbike", "person", "sheep", "pottedplant",
           "train", "sofa", "tvmonitor"]
COLORS = np.random.uniform(0, 255, size=(len(CLASSES), 3))
```

Далі ініціюється мітки CLASS та інші випадкові кольори. Далі завантажується модель та налаштовується відео потік.

```
print("[INFO] loadingmodel...")
net = cv2.dnn.readNetFromCaffe(args["prototxt"], args["model"])

print("[INFO] startingvideostream...")
vs = VideoStream(src=0).start()
time.sleep(2.0)
fps = FPS().start()
```

Завантажується серіалізована модель, що надає посилання на prototxt і моделі, що найбільш зручніше та простіше робиться у OpenCV.

Потім ініціалізується відео потік (це може бути відеофайл або веб-камера). Спочатку запускається VideoStream, потім завантажується камера, і, нарешті, починається відлік кадрів в секунду. Класи VideoStream і FPS є частиною пакета imutils.

Далі розроблюється код для роботи з кадрами. Проходимо по кожному кадру (для того, щоб збільшити швидкість можна пропускати кадри).

```
while True:
```

```
    frame = vs.read()
    frame = imutils.resize(frame, width=400)
    (h, w) = frame.shape[:2]
    blob = cv2.dnn.blobFromImage(cv2.resize(frame, (300, 300)),
                                0.07843, (300, 300), 127.5)
    net.setInput(blob)
    detections = net.forward()
```

Спершу зчитується кадр з потоку, а потім замінюється його розмір. Оскільки трохи пізніше знадобиться ширина і висота, отримуємо їх зараз. Потім слід перетворити кадр в blob з модулем dnn.

Тепер до складного: встановлюється blob як вхідні дані в нейромережу і ці дані передаються через net, що виявляє необхідні предмети.

Після цього проходить фільтрація об'єктів. На цей момент вже знайдені шукані об'єкти в потоці. Переглядається значення на валідність і вирішення питання чи потрібно намалювати квадрат навколо об'єкта і повісити лейбл.

```
for i in np.arange(0, detections.shape[2]):
    confidence = detections[0, 0, i, 2]

    if confidence > args["confidence"]:
        idx = int(detections[0, 0, i, 1])
        box = detections[0, 0, i, 3:7] * np.array([w, h, w, h])
        (startX, startY, endX, endY) = box.astype("int")

        label = "{}: {:.2f}%".format(CLASSES[idx],
                                    confidence * 100)
```

```
cv2.rectangle(frame, (startX, startY), (endX, endY),
              COLORS[idx], 2)
y = startY - 15 if startY - 15 > 15 else startY + 15
cv2.putText(frame, label, (startX, y),
            cv2.FONT_HERSHEY_SIMPLEX, 0.5, COLORS[idx], 2)
```

Починається проходження циклами через `detections`, пам'ятаючи, що кілька об'єктів можуть бути сприйняті як єдине зображення. Також відбувається перевірка на валідність (тобто ймовірність) для кожного виявлення. Якщо валідність досить велика (тобто вище заданого порогу), відображаємо пророкування в терміналі, а також малюється на відео потоці пророкування (обводиться об'єкт в кольоровий прямокутник і вішається лейбл).

Проходимо по `detections`, отримується значення валідності.

Якщо значення валідності вище заданого порогу, витягується індекс лейбла в класі і вираховується координати рамки навколо виявленого об'єкту.

Потім, витягається $(x; y)$ - координати рамки, які будемо використовувати для відображення прямокутника і тексту.

Створюється текстовий лейбл, що містить ім'я з `CLASS` і значення валідності.

Також, малюється кольоровий прямокутник навколо об'єкта, використовуючи кольору класу і ранне витягнуті $(x; y)$ - координати.

В цілому, потрібно, щоб лейбл розташовувався над кольоровим прямокутником, однак, може виникнути така ситуація, що зверху буде недостатньо місця, тому в таких випадках виводиться лейбл під верхньою стороною прямокутника.

У кінці накладається кольоровий текст і рамку на кадр, використовуючи значення 'y', яке тільки що було вираховане.

У кінці залишається виконати наступні задачі:

- відображення кадру,
- перевірка ключа виходу,

– оновлення лічильника FPS.

```
cv2.imshow("Frame", frame)
key = cv2.waitKey(1) & 0xFF
ifkey == ord("q"):
    break
# updatethe FPS counter
fps.update()
```

Код вгорі досить очевидний: по-перше, виводиться кадр. Потім фіксується натискання клавіші, перевіряючи, чи не натиснута клавіша "q" (quit). Якщо умова істинно, ми виходимо з циклу.

Нарешті, оновлюється лічильник FPS.

Якщо відбувається вихід з циклу (натискання клавіші "q" або кінець відео потоку), з'являються речі, про які слід подбати:

```
fps.stop ()
print ("[INFO] elapsedtime: {:.2f}". format (fps.elapsed ()))
print ("[INFO] approx. FPS: {:.2f}". format (fps.fps ()))

cv2.destroyAllWindows ()
vs.stop ()
```

При виході з циклу, зупиняємо лічильник FPS і виводимо інформацію про кінцевий значенні FPS в термінал.

Закриваємо вікно програми, припиняючи відео потік.

Залишилось лише протестувати програму розпізнавання об'єктів в реальному часі на веб-камері.

Для того щоб побачити детектор об'єктів в реальному часі в дії, необхідно переконатися, що завантажені вихідні коди і попередньо підготовлено ConvolutionalNeuralNetwork.

Звідти відкриваємо термінал і виконується наступні команди:

```
$ python real_time_object_detections.py \  
--prototxt MobileNetSSD_deploy.prototxt.txt \  
--model MobileNetSSD_deploy.caffemodel  
[INFO] loading model...  
[INFO] starting videostream...  
[INFO] elapsed time: 55.07  
[INFO] approx. FPS: 6.54
```

За умови, що OpenCV може отримати доступ до веб-камери, має бути видно вихідний кадр з будь-якими виявленими об'єктами.

Приклад роботи програми показаний на рисунку 3.1.

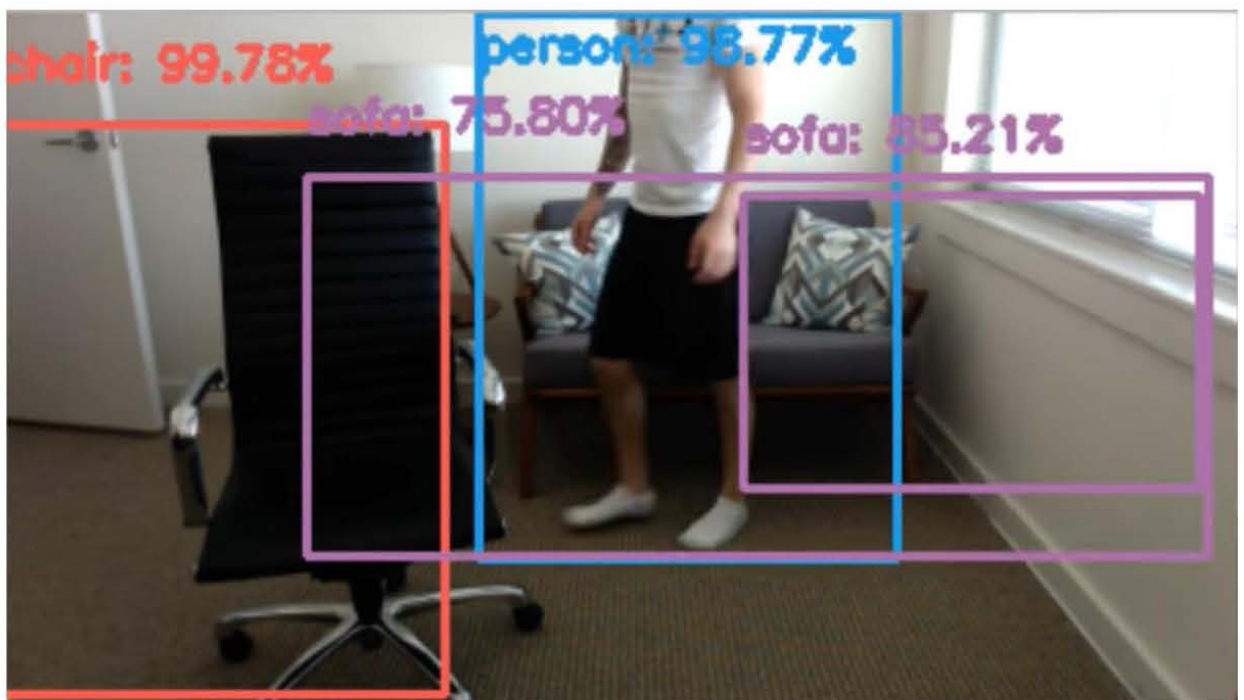


Рисунок 3.1 – результат роботи програми

3.4 Перевірка працездатності

Для реалізації програмного продукту була вибрана мова програмування Python, а також бібліотека вільного доступу OpenCV.

Для запропонованого підходу по виділенню рухомих об'єктів у різних траєкторіях переміщення у просторі позиціях було виконано тестування програми[29]. Тестування на працездатність створеної програми було здійснено для різних умов освітлення і кількості різних положень рухомих об'єктів.

Створена програма перевірялася при різних відстанях між камерою та рухомими об'єктами. Кількість вірних розпізнавань об'єктів визначалася із відношення кількості вірних результатів розпізнавання рухомого об'єкта до загальної кількості виконаних спроб по переміщенню об'єкта. Перевірка функціонування програми здійснювалася при різних значеннях рівня яскравості освітлення та різних варіантах переміщення об'єкта у просторі. Реалізований підхід по виділенню рухомих об'єктів у цифровому зображенні сцени є досить точним, його точність збільшується із зменшенням відстані між рухомих об'єктом та камерою спостереження.

Отримані результати виконаних експериментів показують, що запропонований підхід по виділенню рухомих об'єктів працює із досить непоганими результатами, які досягають 98%. Це пояснюється тим, що у запропонованому підході достовірність результатів виділення є кращою за рахунок використання пошуку точок інтересу у отриманому відео кадрі та виявлення переміщення виділеного об'єкта у зображенні, знайденим за допомогою детекторівточок та остаточно виділеним нейронною мережею.

У підсумку необхідно відзначити, що запропонований підхід по пошуку та виділенню рухомих об'єктів цифрових зображень можна застосовувати у комп'ютерних системах відеоспостереження для виділення рухомих об'єктів.

3.4 Висновок

У даному розділі було розроблено програмну частину магістерської роботи по виділенню рухомих об'єктів зображень мовою програмування Python із використанням OpenCV.

4 РОЗРАХУНОК ЕКОНОМІЧНОЇ ДОЦІЛЬНОСТІ СТВОРЕННЯ ПРОГРАМИ ВИДІЛЕННЯ РУХОМИХ ОБ'ЄКТІВ ЦИФРОВОГО ЗОБРАЖЕННЯ

У сучасних умовах відсутність впровадження інновацій в промисловості відчувається особливо гостро. Підприємства потребують швидкого впровадження досягнень науково-технічного прогресу галузі, а це звісно вимагає зменшення часу на проведення науково-дослідних робіт і скорочення строку окупності витрат. Виходячи з цього, доцільно орієнтуватися на час проведення науково-дослідних робіт та розробку експериментального зразка продукту не більше 1 року, при чому технічні показники результатів плануються на рівні кращих світових зразків; термін окупності витрат у межах 1-2 років і менше. Впродовж подальшого розвитку роботи витрати повинні поступово зменшуватися, це свідчитиме про успішну реалізацію продукту і його своєчасне удосконалення.

На основі економічних розрахунків можна довести економічну доцільність та ефективність впровадження отриманих результатів виконаних науково-дослідних робіт у виробництво, тобто здійснити так звану комерціалізацію наукових розробок [30].

Саме цим завданням присвячено даний розділ магістерської кваліфікаційної роботи і передбачає він виконання таких етапів робіт (рис. 4.1):

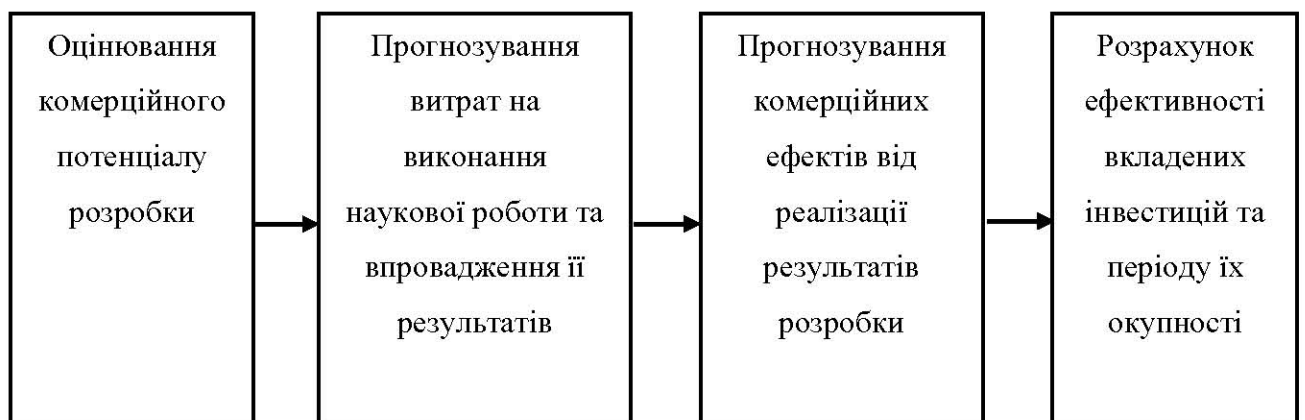


Рисунок 4.1 - Складові економічної частини магістерської кваліфікаційної роботи

Саме на такі складові буде поділено економічну частину даної магістерської роботи. Усі подальші економічні розрахунки, будуть висвітлені у згаданих підрозділах економічної частини. У комплексі ці етапи дозволять побачити цілісну картину доцільності розробки та впровадження запропонованого рішення.

4.1 Технологічний аудит розробки

Метою проведення оцінювання комерційного потенціалу розробки є оцінювання комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності. В результаті оцінювання робиться висновок щодо напрямів (особливостей) організації подальшого її впровадження з врахуванням встановленого рейтингу.

Оцінювання комерційного потенціалу розробки будемо здійснювати за 12-ма критеріями, наведеними в таблиці 4.1.

Таблиця 4.1 - Оцінювання комерційного потенціалу розробки

Критерії оцінювання та бали (за 5-ти 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 працездатність 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

Продовження таблиці 4.1

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
Ринкові перспективи					
6	Ринок малий і має негативну динаміку	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
Практична здійсненність					
7	Активна конкуренція компаній	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкуренція немає
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витратити кошти та час на навчання 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 фінансові ресурси, які відсутні. Джерела фінансування ідеї 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 інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї 3 роки. Термін окупності від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідно регламентні документи та багато дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

На основі складеної таблиці ряд незалежних експертів, у нашому випадку керівник магістерської роботи та викладачі випускової кафедри виставили різні бали. Результати цього оцінювання комерційного потенціалу внесено до таблиці 4.2.

Таблиця 4.2 - Результати оцінювання комерційного потенціалу розробки

Критерії	Прізвище, ініціали, посада експерта		
	1 Черняк О.І., к.т.н., доц. кафедри ОТ	2 Колесник І. С., к.т.н., доц. кафедри ОТ	3 Богомолов С. В., к.т.н., доц. кафедри ОТ
	Бали, виставлені експертами:		
1	3	3	3
2	2	2	2
3	2	3	2
4	3	2	2
5	3	2	2
6	3	4	3
7	0	1	0
8	3	3	3
9	1	1	1
10	4	4	4
11	3	4	3
12	3	3	3
Сума балів	$СБ_1 = 30$	$СБ_1 = 32$	$СБ_1 = 28$
Середньо-арифметична сума балів $\overline{СБ}$	$\overline{СБ} = \frac{\sum_1^3 СБ_i}{3} = \frac{30 + 32 + 28}{3} = \frac{90}{3} = 30$		

За даними таблиці 4.2, а також згідно рекомендацій, що наведені в таблиці 4.3, можна зробити висновок, щодо рівня комерційного потенціалу розробки.

Таблиця 4.3 - Рівні комерційного потенціалу розробки

Середньоарифметична сума балів $\overline{СБ}$, розрахована на основі висновків експертів	Рівень комерційного потенціалу розробки
0 - 10	Низький

11 - 20	Нижче середнього
21 - 30	Середній
31 - 40	Вище середнього
41 - 48	Високий

Взявши до уваги, середньоарифметичну суму балів, $\overline{СБ} = 30$, що були виставлені експертами, можна стверджувати, що рівень комерційного потенціалу даної розробкиє середнім.

Дана розробка буде реалізована в наступними шляхами. Спершу буде підписана ліцензована угода з компаніями, що будуть зацікавлені у покушці програмного забезпечення. Після цього програмне забезпечення буде виставлено на продаж у інтернеті серед звичайних користувачів, які захочуть купити розроблений продукт по фіксованій ціні.

Якість розробки знаходиться на достатньо високому рівні. Оскільки програмне забезпечення займає не велику кількість пам'яті, не потребує великих бюджетних затрат на технічне забезпечення, а також вона працює на усіх операційних системах.

Конкурентоспроможність розробки є середньою, оскільки декілька схожих безкоштовних програм, проте вони працюють на достатньо низькому рівні, що надає змогу зіставляти їм конкуренцію надаючи більш серйозну якість не безкоштовно.

Розроблене програмне забезпечення спрощує спосіб розпізнавання рухомих об'єктів та виділяє їх, що в свою чергу дозволяє зберегти зір та зберігає час.

Програмне забезпечення є екологічно вигідним так як не потребує жодних додаткових технічних засобів, крім комп'ютера та веб-камери. Що в свою чергу зберігає не тільки гроші але й екологію за допомогою мінімалізації витрат на електроенергію.

Програмний продукт повністю готовий до експлуатації, уся документація знаходиться у гарному стані,

Програма завантажена на сайт продажів та готова до покупок. Також ведуться переговори з зацікавленими інвесторами. На даний момент переговори знаходяться на кінцевому етапі, оскільки ще не було остаточно вирішена цінова політика та строки експлуатації.

Складним на даному етапі є не можливість продажу через сайт, оскільки по умовам контракту, що обговорюється з одною з компаній є неможливість продажу її іншим. Через це переговори ведуться з декількома компаніями.

4.2 Прогнозування витрат на виконання та впровадження результатів наукової роботи

У даній магістерській кваліфікаційній роботі розглядається програмне забезпечення для виділення рухомих об'єктів цифрових зображень, це по суті програма, а при розробці програми значна частина витрат - витрати на розробку, а не на виробництво і відтворення. Звідси, й певна специфіка розрахунків [].

1. Основна заробітна плата розробників, які працюють над проектом визначається за формулою 4.1:

$$Z_o = \frac{M}{T_p} \cdot t, \text{ (грн.)} \quad (4.1)$$

де M - місячний посадовий оклад розробника;

T_p - число робочих днів в місяці ($T_p = 22$ дні);

t - число днів роботи розробника.

Над створенням розробки працювали керівник проекту та інженер-програміст, отже, виконаємо для них всі необхідні розрахунки []:

$$Z_o = \frac{14000,00}{22} \cdot 3 = 1909,09 \text{ (грн.)}$$

$$Z_o = \frac{12000,00}{22} \cdot 44 = 24000,00 \text{ (грн.)}$$

Таблиця 4.4- Заробітна плата

Найменування посади	Місячний посадовий	Оплата за робочий день,	Число днів	Витрати на заробітну плату, грн.
1 Керівник	14000,00	636.36	3	1909,09
2 Інженер-програміст	12000,00	545,45	44	24000,00
Всього				$\sum Z_0 = 25909,1$

2. Додаткова заробітна плата Z_d всіх розробників та робітників, які брали участь у виконанні даного етапу роботи, розраховується як (10... 12)% від суми основної заробітної плати всіх розробників та робітників, тобто:

$$Z_d = (10 \dots 12\%) \cdot Z_0, (\text{грн.}) \quad (4.2)$$

де Z_0 - основана заробітна плата.

$$Z_d = \frac{10 \cdot 25909,1}{100} = 2590,91 (\text{грн.})$$

3. Нарахування на заробітну плату N_{zp} розробників та робітників, які брали участь у виконанні даного етапу роботи, розраховуються за формулою:

$$N_{zp} = 22\% \cdot (Z_0 + Z_d), (\text{грн.}) \quad (4.3)$$

$$N_{zp} = \frac{22 \cdot (25909,1 + 2590,91)}{100} = 6269,99 (\text{грн.})$$

4. Амортизація обладнання, комп'ютерів та приміщень A , які використовувались під час (чи для) виконання даного етапу роботи.

Дані відрахування розраховують по кожному виду обладнання, приміщенням тощо.

У спрощеному вигляді амортизаційні відрахування A в цілому були розраховані за формулою 4.4:

$$A = \frac{Ц \cdot N_a \cdot T}{100 \cdot 12}, (\text{грн.}) \quad (4.4)$$

де $Ц$ - загальна балансова вартість всього обладнання, комп'ютерів, приміщень тощо, що використовувались для виконання даного етапу роботи, грн;

N_a - річна норма амортизаційних відрахувань;

T - термін, використання обладнання, приміщень тощо, місяці.

Всі розрахунки зводимо до таблиці 4.5.

Таблиця 4.5 - Амортизація обладнання та приміщень

Найменування обладнання, приміщень	Балансова вартість, грн.	Норма амортизації, %	Термін використання, міс.	Величина амортизаційних відрахувань, грн.
ЕОМ	15000	10%	2	2500
Приміщення для дипломного проектування	10000	2%	2	333,00
Всього				A = 2833,0

5. Витрати на комплектуючі K , що були використані під час виконання даного етапу роботи.

Оскільки у магістерській кваліфікаційній роботі розробляється програмний продукт, який для кінцевого споживача буде розповсюджуватися через інтернет, а саме через веб-сторінку. При розробці сторінки до послуг виробничого характеру сторонніх підприємств та організацій можна віднести надавання послуги «Хостинг», а також направлення обраного доменного імені на сервери хостингу, оскільки на протязі всього існування проекту за ці послуги щорічно потрібно сплачувати абонентську плату.

Інформацію про використані послуги занесемо у таблицю 4.6.

Таблиця 4.6 - Послуги, що використовуються при виготовленні інноваційного продукту

Найменування комплектуючих (робіт, послуг)	Кількість, шт.	Ціна за одиницю, грн.	Сума, грн.
1. Послуга «Хостинг», шт.	1	150	150
2. Послуга «Доменне ім'я», шт.	1	30	30

Всього	K = 180 грн.
--------	--------------

6. Витрати на силову електроенергію V_e , якщо ця стаття має суттєве значення для виконання даного етапу роботи, розраховуються за формулою 4.5:

$$V_e = V \cdot П \cdot \Phi \cdot K_{\pi}, \text{ (грн.)} \quad (4.5)$$

де V - вартість 1 кВт електроенергії, грн.;

$П$ - установлена потужність обладнання, кВт/год;

Φ - фактична кількість годин роботи обладнання, яке задіяне на виготовлення одного виробу, годин;

K_{π} - коефіцієнт використання потужності, $K_{\pi} \leq 1$.

Вартість 1кВт електроенергії рівна 1,68 грн[], потужність обладнання (ноутбука) рівна 90 Вт, що тотожно 0,09 кВт, фактична робота обладнання для роботи рівна 336 год, а коефіцієнт використання потужності приймемо за 0,8.

$$V_e = 2,44 \cdot 0,09 \cdot 336 \cdot 0,8 = 40,64 \text{ (грн.)}$$

7. Інші витрати. Інші витрати охоплюють: загально виробничі витрати (витрати управління організацією, ремонт та експлуатація основних засобів, витрати на опалення, освітлення тощо), адміністративні витрати (проведення зборів, оплата юридичних та аудиторських послуг, тощо), витрати на збут (витрати на рекламу, перепідготовка кадрів) на інші операційні витрати (штрафи, пені, матеріальні допомоги, втрати від знецінення запасів тощо).

Інші витрати можна розрахувати за нормативами відносно основної заробітної плати основних робітників, які виготовляють продукцію, за формулою 4.6.

$$V_{in} = H \cdot Z_o, \text{ (грн.)} \quad (4.6)$$

де H - норматив загальноновиробничих витрат. Для ЕОМ $H = 230-270\%$.

$$V_{in} = \frac{250 \cdot 25909,1}{100} = 64772,75 \text{ (грн.)}$$

8. Сума всіх попередніх статей витрат дає витрати на виконання даної частини (розділу, етапу) роботи - V .

$$V = 25909,1 + 2590,91 + 6269,99 + 2833,0 + 180 + 40,64 + 64772,75 \\ = 102596,39 \text{ (грн.)}$$

9. Прогнозування загальних витрат ЗВ на виконання та впровадження результатів виконаної наукової роботи здійснюється за формулою:

$$ЗВ = \frac{V_{\text{заг}}}{\beta}, (4.7)$$

де β - коефіцієнт, який характеризує етап (стадію) виконання даної роботи. Оскільки, розробка знаходиться на стадії впровадження, то $\beta \approx 0,9$;

$V_{\text{заг}}$ - загальна вартість всієї наукової роботи. У даному випадку $V_{\text{заг}} = V$.

$$ЗВ = \frac{102596,39}{0,9} = 113995,99 (\text{грн.})$$

Отже, розрахований кошторис витрат на розробку програмного забезпечення для виділення рухомих об'єктів цифрових зображень складає 113995,99 грн.

4.3 Прогнозування комерційних ефектів від реалізації результатів розробки

У даному підрозділі виконано прогнозування, яку вигоду можна отримати у майбутньому від впровадження результатів даної наукової роботи.

Передбачається, що виконання наукової роботи та впровадження результатів по розробці програмного забезпечення для виділення рухомих об'єктів цифрових зображень займе 1 рік.

Основні позитивні результати від впровадження розробки очікуються протягом 3 років після її впровадження.

Саме зростання чистого прибутку забезпечить підприємству (організації) надходження додаткових коштів, які дозволять покращити фінансові результати діяльності.

$$\Delta\Pi_i = \sum_1^n (\Delta\Pi_{\text{я}} \cdot N + \Pi_{\text{я}} \Delta N)_i, (\text{грн.}) \quad (4.8)$$

де $\Delta\Pi_{\text{я}}$ - покращення основного якісного показника від впровадження результатів розробки у даному році;

N - основний кількісний показник, який визначає діяльність підприємства у даному році до впровадження результатів наукової розробки;

ΔN - покращення основного кількісного показника діяльності підприємства від впровадження результатів розробки;

$P_{\text{я}}$ - основний якісний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки;

n - кількість років, протягом яких очікується отримання позитивних результатів від впровадження розробки.

В результаті впровадження результатів наукової розробки покращується якість програмного продукту, що дозволяє підвищити ціну його реалізації на 100 грн., а кількість потенційних користувачів ресурсу збільшиться: протягом першого року - на 40000 шт., протягом другого року - ще на 20000 шт., протягом третього року - ще на 10000 шт.

Орієнтовно: реалізація продукції до впровадження результатів наукової розробки складала 1шт., а прибуток, що його отримувало підприємство на одиницю продукції до впровадження результатів наукової розробки - 250 грн.

Спрогнозуємо збільшення чистого прибутку підприємства від впровадження результатів наукової розробки у кожному році відносно базового.

Збільшення чистого прибутку підприємства протягом наступних трьох років складе:

Збільшення чистого прибутку ΔP_1 протягом першого року складе:

$$\Delta P_1 = 100 \cdot 1 + (250 + 100) \cdot 40000 = 1400100$$

Збільшення чистого прибутку ΔP_2 протягом другого року (відносно базового року, тобто року до впровадження результатів наукової розробки) складе:

$$\Delta P_2 = 100 \cdot 1 + (250 + 100) \cdot 20000 = 700100$$

Збільшення чистого прибутку ΔP_3 протягом третього року (відносно базового року, тобто року до впровадження результатів наукової розробки) складе:

$$\Delta P_3 = 1 \cdot 1 + (250 + 100) \cdot 10000 = 350100$$

Загальне збільшення прибутку підприємства, або комерційний ефект, від впровадження розробки за 3 роки становить:

$$\Delta\Pi = 1400100 + 700100 + 350100 = 2450300 \text{ (грн.)}$$

4.4 Розрахунок ефективності вкладених інвестицій та періоду їх окупності

Розрахований комерційний ефект від можливого впровадження розробки ще не означає, що ця розробка реально буде впроваджена. Якщо збільшення прогнозованого прибутку від впровадження результатів розробки є вигідним для підприємства, то це ще не означає, що інвестор погодиться фінансувати розробку.

Основними показниками, що визначають доцільність фінансування розробки інвестором, є абсолютна і відносна ефективність вкладених інвестицій та термін їх окупності.

Розрахунок ефективності вкладених інвестицій передбачає проведення таких робіт:

1-й крок. Розрахуємо теперішню вартість інвестицій PV , що вкладаються в наукову розробку. Такою вартістю, вважаємо прогнозовану величину загальних витрат ZB на виконання та впровадження результатів НДДКР, розраховану раніше за формулою (4.7), тобто вважатимемо, що $ZB = PV = 113995,99$.

2-й крок. Розрахуємо очікуване збільшення прибутку $\Delta\Pi_i$, що його отримає підприємство (організація) від впровадження результатів наукової розробки, для кожного із років, починаючи з першого року впровадження. Таке збільшення прибутку, також було розраховане раніше за формулою 4.8. Сумарне значення $\Delta\Pi_i = 2450300$ (грн.).

3-й крок. Для спрощення подальших розрахунків побудуємо вісь часу, на яку нанесемо всі платежі (інвестиції та прибутки), що мають місце під час виконання науково-дослідної роботи та впровадження її результатів.

Платежі показуються у ті терміни, коли вони здійснюються. Рисунок, що характеризує рух платежів (інвестицій та додаткових прибутків) буде мати вигляд, наведений на рис. 4.2.



Рисунок 4.2 - Вісь часу з фіксацією платежів, що мають місце під час розробки та впровадження результатів НДДКР

4-й крок. Розрахуємо абсолютну ефективність вкладених інвестицій $E_{абс}$.

Для цього користуються формулою:

$$E_{абс} = (ПП - PV), (4.9)$$

де ПП - приведена вартість всіх чистих прибутків, що їх отримає підприємство (організація) від реалізації результатів наукової розробки, грн;
 PV - теперішня вартість інвестицій $PV = 3B = 113,995$ грн.

У свою чергу, приведена вартість всіх чистих прибутків ПП розраховується за формулою:

$$ПП = \sum_1^T \frac{\Delta\Pi_i}{(1 + \tau)^t}, \text{ (тис. грн.)} \quad (4.10)$$

де $\Delta\Pi_i$ - збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої НДДКР, грн;

T - період часу, протягом якого виявляються результати впровадженої НДДКР, роки;

τ - ставка дисконтування [], за яку можна взяти щорічний прогнозований рівень інфляції в країні; для України цей показник знаходиться на рівні 0,3;

t - період часу (в роках) від моменту отримання чистого прибутку до точки «0».

Отримаємо:

$$ПП = \frac{1400100}{(1+0,3)^2} + \frac{700100}{(1+0,3)^3} + \frac{350100}{(1+0,3)^4} = \frac{1400100}{1,69} + \frac{700100}{2,197} + \frac{350100}{2,8561} = 828\,461,53 + 318\,661,81 + 122\,579,74 =$$

1269,702 грн.

$$\text{Тоді } E_{\text{абс}} = (1269,702 - 113,995) = 1155,707 \text{ тис. грн.}$$

Оскільки $E_{\text{абс}} > 0$, то результат від проведення наукових досліджень та їх впровадження може принести прибуток, але це також ще не свідчить про те, що інвестор буде зацікавлений у фінансуванні даної роботи.

5-й крок. Розраховують відносну (щорічну) ефективність вкладених в наукову розробку інвестицій $E_{\text{в}}$. Для цього користуються формулою:

$$E_B = \sqrt[T_{ж}]{1 + \frac{E_{абс}}{PV}} - 1, (\%) \quad (4.11)$$

де $E_{абс}$ - абсолютна ефективність вкладених інвестицій, грн;

PV - теперішня вартість інвестицій $PV = ЗВ$, грн;

$T_{ж}$ - життєвий цикл наукової розробки, роки.

$$E_B = \sqrt[3]{1 + \frac{1155,707}{113,995}} - 1 = \sqrt[3]{1 + 10,13} - 1 = 2,22 - 1 = 122 \%$$

Далі, розрахована величина E_B порівнюється з мінімальною (бар'єрною) ставкою дисконтування $\tau_{мін}$, яка визначає ту мінімальну дохідність, нижче за яку інвестиції вкладатися не будуть. У загальному вигляді мінімальна (бар'єрна) ставка дисконтування $\tau_{мін}$ визначається за формулою:

$$t = d + f, (\%) \quad (4.12)$$

де d - середньозважена ставка за депозитними операціями в комерційних банках; в 2019 році в Україні $d = (0,15...0,25)$;

f - показник, що характеризує ризикованість вкладень; зазвичай, величина $f = (0,05...0,1)$, але може бути і значно більше.

$$t = d + f = 0,25 + 0,1 = 0,3 = 35\%$$

Величина $E_B > \tau_{мін}$, інвестор може бути зацікавлений у фінансуванні даної наукової розробки.

Розрахуємо термін окупності вкладених коштів у реалізацію наукового проекту за формулою:

$$T_{\text{ок}} = \frac{1}{E_{\text{в}}}, (\text{років}) \quad (4.13)$$

$$T_{\text{ок}} = \frac{1}{1,22} = 0,82 \text{ року.}$$

Оскільки $T_{\text{ок}} = 0,82$ року (десять місяців), то фінансування розробки є доцільним.

Висновок

У четвертому розділі магістерської кваліфікаційної роботи проведено розрахунки, що доводять економічну доцільність та ефективність впровадження розробленого продукту. Розрахунки поділено на 4 частини, які утворили відповідні підрозділи цього розділу. У комплексі ці підрозділи дозволяють побачити цілісну картину доцільності нового інтелектуального рішення.

У першому підрозділі здійснено оцінювання комерційного потенціалу розробки. На основі компетентної думки експертів було сформовано систему критеріїв та за 5-ти бальною шкалою, виставлено бали кожному критерію. Виставлені бали, говорять, що рівень комерційного потенціалу є середнім.

Другий підрозділ економічної частини демонструє витрати на розробку, що розраховуються, як сума усіх статей витрат поділена на ступінь готовності продукту. Розрахований кошторис витрат на розробку складав 113995 грн.

Далі прогнозуються комерційні ефекти від реалізації розробки, тобто яку вигоду, можна отримати у майбутньому від впровадження результатів виконаної наукової роботи. Загальне збільшення прогнозованого прибутку за 3 роки становить 2450300 грн.

Останній підрозділ висвітлює основні показники, які визначають доцільність фінансування наукової розробки певним інвестором. Такими

показниками є абсолютна та відносна ефективність вкладених інвестицій, а також термін їх окупності.

Обрахована абсолютна ефективність становить 1155707 грн, що свідчить про те, що інвестор буде зацікавлений у фінансуванні даної розробки.

Відносна (щорічна) ефективність становить 122 %, що більше мінімальної ставки дисконтування, що ще раз підтверджує зацікавленість інвестора.

Термін окупності вкладених коштів у реалізацію проекту становить 0,82, що означає, що вкладені кошти повернуться через десять місяців.

Таким чином, можна стверджувати, що фінансування даної розробки є доцільним.

ВИСНОВКИ

Створення систем пошуку, виділення та розпізнавання рухомих об'єктів із використанням засобів комп'ютерного зору є актуальною задачею сьогодення. Рішення у цій області знаходять широке застосування у самих різноманітних сферах діяльності людства: відстеження появи машин на стоянці, забезпечення захисту від несанкціонованого проникнення зловмисників на різного типу об'єкти, розпізнавання руху людини за ходою, знаходження та розпізнавання обличчя людей у відео послідовності та інше. У даній магістерській кваліфікаційній роботі розроблені метод та засоби пошуку та виділення рухомих об'єктів у цифрових зображеннях шляхом пошуку точок інтересу.

1) У магістерській роботі був виконаний аналіз таких методів, як визначення переміщення областей пікселів між сусідніми кадрами, метод визначення зміщення прямокутних блоків у зображенні, метод кореляції, метод використання шаблонів, метод відстеження переміщення центру мас, розпізнавання рухомих об'єктів у відео послідовності за допомогою неймережі глибокого навчання, а також використання детекторів та дескрипторів для виділення об'єктів руху.

2) У магістерській роботі вдосконалено метод пошуку та виділення рухомих об'єктів у відео послідовності. Основним підходом до пошуку та виділення рухомих об'єктів використано пошук точок інтересу у зображенні для знаходження рухомих об'єктів, що дозволяє підвищити ефективність роботи розробленого програмного забезпечення. Для виділення ознак рухомих об'єктів використано детектори для знаходження цих об'єктів у аналізованій сцені та застосування нейронної мережі глибокого навчання для пошуку рухомих об'єктів.

3) У магістерській роботі розроблено послідовність обробки цифрового зображення для пошуку рухомих об'єктів у вибраній сцені. Запропонований у роботі підхід по використанню детекторів для виділення точок інтересу та застосування нейронної мережі глибокого навчання дозволяє виконати процес обробки комп'ютерних зображень із метою знаходження рухомих об'єктів. Для

реалізації запропонованого підходу по виділенню рухомих об'єктів створена програма.

4) У магістерській роботі також виконані економічні розрахунки із обґрунтування доцільності виконання нової роботи по виділенню рухомих об'єктів у цифрових зображеннях шляхом пошуку точок інтересу, 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 для виділення рухомих об'єктів.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1) Лукьяница А. А., Шишкин А. Г. Цифровая обработка видеоизображений– Москва: «Ай-Эс-Эс Пресс».-2009. –145с.
- 2) Шапиро Л., Штокман Дж. Компьютерное зрение. - М.: Бином, 2009. - 763с.
- 3) Заяць В. М. Методи розпізнавання образів. Навч. посібник. / В. М. Заяць, Р. М. Камінський.- Львів, видав. Національного університету «Львівська політехніка», 2004. – 176 с.
- 4) Путятин Е.П. Нормализация и распознавание изображений [Электронный ресурс] / Е.П. Путятин. Режим доступа до ресурсу: <http://sumschool.sumdu.edu.ua/is-02/rus/lectures/pytyatin/pytyatin.htm>.
- 5) Роженцов А.А. Оценка параметров и распознавание изображений трехмерных объектов с неупорядоченными отсчетами / Роженцов А.А., Наумов А.С.//Автометрия, 2010, 46, №1, С. 57-69.
- 6) Фомин Я. А. Распознавание образов: теория и применение. / Я. А. Фомин. - М.: ФАЗИС, 2012. - 429 с.
- 7) Шевчук О. В. Виділення рухомих об'єктів цифрових зображень шляхом пошуку точок інтересу *Матеріали конференції «Молодь в науці: дослідження, проблеми, перспективи (МН-2019)»*. Збірник матеріалів. - Вінниця, ВНТУ, 2019. - с. 15-16. - [Електронний ресурс]. Режим доступу https://conferences.vntu.edu.ua/public/files/mn/mn-2019_netpub.pdf. Дата звернення: листопад, 2019.
- 8) Горелик А. Л. Методы распознавания. / А. Л. Горелик, В. А.Скрипкин– 4-е изд. – М.: Высшая школа, 1984, 2004. – 262 с.
- 9) Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun. / Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks // IEEE Transactions on Pattern Analysis and Machine Intelligence. June 1 2017, № 6 (39), pp. 1137 – 1149.

10) Hosang J.H. Learning non-maximum suppression. / 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21-26 July 2017 / Curran Associates, Inc, NY, 2018, pp. 6469-6477.

11) Wei Liu, Dragomir Anguelov, Dumitru Erhan and others. SSD: Single Shot MultiBox Detector // 2016 – 16 March [Электронный ресурс]. URL: <https://storage.googleapis.com/pubtools-public-publication-data/pdf/44872.pdf>
(Дата звернення: 17.10.2019)

12) Липанов А.В. Алгоритм распознавания изображений текстур с использованием моментных признаков и методов нейронных сетей / А.В. Липанов, А.Ю. Михайлов // Системи обробки інформації: зб. наук. праць. – Х.: ХУПС, 2007. – Вип. 3 (61). – С. 49-52.

13) Herbert B. SURF: Speeded Up Robust Features / B. Herbert, Andreas Ess, Tinne Tuytelaars, Luc Van Gool // Computer Vision and Image Understanding (CVIU), Vol. 110, No. 3, 2008, pp. 346-359.

14) Moravec H. Rover visual obstacle avoidance // Proc. Intl. Joint Conference on Artificial Intelligence. – 1981. – P. 785–790.

15) Harris C. G., Stephens M. J. Combined corner and edge detector // Proc. Fourth Alvey Vision Conference. – 1988. – P. 147–151.

16) Rosten E., Drummond T. Machine learning for high-speed corner detection // Proc. European Conference on Computer Vision. – 2006. – V. 1. – P. 430–443.

17) B. Smith. SUSAN - A new approach to low level, 1997

18) SURF: Speeded Up Robust Features / H. Bay, A. Ess, T. Tuytelaars, L. Van Gool // Computer Vision and Image Understanding. – 2008. – Vol. 110, № 3. – P. 346-359.

19) CBIR: TextureFeatures [Электронный ресурс]. – Режим доступа до ресурсу:

<http://www.es.auckland.ac.nz/compsci708s1c/lectures/Glecthtml/topic4c708FSC.htm>

20) A method for traffic sign detection in an image with learning from synthetic data [Text] / A. Chigorin, G. Krivovyaz, A. Velizhev, A. Konushin // 14th

International Conference Digital Signal Processing and its Applications. –2012. – Vol. 2. – P. 316 – 335.

21) Roryorangepants. MobileNet: меньше, быстрее, точнее // – [Электронный ресурс]. URL: <https://habr.com/post/352804/>(Дата звернення: 10.10.2019).

22) Andrew G. Howard, Menglong Zhu, Bo Chen and others. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision // arXiv. – 2017 – 17 April

23) Ляшов М.В., Береза А.Н., Бабаев А.М., Коцюбинская С.А. Нейросетевая система отслеживания и распознавания объектов в видеопотоке // Современные наукоемкие технологии. – 2018. – № 12-1. – С. 102-107.

24) Nair V. Rectified linear units improve restricted boltzmann machines / Proceedings of the 27th International Conference on International Conference on Machine Learning, Haifa, Israel, June 21 - 24, 2010 / Omnipress, USA, 2010, pp. 807-814.

25) LeCun Y., Bottou L., Bengio Y., Haffner P. Gradient-based learning applied to document recognition // Proceedings of the IEEE. – 1998. – Т. 86. – №. 11. – С. 2278-2324.

26) Krizhevsky A., Sutskever I., Hinton G.E. Imagenet classification with deep convolutional neural networks // Advances in neural information processing systems. - 2012. - С. 1097-1105.

27) Lowe D.G. Distinctive Image Features from ScaleInvariant Keypoints / D.G. Lowe // International Journal of Computer Vision. – 2004. – Vol. 60, № 2. – P. 91-110.

28) OpenCV documentation [Электронныйресурс] – Режимдоступу: http://docs.opencv.org/3.0beta/doc/py_tutorials/py_video/py_meanshift/py_meanshift.html?highlight=camshift – (Датазвернення:19.10.2019).

29) Рейсиг Д. Инструменты отладки и тестирования.— СПб.: Питер, 2008. 76 с.—ISBN 978-5-91180-904-1.

30) Кавецький В. В., Козловський В. О. Причепа І. В. Економічне обґрунтування інноваційних рішень: Практикум. В.В.Кавецький, В.О.Козловський, І.В.Причепа. – ВНТУ, 2013. – 110 с.