

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

ПРОГРАМНИЙ ЗАСІБ ПОШУКУ І ВИДІЛЕННЯ ЗАДАНИХ ОБ'ЄКТІВ
У ЗОНІ ВІДЕОСПОСТЕРЕЖЕННЯ

Виконав: студент 2 курсу, групи 1KI-23м
спеціальності 123 — «Комп'ютерна інженерія»

Кирилюк О. О. Кирилюк О. О.

Керівник: д.т.н., проф. каф. ОТ

Мартинюк Т. Б. Мартинюк Т. Б.

«12» 12 2024 р.

Опонент: к.т.н., доц. каф. ІІЗ

Черноводик Г. О. Черноводик Г. О.

«12» 12 2024 р.

Допущено до захисту

Завідувач кафедри ОТ

Азаров О. Д. д.т.н., проф. Азаров О. Д.

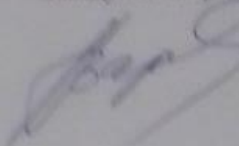
«12» 12 2024 р.

ВІННИЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра обчислювальної техніки
Галузь знань — Інформаційні технології
Освітній рівень — магістр
Спеціальність — 123 Комп'ютерна інженерія
Освітня програма — Комп'ютерна інженерія

ЗАТВЕРДЖУЮ

Завідувач кафедри обчислювальної техніки

 д.т.н., проф. О. Д. Азаров
"18" вересня 2024 р.

ЗАВДАННЯ

НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ

студенту **КИРИЛЮКУ** Олександр Олександровичу

1 Тема роботи «Програмний засіб пошуку і виділення заданих об'єктів у зоні відеоспостереження» керівник роботи Мартинюк Тетяна Борисівна д.т.н., проф. каф. ОТ, затверджено наказом вищого навчального закладу від «19» вересня 2024 року № 340.

2 Строк подання студентом роботи: 12.12.24.

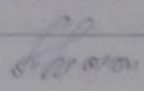
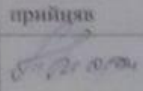
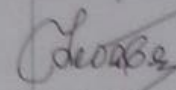
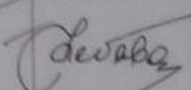
3 Вихідні дані до роботи: технічний опис сучасних методів та засобів розпізнавання об'єктів у відеозображенні, середовище розробки ПЗ Visual Studio.

4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити): вступ, аналіз методів та засобів розпізнавання об'єктів у відеозображенні, розробка технології виявлення і розпізнавання об'єктів у відеозображенні, розробка програми виявлення і розпізнавання об'єктів у відеозображенні, перевірка працездатності програмного продукту, розрахунок економічної доцільності створення програми виявлення і розпізнавання об'єктів у відеозображенні.

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень): методи розпізнавання об'єктів, типи нейронних мереж, структура програми розпізнавання об'єктів.

6 Консультанти розділів проекту (роботи)

Таблиця 1— Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	дата
		завдання видав	завдання прийняв
1-4	д. т. н., професор Мартинюк Тетяна Борисівна		
5	к. е. н., професор Небава Микола Іванович		

7 Дата видачі завдання 09.24

8 Календарний план виконання МКР приведений в таблиці 2.

Таблиця 2— Календарний план

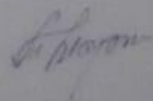
№ етапу	Назва етапів проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Пошук та огляд інформаційних джерел	03.10.2024р.	виконано
2	Огляд та аналіз існуючих підходів щодо пошуку і розпізнавання об'єктів у відеозображенні	14.10.2024р.	виконано
3	Результати проведених досліджень та методів виявлення і розпізнавання об'єктів у відеозображенні	25.10.2024р.	виконано
4	Проектування засобів виявлення і розпізнавання об'єктів у відеозображенні	19.11.2024р.	виконано
5	Розрахунок економічної доцільності створення програми виявлення і розпізнавання об'єктів у відеозображенні	02.12.2024р.	виконано
6	Попередній захист	18.11.2024р.	виконано
7	Перевірка «антиплагіат»	04.12.2024р.	виконано
8	Аналіз виконання роботи, висновки, додатки	05.12.2024р.	виконано
9	Оформлення пояснювальної записки та ілюстративного матеріалу	10.12.2024р.	виконано
10	Перевірка якості виконання магістерської роботи та усунення недоліків	12.12.2024р.	виконано

Студент



Олександр КИРИЛЮК

Керівник



Тетяна МАРТИНЮК

АНОТАЦІЯ

УДК 004.93

Кирилюк О. О. Програмний засіб пошуку і виділення заданих об'єктів у зоні відеоспостереження. Магістерська кваліфікаційна робота зі спеціальності 123 — Комп'ютерна інженерія, освітня програма Комп'ютерна інженерія. Вінниця: ВНТУ, 2024. — 119 с.

На укр. мові. Бібліогр.: 39 назв; рис. 16; табл. 8.

В магістерській кваліфікаційній роботі здійснено огляд існуючих методів та засобів виявлення та розпізнавання об'єктів у відеозображенні, визначено їх переваги та недоліки. Розроблено технологію виявлення і розпізнавання заданих об'єктів у відеозображенні, в основі якої є послідовне поєднання фільтра підкреслення контурів об'єктів і нейронної мережі для розпізнавання заданих об'єктів у обраній сцені цифрового зображення. Розроблено програмний комплекс виявлення і розпізнавання заданих об'єктів у відеозображенні на основі використання згорткової нейронної мережі. Виконано навчання нейронної мережі для виконання розпізнавання заданих об'єктів. Проведено тестування створеного програмного застосунку для визначення його продуктивності. Проведено економічний розрахунок доцільності виконання нової розробки, обчислені фінансові затрати та визначено економічні переваги від впровадження запропонованого рішення у вигляді завершеного програмного продукту.

Ключові слова: відеопотік, детектор виявлення об'єкта, згорткова нейронна мережа.

ANNOTATION

Kyryliuk O. O. Software Tool for Detecting and Highlighting Specified Objects in a Surveillance Zone. Master's Qualification Thesis in Specialty 123 – Computer Engineering, Educational Program Computer Engineering. Vinnytsia: VNTU, 2024. – 119 p.

In Ukrainian. Bibliography: 39 references; illustrations: 16; tables: 8.

The master's qualification thesis reviews existing methods and tools for detecting and recognizing objects in video images, identifying their advantages and disadvantages. A technology for detecting and recognizing specified objects in video images has been developed, based on the sequential combination of object detection detector and a neural network for recognizing specified objects within a selected scene of a digital image. A software package for detecting and recognizing specified objects in video images using a convolutional neural network has been created. The neural network was trained to perform object recognition tasks. The developed application was tested to assess its performance. An economic analysis was conducted to evaluate the feasibility of the new development, financial costs were calculated, and the economic benefits of implementing the proposed solution as a finished software product were determined.

Keywords: video stream, object detection detector, convolutional neural network.

ЗМІСТ

ВСТУП.....	8
1 АНАЛІЗ МЕТОДІВ ВИДІЛЕННЯ ТА РОЗПІЗНАВАННЯ ОБ’ЄКТІВ У ВІДЕОЗОБРАЖЕННЯХ.....	11
1.1 Системи формування ознак та розпізнавання об’єктів.....	11
1.2 Розпізнавання об’єктів як задача розпізнавання образів	16
1.3 Використання нейронної мережі для розпізнавання об’єктів	25
2 РОЗРОБКА ПОСЛІДОВНОСТІ ТА ЗАСОБІВ ВИДІЛЕННЯ І РОЗПІЗНАВАННЯ ОБ’ЄКТІВ	32
2.1 Формування послідовності розпізнавання об’єктів	32
2.2 Розробка способу виділення об’єктів зображень	36
2.3 Вибір нейронної мережі	41
3 РОЗРОБКА ПРОГРАМНИХ ЗАСОБІВ ДЛЯ ВИЯВЛЕННЯ ЗАДАНИХ ОБ’ЄКТІВ У ЗОНІ ВІДЕОСПОСТЕРЕЖЕННЯ	53
3.1 Вибір середовища та інструментальних засобів для створення програмного продукту	53
3.2 Розробка структури програми виявлення об’єктів	56
3.3 Розробка програми виділення об’єктів зображення.....	58
3.4 Вибір апаратних засобів системи відеоспостереження	61
3.5 Налаштування нейронної мережі для розпізнавання об’єктів	63
4 ПЕРЕВІРКА ПРАЦЕЗДАТНОСТІ ПРОГРАМНОГО ПРОДУКТУ ...	67
4.1 Перевірка якості роботи програми.....	67
4.2 Використання датасету для навчання нейронної мережі	69
4.3 Оцінка якості роботи програми	73

					08-54.МКР.011.00.000 ПЗ			
Змн.	Лист	№ докум.	Підпис	Дата				
Розробив		Кирилюк О.О..			Програмний засіб пошуку і виділення заданих об’єктів у зоні відеоспостереження. Пояснювальна записка	Літ.	Арк.	Аркушів
Керівник		Мартинюк Т. Б.					6	119
Опонент		Черноволик Г.О..				ВНТУ, гр. 1КІ-23м		
Н. Контроль		Швець С.І.						
Затверджую		Азаров О.Д.						

5 РОЗРАХУНОК ЕКОНОМІЧНОЇ ДОЦІЛЬНОСТІ СТВОРЕННЯ ПРОГРАМНОГО ЗАСОБУ ПОШУКУ І ВИДІЛЕННЯ ЗАДАНИХ ОБ'ЄКТІВ У ЗОНІ ВІДЕОСПОСТЕРЕЖЕННЯ	77
5.1 Оцінювання економічного потенціалу розробки.....	77
5.2 Розрахунок кошторису витрат на розробку програм	80
5.3 Розрахунок експлуатаційних витрат для нової розробки	86
5.4 Розрахунок ефективності вкладених інвестицій та періоду їх окупності.....	88
ВИСНОВКИ	94
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	96
ДОДАТОК А Технічне завдання	100
ДОДАТОК Б Методи розпізнавання об'єкта.....	104
ДОДАТОК В Типи нейронних мереж.....	105
ДОДАТОК Г Структура програми розпізнавання об'єкта	106
ДОДАТОК Д Лістинг обробки відеокcadру.....	107
ДОДАТОК Е Лістинг налаштування мережі	111
ДОДАТОК Ж Протокол перевірки кваліфікаційної роботи на наявність текстових запозичень	119

					08-54.МКР.011.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		7

ВСТУП

Розвиток інформаційних технологій, у тому числі штучного інтелекту, робить повсякденне життя людини більш оптимізованим та безпечним. Однією із задач, яку люди змогли успішно автоматизувати, є розпізнавання різних об'єктів у реальному масштабі часу. Можливість комп'ютерного виділення та розпізнавання об'єктів є основою для вирішення більш складних задач, однією з яких є визначення зміни положення об'єктів в обмеженому просторі.

Розпізнавання об'єктів є основною складовою частиною інтелектуальних систем та систем комп'ютерного спостереження [1, 2]. Сучасні методи виявлення та розпізнавання заданих об'єктів активно використовуються в машинобудуванні, у робототехніці, у медичній галузі, у системах автоматизованого управління технологічними процесами, криміналістиці, фізиці і т. д. [3, 4]. Існують багато напрямків наукових досліджень по розпізнаванню об'єктів у просторі. Вони різняться постановкою завдань та вибором способів їх вирішення залежно від сфери їх практичного використання.

Складність розв'язання таких завдань зумовлена значною кількістю форми подання об'єктів, що залежить від масштабів, положень та кутів повороту об'єктів, для яких виконується процес розпізнавання [5]. Для розв'язання цих задач існують різні методи розпізнавання, що ґрунтуються на аналізі контурів об'єктів, порівнянні за шаблоном, засновані на 3D-моделюванні, методах обчислення статистичних моментів, теорії нейронних мереж та інше [6, 7]. Важлива увага приділяється напрямку, який пов'язаний із автоматичним виділенням характерних ознак об'єктів.

Розпізнавання зображення заданих об'єктів у реальному масштабі часу є головною задачею при розробці ефективного та дієвого програмного забезпечення. Великий обсяг візуальної інформації, що широко використовується у різних сферах діяльності людства, потребує розробки нових та удосконалення вже існуючих підходів по обробленню цифрових зображень, метою якого є виділення й розпізнавання об'єктів.

Виходячи із розглянутого, задачі подальшої розробки та вдосконалення методів пошуку та виділення об'єктів цифрових зображень, що є кращими за критерієм точність розпізнавання, на теперішній час є **актуальними**.

Метою дослідження є вдосконалення методів пошуку та виділення об'єктів цифрових зображень із використанням пошуку характерних особливостей цих об'єктів.

Задачі дослідження:

- провести аналіз існуючих методів і засобів обробки цифрових зображень для пошуку об'єктів у зоні відеоспостереження;
- запропонувати удосконалений метод оброблення цифрових зображень для пошуку і розпізнавання заданих об'єктів у зоні відеоспостереження;
- розробити послідовність та створити програмний засіб оброблення цифрового зображення для виявлення заданих об'єктів у зоні відеоспостереження;
- виконати обґрунтування доцільності впровадження нового наукового рішення, зробити розрахунок економічних затрат для реалізації програмних засобів пошуку та розпізнаванню заданих об'єктів та проаналізувати економічні вигоди від впровадження нового програмного рішення.

Об'єкт дослідження — процес обробки інформації про об'єкти зображень через пошук у виділеній сцені типових ознак заданого об'єкта.

Предмет дослідження — методи обробки цифрових зображень для пошуку і виділення заданих об'єктів.

Використовувались **методи** диференційного числення, математичної статистики, методи обробки матриць, методи дискретної математики, методи теорії множин. Використовувалися засади об'єктно-орієнтованого програмування при реалізації запропонованого підходу по формуванню ознак об'єктів цифрових зображень.

Наукова новизна одержаних результатів полягає у тому, що:

— вдосконалено метод опрацювання цифрового зображення для пошуку та виділення заданих об'єктів, який відрізняється від існуючих послідовним поєднанням фільтрів підвищення різкості контурів об'єктів та застосуванням згорткової нейронної мережі для виділення заданих об'єктів у виділеній сцені відео зображення, що дозволило підвищити точність розпізнавання об'єктів цифрових зображень.

Практичне значення одержаних результатів:

— запропонована послідовність опрацювання цифрових зображень для виявлення заданих об'єктів;

— створено програмний засіб оброблення відео зображення для пошуку та виявлення заданих об'єктів з урахуванням характерних областей.

Апробація результатів роботи: здійснено доповідь на LIV Всеукраїнській науково-технічній конференції підрозділів ВНТУ.

За результатами виконання магістерської роботи **опубліковано** 1 наукову працю [8]

1 АНАЛІЗ МЕТОДІВ ВИДІЛЕННЯ ТА РОЗПІЗНАВАННЯ ОБ'ЄКТІВ У ВІДЕОЗОБРАЖЕННЯХ

Виділення та ідентифікація об'єктів широко використовуються в різних сферах, таких як робототехніка, транспорт, машинобудування, хімічна промисловість та інші галузі. На сьогодні існує багато методів створення ознак об'єктів для подальшого їхнього розпізнавання, а також засобів для їх впровадження. Цей розділ магістерської роботи присвячений розгляду підходів до виділення ознак об'єктів та методів їх ідентифікації.

1.1 Системи формування ознак та розпізнавання об'єктів

Значний розвиток інформаційних технологій сприяв виникненню нового напрямку — розпізнаванню образів. У рамках цього напрямку важливе місце займають методи виявлення різноманітних об'єктів у цифрових зображеннях. Процес розпізнавання об'єктів і образів можна розглядати з позицій інформаційної теорії як трансформацію вхідних даних у вихідні, що дозволяє класифікувати об'єкти.

З середини XX століття почалося формування напрямку, пов'язаного із створенням штучного інтелекту, що стало основою для розвитку теорії розпізнавання образів як наукового підрозділу кібернетики. Постепенно виник напрямок, що отримав назву «розпізнавання образів», де були визначені ключові поняття, терміни, математичні методи, засоби та принципи побудови систем розпізнавання. Розпізнавання образів є важливою частиною математичної кібернетики, у межах якої розробляються методи та принципи для ідентифікації й класифікації об'єктів, явищ, процесів, сигналів та ситуацій, що можуть бути описані обмеженим набором ознак і властивостей, характерних для даного об'єкта. У деяких методах терміни «клас», «таксон», «група», «кластер» використовуються як синоніми слова «образ».

Розпізнавання образів (pattern recognition) є розділом теорії штучного інтелекту (artificial intelligence), що досліджує методи класифікації об'єктів. Згідно з традицією, об'єкт, який потрібно класифікувати, називається образом

(pattern). У контексті штучного інтелекту розпізнавання образів є частиною ширшої наукової дисципліни — теорії машинного навчання (machine learning), що має на меті розроблення методів для створення алгоритмів, здатних до навчання.

Для опису об'єкта використовується n -мірний вектор, де n — кількість його одновимірних ознак. У цьому описі можливе припущення про відсутність інформації стосовно певної ознаки. Якщо потрібно класифікувати представлені об'єкти на кілька груп (образів) лише на основі їхніх описів, таке завдання називається кластерним аналізом (таксономія, самонавчання, навчання без учителя, автоматична класифікація). У випадках розпізнавання образів з навчанням (з учителем), окрім опису об'єктів, необхідні додаткові дані про належність деяких з них до певного класу (навчальна послідовність, навчальна вибірка, набір еталонів). Кількість класів зазвичай визначається заздалегідь, і класи можуть перетинатися.

Головним завданням розпізнавання образів є визначення класу, до якого належить описуваний об'єкт, виходячи з навчальної послідовності. У цьому контексті постає задача прийняття рішень, основана на аналізі раніше накопиченого досвіду. До завдань розпізнавання образів також входять мінімізація опису вихідних об'єктів та виділення інформативних ознак.

Образ визначається як класифікаційна група, що об'єднує певну кількість об'єктів за спільною ознакою. Властивість образів полягає в тому, що ознайомлення з обмеженою кількістю подій однієї й тієї ж множини дозволяє визначити, скільки представників цієї множини існує. Образи мають певні об'єктивні властивості, оскільки різні люди, навіть навчаючись на різних матеріалах, зазвичай незалежно класифікують однакові об'єкти подібним чином.

Універсальна множина ділиться на частини-образи у типовій постановці задачі розпізнавання. Будь-яке відображення об'єктів на сприймаючі органи системи розпізнавання називається зображенням об'єкта, незалежно від його положення щодо цих органів. Кількість таких зображень, об'єднаних певними загальними властивостями, становить образи.

Системи розпізнавання образів можна класифікувати за кількома критеріями: кількість рівнів датчиків ознак, сукупність ознак, відношення до навчання та методи, що лежать в основі функціонування системи розпізнавання.

За кількістю рівнів використання датчиків ознак виділяють однорівневі та багаторівневі системи розпізнавання образів [7].

Однорівневі системи розпізнавання мають один рівень датчиків ознак, які вимірюють характеристики невідомих об'єктів і передають ці дані до блоку алгоритмів розпізнавання.

Багаторівневі складні системи складаються з кількох рівнів датчиків ознак. Перший рівень включає розпізнавальні локальні датчики нижчого рівня, які обробляють частину ознак і надсилають їх на вищі рівні. Другий рівень формується з датчиків (розпізнавальних пристроїв), призначених для обробки, фільтрації та об'єднання ознак нижчого рівня, а також для визначення нових ознак, які можуть бути передані на третій або наступні рівні.

За сукупністю ознак системи розпізнавання образів поділяються на прості та складні [7]. Прості системи розпізнавання працюють з фізично однорідною інформацією. Наприклад, автоматичні пристрої розпізнавання, у яких ознаки робочого словника обмежуються лише лінійними розмірами елементів (літер), або розмінні автомати для монет, що розпізнають монети за масою. Пристрій для визначення несправності деталей може використовувати різницю в лінійних розмірах або масі як ознаку браку.

Складні системи розпізнавання, в свою чергу, оперують фізично неоднорідною інформацією. Наприклад, у медичних обстеженнях можуть використовуватися такі показники, як температура тіла, частота пульсу, тиск та дані кардіограми.

За методами розпізнавання системи можна поділити на детерміновані, імовірнісні, логічні, лінгвістичні (структурні), нейроноподібні та експертні системи [9]. Детерміновані системи використовуються для створення алгоритмів розпізнавання та застосовують «геометричні» міри близькості. Вони базуються

на дослідженні відстаней між кількісними ознаками об'єктів та класовими стандартами.

Ймовірнісні системи використовують ймовірнісні методи в рамках теорії статистики для створення алгоритмів розпізнавання. Для їх роботи потрібні ймовірнісні залежності між класами і ознаками об'єктів.

Логічні системи базуються на дискретному аналізі та обрахуванні висловлень. У таких системах потрібно вирішити систему булевих рівнянь, використовуючи логічні ознаки об'єктів, щоб визначити, до яких класів ці об'єкти належать. Логічні методи розпізнавання образів ґрунтуються на алгебрі логіки, управляючи інформацією, що міститься в окремих ознаках та їх комбінаціях. Значення ознак у цих методах розглядаються як елементарні події. Угрупування логічних методів досить велике і включає різні рівні складності та аналізу. Вони є досить трудомісткими, тому до них висуваються значні вимоги для організації обчислювального процесу. Логічні методи ефективно працюють на малих розмірах простору ознак і за допомогою сучасних комп'ютерів.

Лінгвістичні (структурні) системи потрібні для розроблення алгоритмів розпізнавання та використовують речення, які описують структуру об'єкта за допомогою атомарних елементів. Ці речення формують спеціальну мову, за допомогою якої класифікація об'єктів здійснюється через порівняння речень невідомих об'єктів зі стандартними реченнями класів. Лінгвістичні методи базуються на спеціальних граматиках, що описують множину характеристик розпізнаваних об'єктів. Для різних класів формуються стандартні атомарні елементи та можливі зв'язки між ними. Граматика визначає правила побудови об'єктів із цих елементів, в результаті чого кожен об'єкт є комбінацією атомарних елементів, поєднаних різними способами з урахуванням певних граматичних правил. Синтаксичний аналіз «пропозиції» дозволяє перевірити її синтаксичну точність та з'ясувати, чи може певна граматика, що описує клас, підтвердити наявний опис об'єкта. Однак задача визначення граматик для заданої множини висловлювань, що описують об'єкти, часто не піддається формалізації.

Нейроноподібні системи використовують моделі нейронів для розпізнавання образів.

Експертні системи застосовуються, коли існують евристичні або інтуїтивні методи рішень, але немає точних розрахунків чи алгоритмів. Експертна система є сукупністю комп'ютерних програм, що містять знання експертів у певній галузі та здатна класифікувати об'єкти, надавати рекомендації або поради, а також відповідати на запитання, запитуючи додаткову інформацію за потреби.

Найвищий рівень цих систем є логічним, але результати в ньому формуються не через порівняння отриманої інформації з вхідною апріорною, а за допомогою методів аналогії, індукції та дедукції — підходів, які зазвичай притаманні лише людині. Логічні висновки, що отримуються в цьому процесі, мають призводити до створення нових речень і знань, які доповнюють базу даних.

Системи розпізнавання образів можна класифікувати за методом навчання на три категорії: без навчання, з навчанням і самонавчанням [4, 7].

Системи без навчання мають достатню інформацію для створення початкового апріорного словника та алфавіту класових ознак. У цьому випадку достатньо визначити відстань між класами, а також основні правила та межі, що дозволяють розділити множину ознак на класи. Зазвичай для їх побудови необхідна повна первісна інформація у вигляді набору даних.

У системах із навчанням початкової інформації достатньо для визначення початкового класового алфавіту та словника ознак, але недостатньо для опису класів мовою ознак. Первинна інформація дозволяє розрізнити конкретні об'єкти, які належать до різних класів. «Вчитель» демонструє системі канонічні образи певного класу кілька разів. Якщо система надає неправильну відповідь, коефіцієнти підсилення сигналів коригуються за певним алгоритмом, поки не буде отримано правильний результат. Після перевірки система переходить до самостійного управління.

Класифікація без учителя (unsupervised classification), також відома як кластеризація або кластерний аналіз, передбачає поділ вибірки об'єктів на

неперетинаючі підмножини (кластери), які формуються на основі схожості між об'єктами. Для ефективної кластеризації об'єкти з різних кластерів повинні суттєво відрізнятися один від одного.

Задача класифікації з учителем (supervised classification) полягає у формуванні вирішального правила на основі навчальної вибірки, щоб зменшити кількість помилок у розпізнаванні. У цьому випадку навчальна вибірка виступає у ролі «вчителя».

У системах із самонавчанням початкової інформації достатньо лише для визначення словника ознак, але недостатньо для класифікації об'єктів. На етапі формування системі ілюструються початкові об'єкти на основі значень своїх ознак. Оскільки початкова інформація є обмеженою, система не отримує вказівок про клас, до якого належать ці об'єкти. У процесі самонавчання система формує класифікацію на основі ряду правил, якими керується в майбутньому. Головною метою самонавчання є здобуття достатньої кількості інформації для повноцінного функціонування системи розпізнавання. Для цього важливо мати зворотний зв'язок, який дозволяє уточнювати та коригувати накопичену апіорну інформацію.

1.2 Розпізнавання об'єктів як задача розпізнавання образів

Розпізнаванням образів називається процес, який полягає у побудові та використанні формальних методів для роботи з символічними або числовими представленнями об'єктів реального чи ідеального світу. Мета цього процесу полягає у встановленні відносин еквівалентності між цими об'єктами. Такі відносини визначають приналежність оцінюваних об'єктів до певних класів, що виступають як незалежні семантичні одиниці.

Під час розробки алгоритмів розпізнавання класи еквівалентності можуть бути надані розробником, який користується власним розумінням або залучає зовнішню інформацію про відмінності та подібності об'єктів у межах конкретної задачі. У такому випадку йдеться про «розпізнавання з учителем». Якщо ж автоматизована система вирішує задачу класифікації без залучення зовнішньої

навчальної інформації, цей процес називають автоматичною класифікацією або «розпізнаванням без учителя».

Багато алгоритмів розпізнавання образів вимагають значних обчислювальних ресурсів, що можуть забезпечити лише високопродуктивні комп'ютерні системи.

Ряд авторів розрізняють параметричні, непараметричні й евристичні методи, інші автори розділяють на такі групи відомі методи, розглядаючи історично сформовані напрямки в даній області. Розрізняють таку типологію методів розпізнавання образів [9]:

- методи, засновані на використанні апарату алгебри логіки;
- статистичні методи;
- методи, побудовані на основі «потенційних функцій»;
- методи, засновані на принципі поділу;
- методи обчислення оцінок шляхом голосування.

Евристичний підхід ґрунтується на знаннях і досвіді розробника, які важко формалізувати. Саме дослідник надає інформацію про те, як система може досягти необхідної мети розпізнавання, використовуючи певні евристики.

Методи дискримінантного аналізу включають групу підходів, заснованих на порівнянні щільності розподілу значень ознак. Серед сучасних параметричних методів особливо виділяється байєсівський підхід до прийняття рішень. Він припускає, що закон розподілу ймовірностей відомий, і потрібно лише оцінити кілька параметрів, таких як коваріаційні матриці та вектори середніх значень. Однак одним з основних викликів цих методів є потреба зберігати всі дані навчальної вибірки для точного оцінювання локальної густини ймовірностей.

Операції пошуку схожостей і відмінностей між об'єктами є спільними для багатьох методів розпізнавання. У певних підходах ці об'єкти виступають як діагностичні прецеденти, значення яких може варіюватися від ключового до другорядного в залежності від конкретного завдання.

Метод порівняння з прототипом є одним з найпростіших екстенсiональних методiв i застосовується тодi, коли класи утворюють компактнi групи в просторi ознак. Прототипом вважається об'єкт, розташований найближче до центру цiєї групи. Для класифiкацiї невідомого об'єкта система знаходить найближчий прототип i зараховує цей об'єкт до вiдповiдного класу. Для вимiрювання близькостi можуть використовуватися рiзнi метрики, наприклад, вiдстань Хеммінга, яка для дихотомiчних ознак вiдповiдає квадрату евклiдової вiдстанi.

Алгоритми, що використовуються для оцiнки об'єктiв, по-рiзному оперують їх описами. В таких алгоритмах об'єкти можуть одночасно перебувати в кiлькох пiдпросторах простору ознак. Завдання пошуку вiдмiнностей i подiбностей мiж об'єктами у цьому пiдходi формулюється як параметричне. Видiляється окремий етап налаштування алгоритмiв на основi навчальної вiбiрки, пiд час якого пiдбираються оптимальнi значення параметрiв. Як критерiї якостi використовується помилка розпiзнавання, а параметрами є:

- правила обчислення близькостi об'єктiв за окремими ознаками;
- правила обчислення загальної близькостi об'єктiв у всiх пiдпросторах ознак;
- значимiсть кожного еталонного об'єкта як дiагностичного прецеденту;
- вагомiсть внеску кожної ознаки в загальну оцiнку подiбностi об'єкта до класу для розпiзнавання.

Параметри зазвичай представляються у виглядi ваг або порогiв. Проте iз збiльшенням складностi завдань цi алгоритми стикаються з проблемами, оскiльки для роботи з великими розмiрностями необхідно вводити евристичнi припущення i обмеження.

У загальному випадку простiр ознак може бути розбитий на областi, де кожна область вiдповiдає об'єктам одного класу. У такому випадку задача розпiзнавання зводиться до пошуку меж, що роздiляють цi областi, для прийняття рiшення на основi навчальної вiбiрки ознак. Навчання завершується, коли всi роздiловi межi мiж класами побудованi. Це досягається, якщо ознаки,

що відповідають об'єктам одного класу, компактно розміщені у просторі і формують локально обмежену область.

Для вирішення задачі розпізнавання образів необхідно здійснити два ключові етапи. Спершу потрібно розбити простір ознак на області, кожна з яких відповідатиме конкретному класу об'єктів. На цьому етапі будується модель, яка дозволяє провести чітке розмежування між класами. Другим етапом є визначення, до якого з цих класів належить конкретний образ, що був розпізнаний. Це передбачає аналіз ознак об'єкта і його віднесення до однієї з виділених областей, відповідно до правил класифікації, побудованих на першому етапі.

Розпізнавання образів — це науковий напрям, що займається дослідженням, розробкою та створенням систем, призначених для визначення, до якого з наперед визначених класів належить певний об'єкт. Кожен об'єкт характеризується сукупністю ознак, а також додатковим параметром S , який вказує на його приналежність до одного з класів. Об'єкти, для яких відомі як характеристики S , так і ознаки X , утворюють навчальну вибірку. Вона використовується для виявлення зв'язків між цими характеристиками. Контрольна вибірка складається з об'єктів, у яких параметр S невідомий. Основним завданням розпізнавання образів є пошук вирішальної функції D , що визначає приналежність об'єкта з контрольної вибірки до одного з класів на основі його ознак X .

Задача розпізнавання образів полягає в тому, щоб, відокремивши суттєві ознаки, класифікувати вихідні дані, виділивши їх з-поміж інших. Це одна з базових проблем у розробці інтелектуальних систем, і вона має вагоме практичне значення, оскільки використовується в різноманітних системах для розпізнавання об'єктів.

У цьому контексті розпізнавання полягає в розподілі універсальної множини на частини (образи) — відповідно до виявлених ознак. Відображення будь-якого об'єкта на сенсорах системи, незалежно від його положення в

просторі, називається зображенням об'єкта. Якщо зображення об'єктів мають спільні властивості, вони розглядаються як образи цих об'єктів.

Теорія розпізнавання образів (pattern recognition) є однією з галузей штучного інтелекту, яка займається дослідженням і розробкою методів класифікації об'єктів. Об'єкт, що підлягає класифікації, називається "образом" (pattern), і прикладами можуть бути букви або цифри (в задачах розпізнавання символів), цифрові фотографії (в розпізнаванні зображень), а також інші типи даних.

У сфері штучного інтелекту розпізнавання образів є частиною ширшої наукової дисципліни — машинного навчання (machine learning), основною метою якого є створення алгоритмів, здатних до самонавчання.

Теорія розпізнавання образів розробляє системи, які визначають, до якого з попередньо визначених класів належить об'єкт, базуючись на його характеристиках. Об'єкти можуть бути різними — предметами, сигналами, процесами чи явищами — і описуються сукупністю ознак NNN та характеристикою MMM , що визначає їх клас. Множина об'єктів з відомими ознаками та характеристиками створює навчальну вибірку, яка використовується для побудови класифікатора.

Одне з головних завдань розпізнавання образів — це розробка вирішальної функції D , яка, на основі значень ознак N , класифікує об'єкт до одного з класів. Кожен образ описується набором числових ознак (features), а впорядкований набір таких ознак називається вектором ознак (feature vector). Цей вектор відповідає певній точці в просторі ознак (feature space). Класифікатор або вирішальне правило (decision rule) — це функція, що визначає приналежність вектора ознак до певного класу [4].

Процес розпізнавання образів можна розділити на кілька основних етапів:

- генерування ознак (feature generation) — це обчислення або вимірювання числових ознак, які характеризують об'єкт, що підлягає розпізнаванню;

— вибір ознак (feature selection) — це відбір найбільш інформативних ознак для класифікації, а до вибору можуть входити як початкові ознаки, так і їхні похідні чи функції;

— побудова класифікатора (classifier construction) — це створення вирішального правила (алгоритму), на основі якого здійснюється класифікація об'єктів;

— оцінка якості класифікації (classifier estimation) — це визначення показників ефективності класифікації, таких як точність, чутливість, специфічність, а також помилки першого і другого роду.

Образ (pattern) — це модель, яка відображає властивості об'єкта, що розпізнається. Він описується вектором ознак (feature vector), що є впорядкованою сукупністю ознак розпізнавання, представлених у вигляді вектора-рядка або вектора-стовпця.

Метрика — це міра для визначення відстані між об'єктами у просторі ознак. Чим менша відстань між об'єктами, тим більше вони схожі. Вибір метрики впливає на точність розпізнавання, адже той самий алгоритм може давати різні результати залежно від обраної міри схожості.

Існує кілька популярних метрик [10]:

— евклідова відстань, яка вимірює схожість між об'єктами, визначаючи відстань між ними в просторі ознак і чим менша ця відстань, тим більше об'єкти схожі;

— квадрат евклідової відстані підсилює вплив об'єктів, що знаходяться далеко один від одного, на результат розпізнавання;

— манхеттенська відстань (лінійна міра), що зменшує вплив великих різниць між окремими ознаками;

— узагальнена степінна відстань, яка дозволяє налаштувати міру схожості, збільшуючи або зменшуючи вагу ознак, що мають великі відмінності.

Таким чином, вибір метрики і способу вимірювання відстані має вирішальний вплив на ефективність і достовірність класифікації об'єктів.

Процес формування бажаної реакції системи на зовнішні сигнали шляхом багаторазового подання коригуючих впливів називається навчанням. Алгоритм навчання визначає механізм цього коригування. Якщо системі не надається інформація щодо правильності її реакцій, цей процес називається самонавчанням.

Розпізнавання образів полягає у віднесенні початкових даних до певного класу через виділення ключових ознак або властивостей серед загальної маси даних. Найчастіше вихідним матеріалом є цифрове зображення, отримане з камери. Завдання розпізнавання можна сформулювати як обробку векторів ознак для різних класів на основі цього зображення. Така обробка розглядається як процес кодування, при якому кожній ознаці зі простору ознак присвоюється значення, що відповідає певному класу.

Одне з ключових завдань — вибір характерних ознак з початкових зображень. Це належить до етапу попередньої обробки. Ознака повинна відображати властивість, яка є спільною для об'єктів одного класу. Важливими є також ознаки, що описують відмінності між класами, відомі як міжкласові ознаки. Завданням є вибір і формування таких ознак для створення системи розпізнавання.

Після вибору ознак необхідно розробити оптимальний процес класифікації. У цьому контексті завдання розпізнавання об'єктів можна розглядати як задачу побудови меж, що розділяють класи в просторі ознак на основі отриманих векторів вимірювань. Вирішення задач розпізнавання часто вимагає оцінювання низки параметрів, і при виборі ознак можна використовувати додаткову інформацію про природу об'єктів для покращення класифікації.

Порівняння об'єктів можна здійснювати на основі їхнього представлення у вигляді векторів вимірювань. Результати таких вимірювань зручно подавати у формі дійсних чисел. У цьому випадку схожість векторів ознак двох об'єктів можна визначити за допомогою евклідової відстані.

Успіх розпізнавання образів у великій мірі залежить від правильно вибраних ознак X . Первісний набір характеристик часто є досить обширним, тому для прийняттого рішення доцільно використовувати лише невелику кількість ключових ознак, які мають найбільше значення для виявлення відмінностей. Таким чином, проблема вибору інформативних ознак є однією з основних та важливих частин процесу розпізнавання образів.

Пошук зображення за зразком є частиною більш складної задачі розпізнавання образів. Цей процес може займати значний час і не завжди призводить до успішного результату, особливо в умовах несистематизованого та ненаправленого пошуку об'єктів серед великої кількості. Об'єкти можуть мати різноманітні ідентифікаційні ознаки, такі як форма, колір, положення, рухливість та їх комбінації. Залежно від цих характеристик, об'єкти можна класифікувати. Як правило, завдання полягає не в загальному розпізнаванні всіх об'єктів, а в ідентифікації конкретних типів об'єктів у потоці відеозображень. Найбільш поширеними класифікаційними ознаками є:

Класифікація за формою. Для виділення об'єктів за формою важливо визначити класифікаційні примітиви. У багатьох методах пошуку такими примітивами є кола, квадрати, еліпси або прямі лінії [12].

Пошук за шаблоном. Пошук за шаблоном є універсальним методом для визначення об'єктів за формою. Шаблон, що представляє форму об'єкта, переміщується по цифровому зображенню, при цьому для кожної позиції розраховується характеристика. Якщо ця характеристика перевищує заданий поріг, можна припустити наявність об'єкта з визначеною формою. Найпоширенішим способом є обчислення середньоквадратичної різниці яскравості між шаблоном і аналізованим кадром. Проте метод має суттєві недоліки, включаючи високу ресурсомісткість через багаторазове звернення до тих самих частин зображення. Більш того, шаблон може не бути достатньо адаптивним: якщо об'єкт у кадрі трохи більший або менший, його, найімовірніше, не вдасться розпізнати. Одним із можливих рішень є використання аналітичного опису форми об'єкта [13].

Пошук за аналітичним описом. У цьому методі об'єкти описуються математично, наприклад, еліпс або коло можуть бути виражені відомими формулами з аналітичної геометрії. Аналізуються характеристики багатьох точок зображення, включаючи значення яскравості, які повинні відповідати аналітичній залежності. Однак реальність така, що об'єкт рідко з'являється у кадрі у своїй ідеальній формі. Часто деякі елементи можуть бути неповними, а об'єкт може бути повернутий, що ускладнює його виявлення. У таких випадках можна виділяти окремі сегменти форми, наприклад, лінійні відрізки.

Класифікація за положенням. Методи пошуку за розташуванням є більш евристичними в розпізнаванні образів. Якщо на об'єкті є помітні деталі або мітки, їх легше ідентифікувати та класифікувати, ніж весь об'єкт загалом. Виявивши ці деталі, можна легко ідентифікувати весь об'єкт. Наприклад, виявлення очей або носа у обличчі дозволяє зробити припущення про інші частини об'єкта.

Класифікація за кольором. Багато об'єктів можна класифікувати на основі їх кольору, оскільки вони можуть мати певне забарвлення. Існує безліч колірних базисів (RGB, YUV, YCrCb, HSV тощо), в яких об'єкти можуть бути класифіковані практично безпомилково. Проте правильний вибір базису та організації пошуку залежить від конкретного випадку і часто вимагає експериментального підходу.

Дія розпізнавання образів об'єктів полягає у визначенні областей у просторі через аналіз описів множини об'єктів, для яких відомі їх справжні класи. Класифікація об'єктів — це процес, який дозволяє віднести об'єкт до певного класу лише тоді, коли його опис потрапляє у відповідну область цього класу. Класифікація є коректною, якщо об'єкт дійсно належить до зазначеного класу. Отже, процес розпізнавання об'єктів включає дві ключові дії: розпізнавання образів (формування класів об'єктів) та класифікацію об'єктів (правила віднесення об'єктів до різних класів) [14].

У загальному випадку, область ознак складається з кількох підобластей, кожна з яких містить точки, що відповідають об'єктам одного класу. Завдання

розпізнавання можна розглядати також як побудову меж між областями рішень, які відокремлюють класи, на основі зареєстрованої множини векторів-ознак. Ці межі можуть бути визначені функціями, які розділяють об'єкти за класами. Таким чином, якщо об'єкт відповідає певним умовам, він належить до відповідного класу.

Основна проблема полягає в знаходженні цих граничних поверхонь, для чого необхідна навчальна вибірка об'єктів. Процес навчання вважається завершеним, коли всі роздільні кордони між множинами образів (класів) визначені. Це зазвичай вдається реалізувати за умови компактності, коли кожна група (кластер) векторів-ознак, що належать одному класу, формує компактну (локально обмежену) область у просторі опису. Якщо кластери, що відповідають різним класам, знаходяться на достатній відстані один від одного, можна використовувати прості схеми розпізнавання, такі як класифікація об'єкта за «середньою відстанню» до всіх елементів навчальної вибірки відповідних образів або за відстанню до центру ваги кластерів тощо.

1.3 Використання нейронної мережі для розпізнавання об'єктів

Штучні нейронні мережі — це обчислювальні системи, які функціонують подібно до біологічних нейронних мереж, що складають мозок людей і тварин. Ці системи здатні навчатися виконувати різні завдання та поступово покращувати свою ефективність. Основу штучних мереж складає серія зв'язків, які називаються штучними нейронами; вони нагадують біологічні нейрони, що присутні в людському мозку. Кожне з'єднання, схоже на синапс, передає сигнал від одного нейрона до іншого. Отримавши сигнал, штучний нейрон може його обробити та передати далі іншим підключеним нейронам [15].

У звичайних реалізаціях штучної мережі сигнал, що передається між нейронами, є дійсним числом, а вихід кожного нейрона обчислюється за допомогою нелінійної функції, яка сумує його входи. Штучні нейрони та їх зв'язки зазвичай мають вагу, що коригується під час навчання мережі. Ця вага змінює величину сигналу, що проходить через з'єднання. Крім того, штучні

нейрони можуть мати певний поріг: сигнал буде передаватися далі лише тоді, коли загальна сума перевищує цей поріг. Нейрони в мережі зазвичай організовані у шари, і різні типи шарів можуть виконувати різні види перетворень сигналів на своїх входах. Сигнали переміщуються від першого (вхідного) шару до останнього (вихідного) після проходження через кілька типів шарів.

Один із видів штучних нейронних мереж — це згорткова нейронна мережа (CNN). Вона являє собою клас глибоких мереж, які безпосередньо поширюють сигнали і складаються з одного або кількох згорткових шарів, з'єднаних з шарами, що повністю взаємопов'язані (аналогічними традиційним шарам штучних мереж). Ця архітектура дозволяє згортковій нейронній мережі зберігати ефект двовимірної структури вхідних даних, які вона обробляє [16].

Згорткова нейронна мережа (CNN) є ефективним інструментом для обробки візуальних та інших двовимірних даних, що сприяє успішному вибору та розпізнаванню об'єктів у цифрових зображеннях, і є частиною технологій глибокого навчання. Ці мережі продемонстрували відмінні результати в обробці зображень і мовлення. Вони можуть бути налаштовані за допомогою стандартного методу зворотного розсіювання. Згорткова нейронна мережа є більш простим варіантом навчання у порівнянні з традиційними глибокими нейронними мережами прямого поширення, оскільки має менше параметрів для налаштування та оцінки.

Мережа ZNM — це тип багатошарової нейронної мережі, назва якої походить від операції згортки, що широко використовується в цифровій обробці зображень. У цій операції fff представляє матрицю виведення цифрового зображення, а ggg — матрицю ядра згортки. Операція згортки полягає в тому, що вікно з розміром ядра ggg проходить по всьому цифровому зображенню fff з певним кроком (зазвичай 1), і на кожному кроці виконується поелементне множення між вибраним вікном та ядром ggg , а результати збираються в таблицю.

Основна ідея побудови згорткових нейронних мереж полягає в чергуванні шарів згортки та підвибірки. Структура такої мережі односпрямована, без

зворотного зв'язку і багатошарова. Для навчання мережі використовується метод зворотного поширення помилки. Робота мережі ґрунтується на принципі переходу від конкретних рис аналізованого зображення до виявлення більш абстрактних деталей та концепцій вищого рівня. Зазвичай згорткова мережа самостійно налаштовується і створює необхідну ієрархію абстрактних ознак, відкидаючи незначні деталі та підкреслюючи важливі характеристики.

Модель згорткової мережі складається з трьох типів шарів: згорткового шару, шару підвибірки та «нормального» шару нейронної мережі у формі персептрона (рис. 1.1) [16]. Архітектура згорткових нейронних мереж реалізує три основні принципи, які забезпечують незмінність мережі до незначних зсувів, масштабування та спотворень. По-перше, кожен нейрон мережі отримує вхід від локальних рецептивних полів нейронів попереднього шару; по-друге, кожен прихований шар мережі складається з набору карт ознак, де всі нейрони мають спільні ваги, що забезпечує інваріантність до зсувів і зменшує загальну кількість ваг у мережі; по-третє, за кожним згортковим шаром розташований обчислювальний шар, який локально усереднює та виконує підвибірки, гарантуючи, що карти характеристик мережі не розширюються.

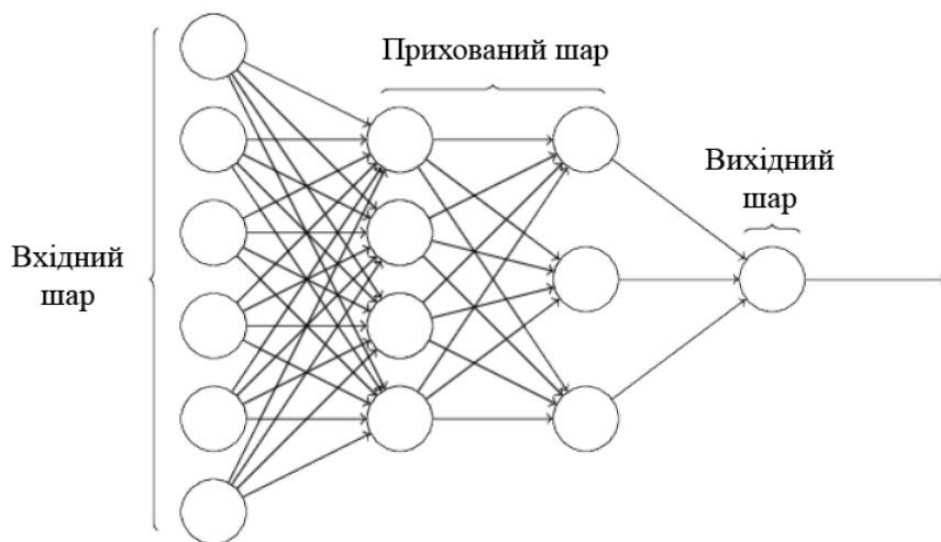


Рисунок 1.1 — Структура нейронної мережі

Функціонування згорткової нейронної мережі базується на двох основних компонентах: фільтрах для виявлення ознак та картах ознак. Фільтр представляє собою невелику матрицю, яка відповідає ознакам вихідного вхідного зображення. Верхній фільтр аналізує частини зображення з вертикальними лініями, а нижній — з горизонтальними. Процес виявлення ознак ґрунтується на створенні конкретного фільтра виходу для зображення. Результати згортки, що показують розташування об'єктів на вихідному зображенні, називаються картами об'єктів [17].

Основна мета згортки полягає в зменшенні розмірності карти ознак до такого рівня, щоб повний набір ознак міг успішно працювати з мережею прямого поширення (зазвичай використовується багатошаровий персептрон). Згортковий шар мережі реалізує концепцію локальних рецептивних полів, де кожен вихідний нейрон підключений лише до невеликої області вхідної матриці, що моделює деякі аспекти людського зору.

Персептрон є штучним нейроном, який імітує повністю з'єднану нейронну мережу, де кожен наступний нейрон з'єднаний із усіма нейронами попереднього шару. Всі з'єднання в цій мережі мають свої ваги. У згортковій мережі операція згортки виконується за допомогою невеликої вагової матриці, яка переміщується по вихідному цифровому зображенню, генеруючи специфічний сигнал активації для наступного нейрона з таким же положенням. Ця вагова матриця для вхідних нейронів використовується як специфічне ядро згортки. Її відображення є графічним кодуванням певної ознаки цифрового зображення. Результируючий мережевий рівень, отриманий внаслідок цієї згортки через вагову матрицю, вказує на наявність конкретної ознаки в цьому мережевому рівні та її координати. У результаті формуються карти мережевих об'єктів (об'єктні карти).

У згорткових мережах існує не один набір ваг, а кілька елементів, які кодують різні аспекти аналізованого цифрового зображення (наприклад, дуги або лінії під різними кутами). Ці ядра згортки є унікальними для мережі, оскільки створюються під час процесу навчання і не налаштовуються вручну оператором.

При проходженні згорткової мережі через серію ваг формується набір унікальних ознак, що робить цю мережу багатоканальною. Під час обробки мережевого шару вагова матриця зміщується не на весь розмір матриці, а на невелику відстань. Наприклад, якщо матриця ваги має розмір 5×5 , вона буде зміщена на один або два нейрони замість п'яти, щоб не втратити або не пропустити важливі властивості.

Підвибірка (іноді також злиття) використовується для зменшення розміру згенерованих карт об'єктів. Після виконання цієї операції важливо не лише визначити, чи присутня шуканий ознака в певному фрагменті цифрового зображення, але й отримати його координати. Для цього вибирається максимальне значення з сусідніх нейронів мережі за їх картами ознак, і це значення тепер називається мережевим нейроном, який формує нову компактну карту об'єктів з меншими розмірами. Ця операція прискорює обчислення та робить згорткову мережу менш чутливою до розмірів вхідного цифрового зображення.

Типова структура згорткової мережі складається з багатьох шарів. Вхідне цифрове зображення подається на верхній шар мережі, після чого сигнали проходять через серію згорткових шарів, де виконуються операції згортки та підвибірки. Чередування мережевих шарів дозволяє створювати карти ознак, при цьому в кожному наступному шарі карта зменшується, але кількість мережевих каналів зростає. Після проходження кількох шарів одночасно карта об'єктів перетворюється на скалярний або векторний масштаб, створюючи при цьому сотні карт об'єктів. Після згорткових шарів до мережі додаються кілька шарів персептрона як повністю з'єднаної нейронної мережі, вхідні дані якої подаються до остаточно сформованих карт характеристик мережі [16]. Типова структура згорткової нейронної мережі зображена на рисунку 1.2.

Основою побудови згорткового шару є певний набір ваг. Внаслідок виконання цієї операції отримуємо нове зображення у вигляді карти об'єктів. Залежно від обраної матриці для згортки, карта об'єктів виділяє певні елементи з обробленого вхідного зображення. Щоб отримати найкращі характеристики

вхідного цифрового зображення, використовують кілька різних ядер згортки, що призводить до створення декількох різних карт ознак на виході згорткового шару. Ядро згортки фактично діє як фільтр, який переміщується над попередньою картою та виділяє певні характеристики об'єктів на зображенні.

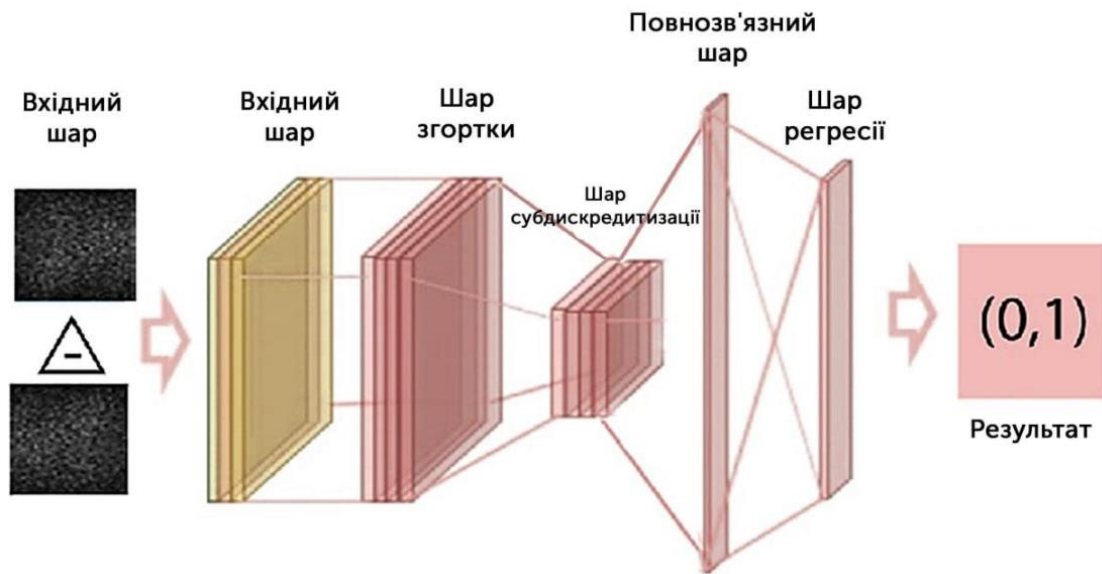


Рисунок 1.2 — Структура згорткової нейронної мережі

У згортковій мережі шар підвибірки виконує операцію нелінійного стиснення результуючої карти ознак, замінюючи групу пікселів (зазвичай 2x2) на один піксель. Найчастіше для реалізації цього процесу використовується операція знаходження максимуму. Ця операція виділяє неперекриваючі прямокутники або квадрати, замінюючи їх на піксель з максимальним значенням. Таким чином, підвибірка може значно зменшити розмірність аналізованого цифрового зображення. Це можна трактувати так: якщо важливі ознаки цифрового зображення були виявлені на попередньому етапі згортки, саме зображення стає менш важливим для наступної обробки, що дозволяє зменшити його деталізацію. Виконання операцій фільтрації на частинах вихідного зображення, які більше не є необхідними, також допомагає мережі уникнути навчання з нуля. У мережі шари згортки та підвибірки чергуються, причому шар підвибірки завжди йде після попереднього шару згортки, перед наступним шаром субдискретизації.

Після кількох таких переходів згортки цифрового зображення та наступного стиснення субдискретизації мережа переходить від високої роздільної здатності піксельної сітки вихідного зображення до абстрактних карт ознак. Кожен наступний шар збільшує кількість каналів і зменшує розміри цифрового зображення. Результатом є ряд мережових вихідних каналів, які містять зменшену кількість даних, інтерпретованих як абстрактні концепції, що виокремлюються з оригінального цифрового зображення.

Згенеровані таким чином дані об'єднуються і надсилаються в звичайну повністю пов'язану нейронну мережу, яка може містити кілька шарів. У цій мережі повністю взаємопов'язані шари більше не мають просторової структури пікселів і відзначаються невеликим розміром відносно кількості пікселів у вихідному цифровому зображенні.

У цьому розділі магістерської роботи був проведений аналітичний огляд особливостей функціонування систем комп'ютерного зору, виконаний аналіз методів та систем розпізнавання об'єктів та особливостей їх побудови, розглянуті підходи до формування ознак для виділення і розпізнавання об'єктів та запропоновано використати нейронні мережі глибокого навчання для розпізнавання заданих об'єктів у отриманій послідовності відео кадрів.

2 РОЗРОБКА ПОСЛІДОВНОСТІ ТА ЗАСОБІВ ВИДІЛЕННЯ І РОЗПІЗНАВАННЯ ОБ'ЄКТІВ

Пошук, виділення та розпізнавання об'єктів є однією із задач аналізу зображень для ідентифікації об'єктів та складається із ряду етапів. Ці етапи в залежності від вибраного способу опису об'єктів можуть виконуватися в різній послідовності. Даний розділ роботи присвячений розробці послідовності та засобів розпізнавання об'єктів у отриманому відео зображенні.

2.1 Формування послідовності розпізнавання об'єктів

Використання систем комп'ютерного зору для розпізнавання об'єктів є важливою частиною досліджень та практичних застосувань у сфері штучного інтелекту. Цей процес передбачає обробку та інтерпретацію візуальних даних, що дозволяє комп'ютерам сприймати зображення чи відео та відповідно реагувати на них. Розпізнавання об'єктів належить до ширшої області досліджень, відомої як розпізнавання образів.

Процес розпізнавання об'єктів починається з етапу їх виявлення на зображенні. Завданням цього етапу є визначення об'єкта у відеопотоці, що включає виявлення його ключових особливостей, визначення форми та позиції об'єкта у просторі для подальшої обробки комп'ютером [4, 11].

Перший крок ідентифікації об'єкта полягає у виявленні його на зображенні. Інструменти, що здійснюють це, працюють з цифровими зображеннями, що дозволяє організувати весь процес виявлення у кілька послідовних етапів [18].

Отримання цифрового зображення з відеокамери або іншого джерела. Аналіз та визначення наявності необхідного об'єкта на зображенні. Виділення ключових характеристик цього об'єкта. Розпізнавання об'єкта на основі обробленої інформації. Цей процес може бути виконаний за допомогою різних методик [2, 12].

Існує метод виділення та розпізнавання об'єктів на цифрових зображеннях, який полягає у поетапному співставленні нормалізованих зображень об'єктів, які мають бути розпізнані, та розміщенні їх у однакових за розміром осередках

таблиці об'єктів. Аналогічним чином здійснюється нормалізація та центрування шаблонних зображень, які також поміщаються в комірки тієї ж таблиці для подальшого порівняння [19]. Важливою умовою цього методу є попередня нормалізація та центрування об'єкта, що аналізується.

Існує також інший підхід до пошуку, виділення та розпізнавання об'єктів на цифрових зображеннях, який передбачає виділення замкнутого контуру, що максимально точно відповідає межах об'єкта. Після цього відбувається послідовне накладення шаблонів, що зберігаються в пам'яті комп'ютера, та їх поетапне порівняння [20]. Однак цей метод ефективний лише тоді, коли на зображенні присутній один об'єкт. Якщо на зображенні кілька об'єктів, або шуканий об'єкт відсутній, метод може дати неточні результати.

Існує спосіб, заснований на виділенні замкнутих контурів або контурів одного кольору, з подальшим представленням їх у вигляді коефіцієнтів, отриманих за допомогою перетворення Фур'є [21]. Ступінь подібності між двома об'єктами на різних зображеннях оцінюється за певним набором коефіцієнтів. Однак цей підхід надзвичайно чутливий до якості зображення: при роботі з зернистими або дрібними зображеннями метод може витратити багато часу на обробку, що ускладнює процес виділення кордонів об'єктів і навіть може призвести до виділення артефактів зображення як окремих об'єктів.

Ще один метод передбачає аналіз цифрових зображень для виділення областей, у яких можуть бути розміщені об'єкти [22]. Цей підхід виконує послідовні етапи: спочатку здійснюється пошук об'єктів, далі – їх розпізнавання, формування ідентифікаційного набору даних для кожного знайденого об'єкта. Ці дані можуть бути використані для створення груп схожих зображень, що містять об'єкти однакового типу.

Процес розпізнавання об'єктів відбувається через аналіз великої кількості навчальних даних, після чого формується вектор знайденого об'єкта, який може використовуватися для пошуку схожих об'єктів. Для вимірювання подібності пропонується використовувати різницю між векторами двох об'єктів.

Цей метод дозволяє ефективно виявляти та розпізнавати об'єкти на зображеннях, аналізуючи їхні цифрові представлення, виділяючи області з об'єктами, і порівнюючи їх із базою шаблонів. Підготовлені шаблони масштабуються та нормалізуються для порівняння з розпізнаваними об'єктами, що полегшує процес пошуку та класифікації.

Процес пошуку та виявлення об'єктів на цифрових зображеннях можна розбити на кілька ключових етапів:

- спочатку завантажується вихідне цифрове зображення;
- виконується його попередня обробка;
- далі на зображенні виділяються об'єкти, що підлягають пошуку;
- у виявлених об'єктах проводиться пошук характерних ознак за допомогою одного або кількох відомих методів;
- після цього формується набір ознак знайдених об'єктів;
- ідентифікація об'єкта здійснюється шляхом порівняння отриманого набору ознак з базою даних, що містить множину ознак базових об'єктів.

Два завершальних кроки стосуються етапу розпізнавання об'єкта, що можуть бути реалізовані по різному.

У нашій методології пошуку та виявлення об'єктів на цифрових зображеннях передбачається додаткова перевірка, яка включає уточнення позицій знайденого об'єкта на основі розміщення ключових точок, характерних для певного типу об'єкта. Деякі етапи пошуку потребують коригування, особливо стосовно обробки зображення на початкових стадіях розпізнавання. З урахуванням цих моментів, процес пошуку та виділення заданого об'єкта включає наступні кроки:

- завантаження цифрового зображення;
- фільтрація зображення для усунення можливих шумів;
- виконання вирівнювання яскравості зображення — це визначення мінімальних і максимальних значень яскравості та коригування зображення по всьому діапазону;
- виділення об'єктів на зображенні;

- пошук і визначення характерних ключових точок об'єкта;
- пошук об'єкта та прийняття рішення щодо його відповідності.

Уточнення результату шляхом порівняння вибраного шаблону з виділеним фрагментом зображення.

Виділення ознак для подальшої ідентифікації об'єкта.

Заключним етапом роботи є розпізнавання знайденого об'єкта.

Пропонується перед початком виконання процесу розпізнавання об'єктів використати фільтр для підкреслення контурів об'єктів зображення. Таким фільтром може бути видозмінений оператор Лапласа.

Роль оператора Лапласа в завданнях обробки зображення, у тому числі сегментації зображення, зводиться до використання його властивості виділення перетину нульового рівня функції для локалізації контуру об'єкта і з'ясування, на темній або ж на світлій стороні контуру знаходиться даний піксель зображення. Цей процес фактично є пошуком другої похідної функції, що описує яскравість зображення. Якщо змінити у сторону збільшення значення центрального елемента матриці Лапласа, то зміниться середнє значення яскравості зображення і одночасно границі будуть різкіше виділятися на отриманому фоні зображення. Варіанти такого видозміненого фільтру приведено у виразі (2.1).

При використанні такого оператора здійснюється множення кожного елемента вибраної двовимірної апертури розміром 3x3 пікселя на відповідний елемент матриці підкреслення контурів об'єктів зображення (2.1):

$$G_1 = \begin{bmatrix} +1 & +1 & +1 \\ +1 & 10 & +1 \\ +1 & +1 & +1 \end{bmatrix}, \quad G_2 = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}. \quad (2.1)$$

Одним із недоліків таких операторів є досить висока чутливість до завад. Можливі також появи розривів в контурах об'єктів, а також їх подвоєння.

Головною перевагою таких операторів є те, що вони нечутливі до орієнтації границь областей об'єктів і мають малу ресурсоемкість.

Запропонований підхід виділення об'єктів спрямований на підвищення точності розпізнавання у зображенні шуканих об'єктів за рахунок використання попередньої обробки, масштабування й нормалізації отриманого вхідного зображення, виконання операцій по виділенню ознак та прийняття рішення про знайдений об'єкт шляхом порівняння сформованих ознак об'єкта із базовими шаблонами із використанням для зрівняння виділених ключових точок об'єкта. Запропоновану послідовність обробки відео зображення використовуємо для пошуку та виділення об'єктів із використанням підходу по виділенню базових характерних ключових точок.

2.2 Розробка способу виділення об'єктів зображень

На теперішній час існує значна кількість різних методів виділення ознак і розпізнавання об'єктів, такі як методи вилучення ознак, методи на основі машинного навчання, використання глибоких нейронних мереж та інші методи розпізнавання об'єктів [23].

Для вирішення завдань розпізнавання об'єктів у різних областях широко використовуються традиційні методи вилучення ознак. Використовуються такі методи, як інваріантне до масштабу перетворення функцій (Scale-Invariant Feature Transform, SIFT) і прискорені надійні функції (Speeded-Up Robust Features, SURF) [10]. Ці методи націлені на виділення стійких і унікальних характеристик зображень, що забезпечує ефективне порівняння та розпізнавання об'єктів з різних ракурсів та в різних умовах. Існують алгоритми, які використовують ключові точки (feature points) для аналізу зображення. Під ключовими точками розуміються особливі ділянки зображення, які є унікальними для нього. Для знаходження та порівняння таких точок на цифрових зображеннях використовуються три основні компоненти: детектор, дескриптор і матчер. Детектор (feature detector) відповідає за пошук ключових точок на зображенні, дескриптор (descriptor extractor) описує знайдені точки, аналізуючи

навколишні області, а матчер (matcher) зіставляє точки між різними зображеннями.

Алгоритми, що шукають ключові точки, демонструють високу стійкість до перешкод і змін у вигляді трансформацій, що дозволяє виявляти об'єкти навіть за наявності різного роду завад. Деякі з цих алгоритмів працюють настільки швидко, що можуть бути використані для пошуку об'єктів у реальному часі.

Метод SIFT знаходить і описує ключові точки, які є інваріантними до змін масштабу, повороту та афінних перетворень. Він ідентифікує характерні точки на зображенні, обчислює локальні дескриптори навколо цих точок і створює вектор ознак, що представляє кожну точку. Ці дескриптори потім використовуються для зіставлення точок між різними зображеннями, що дозволяє ідентифікувати й локалізувати об'єкти.

Метод SURF пропонує альтернативний підхід до вилучення ознак, акцентуючи увагу на ефективності та надійності. Цей метод визначає ключові точки на основі масштабу й орієнтації зображення, а дескриптори формуються за допомогою відповідей Хаара у вейвлет-просторі. Дескриптори SURF розроблені так, щоб бути стійкими до змін масштабу, поворотів і часткових перекриттів, що робить цей метод ефективним для розпізнавання об'єктів в умовах зміни масштабу та ракурсів.

Методи SIFT і SURF продемонстрували надійність при розпізнаванні об'єктів з урахуванням варіацій у масштабі, обертанні та зміні точки огляду. Однак важливо зазначити, що ці методи виділення ознак можуть стикатися з труднощами у випадках, коли присутні значні оклюзії, складний фон або обмежена текстурна інформація на зображеннях. Крім того, їх висока обчислювальна складність може негативно впливати на продуктивність у реальному часі, особливо під час обробки великих обсягів даних або зображень з високою роздільною здатністю.

Їхня здатність залишатися стійкими до змін масштабу, поворотів і афінних трансформацій робить ці методи придатними для роботи з варіаціями зображень

об'єктів, але їхні обмеження при обробці оклюзій та обчислювальна ефективність звужують їх сферу використання.

Контурний аналіз є методом, який використовується для опису, пошуку, порівняння, розпізнавання та збереження графічних зображень об'єктів, представляючи їх у формі контуру [24]. Під контуром розуміють лінію будь-якої форми, що окреслює межі об'єкта на зображенні, яке аналізується. Цей підхід передбачає, що контур містить достатньо інформації для опису форми об'єкта. Важливо зазначити, що внутрішні точки об'єкта при цьому не враховуються. Основною перевагою методу є його стійкість до змін масштабу, обертання та зміщення контуру на зображенні, що робить контурний аналіз ефективним для пошуку об'єктів заданої форми.

Проте, ці припущення стосовно контуру об'єкта значно обмежують сферу застосування даного методу. Проблеми можуть виникати через складність виділення чіткої межі об'єкта: якщо яскравість об'єкта близька до фону, або об'єкт перекритий іншими елементами, це може ускладнити або навіть унеможливити коректне виділення контуру. При перекритті об'єктів або їх групуванні контур часто виділяється неправильно і не відповідає реальним межам об'єкта.

Таким чином, контурний аналіз має низьку стійкість до перешкод: при поганій видимості або порушенні цілісності контуру метод стає неефективним, призводячи до неправильних результатів або відсутності можливості виявлення об'єкта.

Метод зіставлення шаблонів (Template Matching) є хорошим варіантом, коли потрібно швидко перевірити наявність певного об'єкта на зображенні [21].

Серед методів машинного навчання для розпізнавання об'єктів часто використовується метод опорних векторів (SVM). SVM є моделями, що працюють на основі навчання з учителем і виконують класифікацію об'єктів, знаходячи оптимальну гіперплощину для розділення різних класів [14]. Цей метод добре підходить для обробки великих об'ємів даних та аналізу зображень

завдяки здатності працювати з високорозмірними просторами та обробляти нелінійні залежності.

Ще один підхід, це метод випадкових лісів (Random forest), що базується на машинному навчанні, і також показав чудові результати в задачах розпізнавання об'єктів. Випадкові ліси складаються з ансамблю рішень, де кожне дерево здійснює класифікацію об'єктів на основі обмеженого набору ознак, незалежно від інших дерев. Ця методика відзначається високою стійкістю до перенавчання та здатністю обробляти складні залежності між ознаками, що робить її ефективною для розпізнавання об'єктів.

Такі розширені алгоритми, як AdaBoost і Gradient Boosting, також активно використовуються у сфері розпізнавання об'єктів [11]. Вони функціонують через ітеративне навчання слабких класифікаторів, перетворюючи їх на потужніші моделі. Зосереджуючись на вибірках, які були неправильно класифіковані, і відповідно коригуючи вагові коефіцієнти, ці алгоритми покращують результати на кожній наступній ітерації. Завдяки своїй здатності добре працювати зі складними та різноманітними наборами даних, ці методи є ефективними для розпізнавання об'єктів, коли їх зовнішній вигляд та контекст значно відрізняються.

Методи машинного навчання пропонують значну перевагу за рахунок своєї адаптивності та гнучкості, що дозволяє створювати високоточні моделі для розпізнавання об'єктів. Однак, для досягнення максимальних результатів такі методи потребують великого обсягу анотованих даних для навчання та ретельного налаштування параметрів, а тому у нашому випадку їх не будемо використовувати.

Ще одним підходом до розпізнавання об'єктів є методи, що базуються на шаблонах. Вони використовують заздалегідь визначені або навчені шаблони для порівняння з аналізованим зображенням, щоб визначити, до якого класу об'єктів воно належить. Метод пошуку за шаблоном застосовується для виявлення ділянок зображення, які найбільше відповідають визначеному шаблону.

Основними вхідними даними є зображення, у якому здійснюється пошук, і зображення об'єкта, який потрібно знайти.

Однією з умов успішного застосування цього методу є те, що розмір шаблону має бути меншим за аналізоване зображення. Мета полягає у тому, щоб знайти ділянку на зображенні, яка найбільш точно збігається з шаблоном.

Пошук здійснюється шляхом поступового переміщення шаблону на один піксель по зображенню та оцінки подібності кожної нової ділянки до шаблону. За результатами цього процесу визначається область, яка має найвищий коефіцієнт подібності, що фактично представляє відсоток збігу між ділянкою зображення та шаблоном.

Глибинні нейронні мережі, зокрема згорткові нейронні мережі (CNN), стали революційним досягненням у сфері розпізнавання об'єктів в різних галузях. CNN трансформували комп'ютерний зір, оскільки здатні автоматично вивчати ієрархічні представлення прямо з необроблених вхідних даних, що дозволяє обійти потребу у ручному відборі ознак.

У контексті розпізнавання об'єктів нейронні мережі продемонстрували високу ефективність у їх виявленні та класифікації. Архітектура CNN зазвичай складається з кількох рівнів, включаючи згорткові, об'єднувальні та повністю підключені шари [25]. Згорткові шари відповідають за витягування локальних ознак із зображення, застосовуючи операції згортки, тоді як об'єднувальні шари зменшують просторові розміри, що сприяє формуванню ознак, інваріантних до трансляції. Наприкінці мережі повністю підключені шари об'єднують вивчені ознаки та роблять прогнози про класи об'єктів.

Однією з ключових переваг CNN є їх здатність захоплювати просторові відносини та ієрархічні структури, що дозволяє їм ефективно обробляти зміни в масштабі, обертанні та ракурсі. Навчаючись на великих наборах даних, CNN можуть автоматично виявляти дискримінаційні ознаки, важливі для розпізнавання об'єктів, що робить їх особливо корисними для виявлення об'єктів у складних сценах з насиченим фоном.

Традиційні методи, такі як шаблонне зіставлення та функціональні підходи, забезпечують простоту та ефективність, проте можуть стикатися з труднощами в складних сценах, а також при варіаціях освітлення і ракурсу. У той же час, методи на основі глибокого навчання, такі як Faster R-CNN [26] та YOLO [27], показують відмінні результати в плані точності і надійності, особливо при виявленні кількох об'єктів у режимі реального часу.

З огляду на використання методів розпізнавання об'єктів, важливість продуктивності в реальному часі є критичною через динамічний характер сцен. Тому глибокі методи навчання, як-от Faster R-CNN і YOLO, виявляються більш підходящими для цього. Вони використовують згорткові нейронні мережі для ефективної обробки великих обсягів даних і виявлення об'єктів у реальному часі, що дозволяє безпілотним літальним апаратам виконувати завдання з відстеження об'єктів. При виборі методу розпізнавання також слід враховувати доступні обчислювальні ресурси, оскільки алгоритми на основі глибокого навчання можуть вимагати значних обчислювальних потужностей.

Отже, методи на основі нейронних мереж, таких як YOLO і Faster R-CNN, пропонують прекрасну продуктивність і можливості для розпізнавання об'єктів у масштабі реального часу. Для досягнення оптимальних результатів по здійсненню розпізнавання об'єктів важливим є також узгодження вибору методу із конкретними вимогами, використовуваними обчислювальними ресурсами і обмеженнями при їх застосуванні.

2.3 Вибір нейронної мережі

У процесі розпізнавання об'єктів необхідно виділити кадр зображення, виконати його попередню обробку, здійснити пошук зони, де знаходяться об'єкти, виділити їх ознаки та виконати розпізнавання об'єктів. Завершальні етапи цієї послідовності можна виконати із використанням нейронної мережі. Як було показано у попередньому параграфі, це можна здійснити із застосуванням мережі Faster R-CNN або YOLO. Мережа Faster R-CNN, як показано у джерелах [28], має більшу точність при розпізнаванні об'єктів, хоча поступається у

швидкодії. Розглянемо особливості її роботи та можливості для виконання операції розпізнавання об'єктів.

Нейронна мережа — це модель статистичного навчання, заснована на принципах біологічних нейронних мереж. Кожен нейрон у мережі самостійно виконує прості функції: отримує вхідні дані, обробляє їх і передає на вихід. Але завдяки великій кількості взаємозв'язаних нейронів мережа може демонструвати досить складну, «інтелектуальну» поведінку. Зазвичай нейрони організовані в шари, по яких дані передаються від початкового (вхідного) до кінцевого (вихідного) шару, причому часто кілька разів.

Однією із кращих мереж для виділення заданих об'єктів є метод Faster R-CNN [26, 28]. Вибірковий пошук є трудомістким і повільним, що знижує ефективність роботи мережі. У методі Faster R-CNN застосовується підхід, який замінює вибірковий пошук алгоритмом виявлення об'єктів, що дозволяє мережі самостійно вивчати пропозиції регіонів. Зображення подається як вхідний сигнал у згорткову мережу для обчислення згортки. Для ідентифікації пропозицій регіонів замість вибіркового пошуку використовується окрема мережа, яка прогнозує ці регіони. Прогнозовані області пропозицій далі обробляються шаром об'єднання RoI, що дозволяє класифікувати об'єкти всередині цих областей і передбачати зміщення для обмежувальних рамок (рис. 2.1).

У Faster R-CNN згорткові шари, які відповідають за виявлення ознак і шаблонів, одночасно використовуються для RPN і шарів, що відповідають за визначення класу і місця розташування об'єктів. Це забезпечує ефективніше використання обчислювальних ресурсів і знижує час навчання та виконання. Завдяки RPN, який генерує області інтересу, і спільному використанню параметрів згорткових шарів Faster R-CNN може працювати значно швидше, ніж попередні архітектури, зберігаючи або навіть підвищуючи точність виявлення об'єктів. Faster R-CNN — це вдосконалена версія Fast R-CNN і включає кілька етапів розпізнавання.

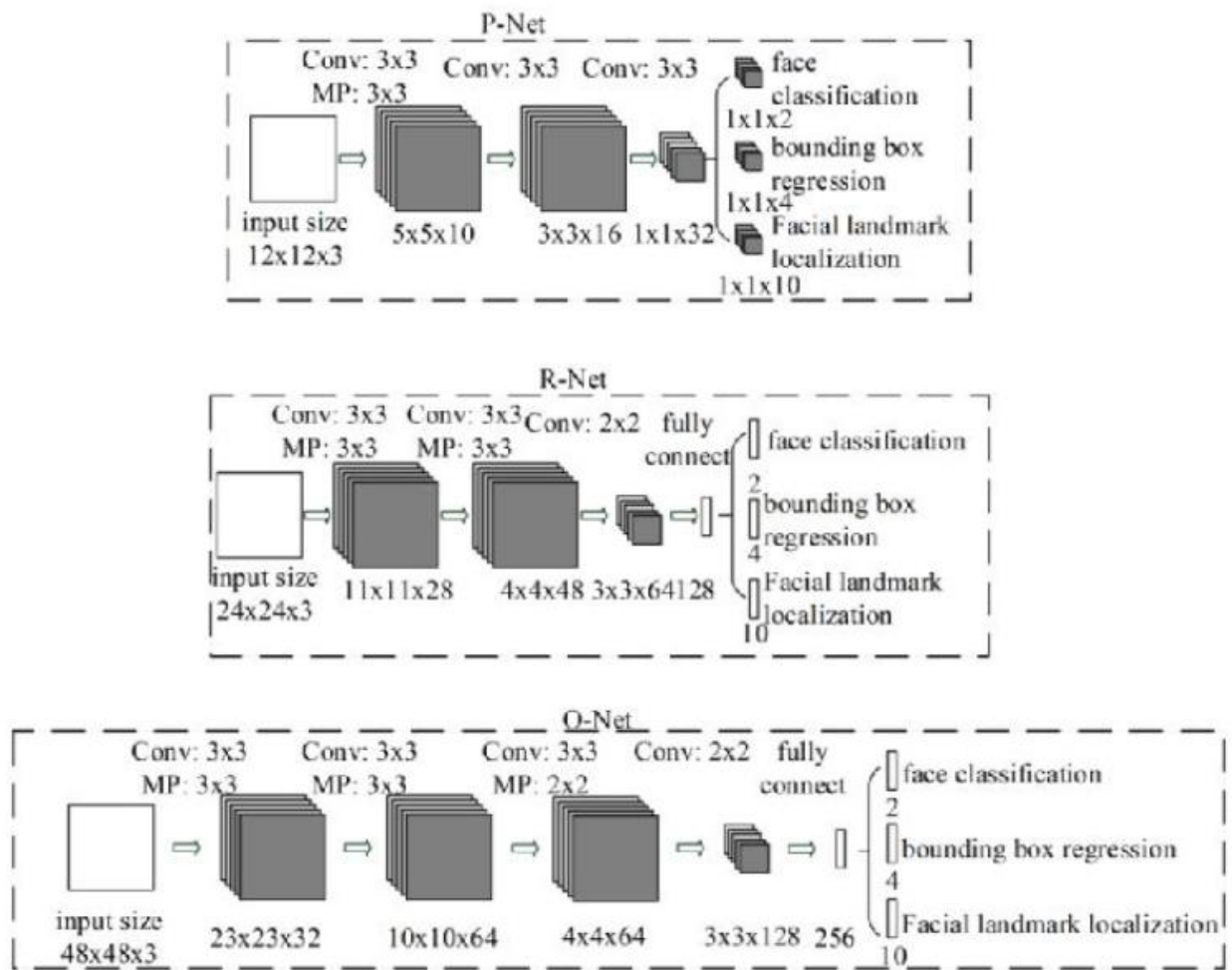


Рисунок 2.1 Схема нейронної мережі R-CNN

Процес починається з отримання вхідного зображення, яке може бути будь-якого розміру. Наступним етапом є виділення ознак. Так як і у методі Fast R-CNN, метод Faster R-CNN формує ознаки для усього зображення за допомогою CNN. Наступним є формування регіональної мережі пропозицій. У Faster R-CNN формується мережа пропозицій регіонів (RPN), яка використовує для цього ту ж саму базову нейронну мережу CNN, що й мережа виявлення. Метод RPN генерує пропозиції регіонів, накладаючи невелику мережу CNN на отримані карти об'єктів. Здійснюється генерація якірних блоків. Метод RPN генерує опорні рамки в різних масштабах й співвідношеннях сторін на картах об'єктів у зображенні. Ці опорні блоки виступають у ролі еталонних блоків для регіональних пропозицій. Виконується об'єднання методом RoI. Подібно до методу Fast R-CNN, регіони інтересу виділяються із карт об'єктів. Потім використовується об'єднання RoI для деформації виділених об'єктів у межах

кожного із RoI і переведення їх у карту об'єктів фіксованого розміру. Здійснюється виділення об'єктів за сформованими регіонами. Об'єкти, що відповідають кожному із RoI, виділяються із карт об'єктів. Сформовані та вирівняні за RoI об'єкти проходять обробку через повністю з'єднані шари, які надалі обробляють ці об'єкти й готують їх до класифікації і регресії граничних областей. Виконується класифікація та регресія на основі сформованих обмежувальних рамок.

Подібно до Fast R-CNN, Faster R-CNN використовує два вихідних шари для процедури немаксимального придушення. Після етапів класифікації та регресії обмежувальних рамок немаксимальне придушення (NMS) видаляє зайві рамки, залишаючи лише найбільш достовірні та точні варіанти. На фінальному етапі алгоритм повертає виявлені об'єкти, зберігаючи обмежувальні рамки з відповідними мітками класів і довірчими оцінками, що відображають знайдені об'єкти на вихідному зображенні. Загалом, Faster R-CNN є суттєвим кроком у напрямку створення швидших і ефективніших алгоритмів виявлення об'єктів, що забезпечують високу швидкість і точність, завдяки чому метод широко застосовується в багатьох галузях.

Розглянемо основні дії для налаштування згорткової нейронної мережі.

Одна з ключових функцій нейронної мережі — це здатність навчатися. Нейронна мережа є складною адаптивною системою, що може змінювати свою внутрішню структуру на основі вхідних даних. Це досягається шляхом регулювання вагових коефіцієнтів – чисел, які визначають силу сигналу між двома нейронами, що можуть бути як позитивними (стимулюючими), так і негативними (гальмуючими) [29].

Якщо мережа видає правильний результат, ваги не коригуються. Однак при неправильному результаті коефіцієнти налаштовуються для покращення наступних результатів. Важливим параметром є порогове значення, і сигнал передається лише тоді, коли сума вхідних сигналів перевищує цей поріг. Активованій нейрон обробляє сигнали, зважені на відповідні коефіцієнти, і передає їх далі, якщо результат перевищує поріг.

Взаємодія нейронів у мережі є адаптивною: зв'язки можуть утворюватися, змінюватися чи зникати. Це дозволяє мережі вчитися, тобто автоматично налаштовувати свої параметри для досягнення правильного результату. Інформація, що проходить крізь мережу, впливає на її структуру, яка адаптується на основі вхідних та вихідних даних.

Простіше кажучи, нейрони активуються через зважені сигнали від попередніх нейронів. Отриманий сигнал обробляється та може передаватися наступним нейронам. Навчання здійснюється шляхом подачі вхідних даних із відповідними вихідними. Під час цього контрольованого навчання мережа порівнює свій фактичний результат із очікуваним. Різниця між ними коригується за допомогою методу зворотного розповсюдження помилки, який працює від виходу до входу, налаштовуючи ваги та пороги, доки різниця між фактичним та бажаним результатом не стане мінімальною. Навчання включає багато (тисячі) ітерацій, що допомагають зменшити похибку.

Другий етап навчання – подача нових даних і перевірка результату для оцінки точності мережі. Вагові коефіцієнти – це параметри, які перетворюють вхідні дані в прихованих шарах. Усередині кожного вузла є вхідні дані, ваги і значення зсуву. Вхідні дані множаться на ваги і передаються далі, а основна частина ваг зосереджена в прихованих шарах.

Ваги застосовуються в вузлах прихованих шарів нейронної мережі. Наприклад, вузол може взяти вхідні дані, помножити їх на відповідні вагові значення, додати зсув і передати оброблену інформацію на наступний шар. Останній шар мережі, відомий як вихідний, адаптує дані, що надійшли з прихованих шарів, для отримання необхідного результату в певному діапазоні.

Ваги та зсуви є ключовими параметрами в мережі. Під час навчання нейронна мережа коригує значення ваг і зсувів, щоб досягти правильного виходу. Ваги можна сприймати як «міцність» зв'язку між нейронами: вони впливають на ступінь, до якого зміна вхідного значення відобразиться на виході. Малі вагові значення незначно впливатимуть на вихід, тоді як високі значення ваг матимуть значний вплив.

Функція активації, що отримує вхідний сигнал і створює вихідний сигнал на основі граничного значення, є важливим компонентом роботи мережі. Вона приєднана до кожного нейрона і визначає, чи буде нейрон активовано, залежно від значущості його вхідного сигналу для моделі (рис. 2.2). Функції активації також нормалізують вихід кожного нейрона, обмежуючи їх значення між 0 і 1 або між -1 і 1, що спрощує навчання.

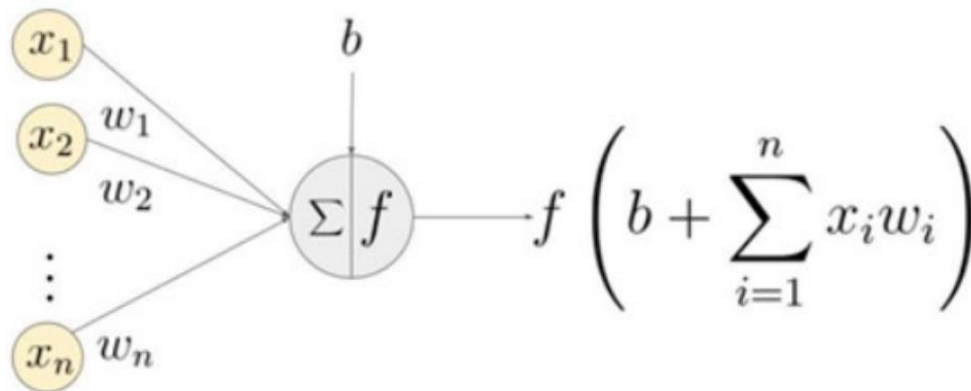


Рисунок 2.2 Приклад роботи функції активації

Ці функції повинні бути обчислювально ефективними, адже вони виконуються тисячами або навіть мільйонами разів для кожного зразка. Сучасні нейронні мережі використовують алгоритм навчання, зворотне розповсюдження помилки, який підвищує обчислювальну складність, адже залежить не лише від функцій активації, а й від їхніх похідних.

Функція активації – це перетворення вхідного сигналу поточного нейрона на його вихідний сигнал, що передається далі. Це може бути проста ступінчаста функція, яка активує або вимикає нейрон залежно від порогу, або функція, що перетворює вхідні дані у вихідні сигнали, потрібні для роботи нейронної мережі.

Існують такі розповсюджені типи функцій активації, як двійкові крокові функції, функція лінійної активації та нелінійні функції активації [16].

Функція двійкового кроку є пороговою функцією активації. Вона працює так: якщо вхідне значення перевищує певний поріг, нейрон активується, передаючи сигнал далі. Якщо вхідне значення менше порогу, нейрон залишається неактивним і не передає сигнал. Основна проблема функції

двійкового кроку полягає в тому, що вона не може передавати різні значення виходу, крім простого ввімкнення або вимкнення. Це обмежує її використання в задачах, де потрібно класифікувати дані на більше, ніж два класи, або представляти складніші, багатозначні зв'язки між вхідними даними та виходом.

Функція лінійної активації приймає вхідні сигнали і помножує їх на ваги для кожного нейрона і таким чином створює вихідний сигнал, що пропорційний вхідному. Функція лінійної активації має такі проблеми: для навчання моделі неможливо використовувати зворотне розповсюдження (градієнтний спуск) та усі шари нейронної мережі перетворюються в один.

Нейронна мережа з функцією лінійної активації діє як лінійна модель регресії, оскільки лінійна функція не може створювати складних відображень між вхідними та вихідними даними. Це обмежує можливість нейронної мережі працювати із завданнями, де потрібна складна обробка вхідних параметрів. Тому сучасні нейронні мережі використовують нелінійні функції активації, які розширюють їхню здатність моделювати складні та нерівномірні дані.

Нелінійні функції активації, такі як ReLU, сигмоїдна чи тангенсна функція, вирішують проблеми лінійної активації завдяки своїм похідним, які дозволяють застосовувати зворотне розповсюдження. Це дає змогу навчати мережу, ефективно налаштовуючи ваги та параметри для складніших моделей. Завдяки нелінійним активаціям стає можливим використання кількох шарів нейронів, утворюючи глибокі нейронні мережі, які здатні працювати з великомасштабними та багатовимірними даними та досягати високої точності.

Відомі такі нелінійні функції активації, як сигмоїдна (логістична), гіперболічна дотична, випрямлений лінійний блок ReLU, функція Leaky ReLU (рис 2.3) та функція Swish [27]. Кожна із них має свої переваги та недоліки. Кращою із них є функція Leaky ReLU, яку і вибираємо для налаштування мережі.

Для оцінки якості підготовки нейронної мережі до роботи використовуємо критерій оптимальності налаштування, в якості якого скористаємося квадратичним критерієм ξ [16].

$$\xi = \frac{1}{2} \sum_{s=1}^S \sum_{p=1}^M \left(y_p^s - \text{HM}_p^s(x^s) \right)^2 \quad (2.2)$$

де s – номер екземпляра, що перевіряється;

S – кількість екземплярів для перевірки;

p – порядковий номер вихідної ознаки;

M – кількість вихідних ознак;

y^s – набір значень вихідних ознак, що зіставлені s -му екземпляру;

HM_p^s – розраховане значення сигналу на p -му виходу нейронної мережі при надходженні на її входи x набору із значень вхідних ознак x^s ;

x^s – набір вхідних ознак для обробки.

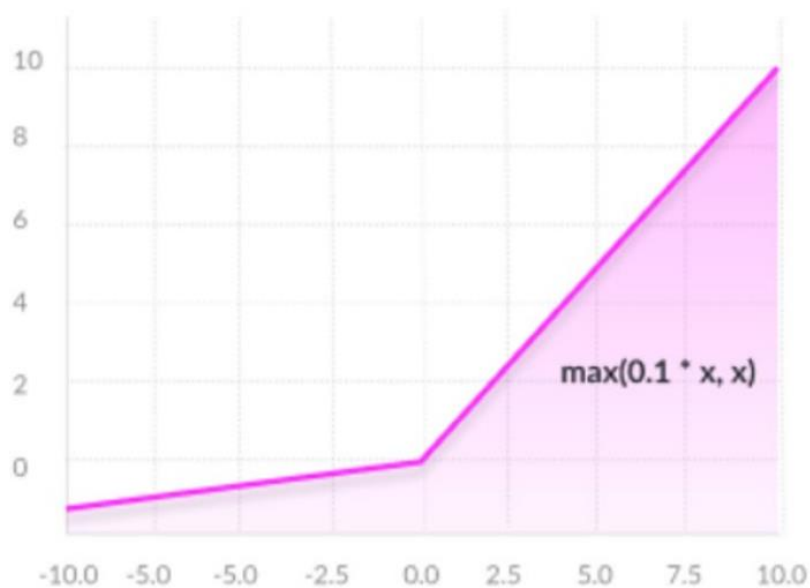


Рисунок 2.3 Функція активації Leaky ReLU

Для тренування нейронних мереж найбільш використовуваним алгоритмом є градієнтний спуск. Градієнт – це числовий показник, що вказує, як змінюються вихідні дані мережі відносно її параметрів. Завдяки градієнту можна коригувати ваги та параметри мережі так, щоб мінімізувати відхилення між передбаченим і фактичним результатами. Градієнтний спуск застосовує цю інформацію, поступово зменшуючи похибку на кожному кроці навчання.

Існує кілька варіантів алгоритму градієнтного спуску, що відрізняються за кількістю зразків, які вводяться в мережу для кожної ітерації: пакетний градієнтний спуск, стохастичний градієнтний спуск та міні-пакетний стохастичний градієнтний спуск [16].

Пакетний градієнтний спуск: усі доступні дані вводяться одночасно. Цей метод має великий ризик застрягання, оскільки градієнт обчислюється на основі всіх зразків, і варіації з часом зменшуються. Для нейронної мережі корисно вводити дані з певною випадковістю.

Стохастичний градієнтний спуск — на кожній ітерації вводиться один випадковий зразок. Градієнт обчислюється лише для цього конкретного зразка, що додає необхідну випадковість і зменшує ймовірність застрягання. Однак основною проблемою цього методу є його повільність, адже потрібно провести значно більше ітерацій, і він не використовує всі доступні обчислювальні ресурси.

Міні-пакетний градієнтний спуск — замість того, щоб вводити мережі поодинокі зразки, на кожній ітерації використовуються N випадкових елементів. Цей підхід зберігає переваги стохастичного методу, одночасно забезпечуючи швидшу обробку завдяки паралельним обчисленням.

Розглянемо ітерації алгоритму стохастичного міні-пакетного градієнтного спуску, який використовується для тренування нейронних мереж.

На вхід подається міні-пакет з N випадковими вибірками з раніше визначеного навчального набору даних. Після виконання обчислень на кожному шарі нейронної мережі отримуються вихідні прогнози. Цей етап називається поширенням вперед вхідного сигналу.

Далі проводиться оцінка функції втрат для міні-пакету. Ця функція є математичною моделлю, обраною з урахуванням специфіки проблеми, яка дозволяє визначити різницю між прогнозами мережі та фактичними мітками. Значення функції втрат є числом, яке алгоритм прагне мінімізувати, а наступні дії спрямовані на досягнення цієї мети.

При навчанні згорткової мережі, яка включає згортку із ядром K , що застосовується до багатоканального зображення V із кроками, які визначаються як $c(K, V, s)$. Тоді постає завдання мінімізації функції втрат $J(V, K)$. Під час прямого поширення помилки потрібно використовувати крок $c(K, V, s)$ для виведення значення Z , яке потім передається через решту шарів мережі і використовується для визначення функції витрат J . При виконанні процесу навчання мережі потрібно здійснювати обчислення похідних відносно ваг в ядрі. Для цього використовується функція

$$g(G, V, s)_{i,j,k,l} = \frac{\partial}{\partial K_{i,j,k,l}} J(V, K) = \sum_{m,n} G_{i,m,n} V_{j,(m-1) \times s + k, (n-1) \times s + l} \quad (2.3)$$

Гradient обчислюється як похідна функції втрат по всіх параметрах мережі. У графічному відношенні це нахил дотичної лінії до функції втрат у заданій точці. Математично gradient представляє собою вектор, який вказує напрямком, у якому функція втрат зростає швидше, отже, для її мінімізації необхідно рухатися в протилежному напрямку.

Виконати розрахунок gradientного вектору в нейронній мережі досить складно через значну кількість параметрів та їх розміщення у декількох шарах. Існує алгоритм, який дозволяє оцінити реальний вплив зміни параметра першого шару на кінцеву втрату, враховуючи, що ця зміна також вплине на всі нейрони в наступних шарах. Цей алгоритм називається зворотним поширенням. Він починається з обчислення часткових похідних функції втрат стосовно параметрів останнього шару, які не впливають на інші параметри мережі. Після отримання цих похідних алгоритм переходить до попереднього шару та обчислює часткові похідні функції втрат. Цей процес триває, поки не буде досягнуто початкового шару мережі.

Отримавши вектор gradientа, відбувається оновлення параметрів мережі: від їх поточних значень віднімається відповідне значення gradientа, помножене на швидкість навчання. Це дозволяє контролювати величину кроків. Важливо

правильно обрати рівень навчання, щоб уникнути небажаних результатів. Рівень навчання визначається із таких міркувань. Рівень навчання повинен бути обраний обережно: якщо він занадто високий, це може призвести до коливань, а якщо занадто низький — до повільного навчання і потреби в більшій кількості ітерацій. Його можна встановити як фіксованим, так і змінним. Наприклад, його можна зменшувати з часом, щоб уникнути коливань, коли модель наближається до мінімуму функції втрат. Також можливо використовувати циклічну швидкість, яка спочатку збільшується, а потім зменшується, що може допомогти уникнути місцевих мінімумів.

Рівень навчання може бути однаковим для всієї мережі або варіюватися між шарами (або групами шарів). Під час трансферного навчання рекомендується встановити низький рівень навчання для частини мережі, що ґрунтується на попередньо навченій моделі, та більш високий для нових шарів.

Епоха — це повний цикл, у якому мережа проходить через усі доступні зразки. Занадто велика кількість епох може призвести до обробки одних і тих же зразків занадто багато разів, що викликає надмірне навчання. Це може статися через невеликий набір даних у порівнянні з розмірами мережі, коли кількість навчальних зразків значно менша за кількість параметрів у мережі, а також якщо рівень навчання занадто малий, що заважає оптимізації параметрів.

Спуск градієнта — це метод оптимізації першого порядку, оскільки він використовує перші похідні функції втрат. Це дає інформацію про нахил функції, але не враховує її кривизну, що обмежує контекст. Хоча можливо обчислити другі похідні, це потребує значних обчислювальних витрат. Існують методи наближення, які оцінюють другі похідні з обмеженим використанням пам'яті, наприклад, LBFGS; проте найбільш поширеним є оптимізація стохастичного градієнтного спуску.

Деякі з найпопулярніших алгоритмів обчислення градієнту є Momentum, RMSProp (поширення середньоквадратичного розповсюдження) та Адам (оцінка адаптивного моменту) [16].

Операції нейронних мереж у процесі зворотного поширення помилки можна поділити на два основні етапи: пряме (подача вперед) та зворотне (подача назад). На етапі подачі вперед вхідні дані подаються на вхідний шар, і їх вплив поширюється через мережу, шар за шаром, поки не буде отримано вихід. Фактичне значення виходу мережі порівнюється з очікуваним, і для кожного з вихідних вузлів обчислюється сигнал помилки. Оскільки всі приховані вузли певною мірою вплинули на помилки, виявлені в вихідному шарі, сигнали помилки передаються назад з вихідного шару до кожного вузла в прихованому шарі, який безпосередньо вплинув на вихід. Цей процес повторюється по шарах, поки кожен вузол не отримає сигнал помилки, що описує його відносний внесок у загальну помилку.

Після того, як сигнал помилки для кожного вузла визначено, ці помилки використовуються для оновлення значень ваг зв'язків. Процес триває, поки мережа не досягне стану, що дозволяє їй ефективно кодувати всі аспекти навчання. Алгоритм зворотного поширення шукає мінімальне значення функції помилок у ваговому просторі, застосовуючи правило дельти або градієнтний спуск. Ваги, які мінімізують функцію помилок, вважаються оптимальними для задачі навчання.

У цьому розділі магістерської роботи вдосконалено метод пошуку заданих об'єктів у відео послідовності шляхом підкреслення їх контурів та подальшого розпізнавання із використанням нейронної мережі. Основним підходом до пошуку й виділення заданих об'єктів використано пошук області із заданими об'єктами, застосовано фільтри для підкреслення контурів об'єктів та використання згорткової нейронної мережі для розпізнавання заданих об'єктів у отриманому відео зображенні зони спостереження.

3 РОЗРОБКА ПРОГРАМНИХ ЗАСОБІВ ДЛЯ ВИЯВЛЕННЯ ЗАДАНИХ ОБ'ЄКТІВ У ЗОНІ ВІДЕОСПОСТЕРЕЖЕННЯ

Даний розділ магістерської роботи присвячений формуванню програмних і вибору апаратних засобів для пошуку і виділення заданих об'єктів у зоні відеоспостереження.

3.1 Вибір середовища та інструментальних засобів для створення програмного продукту

Об'єктно-орієнтовані мови є основним інструментом сучасного програмування, який широко використовується для реалізації програмних рішень. Наразі серед найпопулярніших мов є C++, C#, Java і Python.

Серед них, мова C++ вирізняється як високорівнева загальна мова програмування, що підтримує кілька парадигм, зокрема об'єктно-орієнтовану, процедурну та загальну. Бібліотеки в C++ базуються на стандартній бібліотеці шаблонів (STL), що дозволяє користувачам застосовувати потужні інструменти, як-от контейнери (наприклад, вектори або списки) та ітератори, які дають змогу отримати доступ до елементів контейнерів, що функціонують подібно до масивів. Бібліотека шаблонів також підтримує типи контейнерів, як-от стеки даних, асоціативні списки та черги.

Основним механізмом для організації даних у C++ є класи. На відміну від структур C, які складаються лише з полів, класи C++ включають поля та функції-члени або методи. Поля можуть бути з різним рівнем доступу: загальнодоступними, захищеними або приватними. Структури в C++ схожі на класи, однак у них стандартно поля і методи є відкритими, тоді як у класах вони за замовчуванням приватні. Загальнодоступні функції доступні всім користувачам, тоді як захищені й обмежені функції можуть використовувати лише учасники або друзі класу.

C# є об'єктно-орієнтованою мовою програмування, яка забезпечує безпечну систему типізації для платформи .NET, що розроблена для створення веб-додатків і базових програм [30]. C# за синтаксисом схожий із Java та C++ та

підтримує такі можливості, як поліморфізм, оператори функцій класу, XML-коментарі, властивості, винятки, події, атрибути, перевантаження операторів тощо. Це мова зі строгою статичною типізацією, яка запозичила функції з інших мов, як-от Smalltalk, C++, Modula та Delphi.

Java, на відміну від C++, є більш орієнтованою на об'єкти мовою. Всі дії та дані в Java організовані через класи об'єктів, за винятком базових типів (таких як int, float тощо). Через це Java не вважають цілком об'єктно-орієнтованою мовою. У Java всі об'єкти походять від базового класу, що називається "Object", і вони автоматично успадковують властивості й поведінку від нього. Хоча в C++ підтримується множинне успадкування, у Java реалізовано лише однократне успадкування, що запобігає конфліктам між членами класів, які успадковуються з батьківських класів.

Синтаксично Java частково схожа з C та C++. Заключна версія Java компілюється у переносимий байт-код, який потім виконується віртуальною машиною на відповідній платформі. Віртуальна машина Java та компілятор розроблені компанією Sun Microsystems. На відміну від C++, основну частину обчислень у Java виконує саме віртуальна машина. Java спершу була створена як кросплатформна мова, тому вона має обмежені можливості для низькорівневої взаємодії з апаратним забезпеченням. Ці ситуації вирішуються за допомогою викликів підпрограм, написаних іншими мовами.

Python — це мова програмування високого рівня загального призначення, що використовує динамічну типізацію та автоматичне керування пам'яттю [30]. Метою Python є підвищення продуктивності розробників, покращення читабельності коду та забезпечення високої якості програм, а також створення портативності написаного програмного забезпечення. Python є об'єктно-орієнтованою мовою, де все вважається об'єктом. Завдяки простому синтаксису значно знижується потреба у зверненні до документації. Python підтримує модулі та пакети, що полегшує модульність та повторне використання коду. Інтерпретатор Python і його стандартні бібліотеки доступні у вихідному коді на всіх основних платформах. Python підтримує кілька парадигм програмування,

зокрема об'єктно-орієнтовану, функціональну, процедурну та аспектно-орієнтовану.

Стандартна бібліотека Python містить безліч корисних функцій, починаючи від текстових обробок і закінчуючи мережевими інструментами. Python також підтримує додаткові функції, які можна реалізувати через широкий спектр сторонніх бібліотек, написаних на C або C++. Інтерпретатор Python легко інтегрується в проекти, створені на цих мовах. Крім того, існує спеціалізоване програмне забезпечення RuRI, розроблене для Python. Воно спрощує встановлення пакетів в операційній системі і є стандартом для Python.

Враховуючи ці переваги, Python є раціональним вибором для розробки програмного забезпечення.

Інтегроване середовище розробки (IDE) — це набір інструментів, призначених для розробки програм. IDE включає редактор коду та засоби для автоматизації компіляції та налагодження. Більшість сучасних IDE підтримують автозавершення коду, що прискорює процес написання. Основне завдання IDE — зменшити час на налаштування різних інструментів і надати розробнику єдиний набір функцій.

Деякі IDE створені для конкретних мов програмування, щоб запропонувати інструменти, які краще відповідають їхній парадигмі. Прикладами таких середовищ є Xcode, PhpStorm, Xojo та Delphi.

PyCharm — це інтегроване середовище розробки для Python, яке має набір додаткових інструментів для підвищення ефективності. PyCharm надає «розумний» редактор коду з автозавершенням та шаблонами коду. PyCharm Community Edition, безкоштовна версія, має відкритий код. Це середовище також перевіряє код в режимі реального часу, пропонує автозаповнення на основі аналізу коду під час виконання, а також містить інструменти для рефакторингу.

Серед основних особливостей PyCharm варто виділити такі [31]:

- значна кількість інструментів для перевірки згенерованого коду;
- розширена та зручна навігація по коду з підсвічуванням синтаксису;
- підтримка міжплатформного розгортання (Windows, Linux, macOS);

- можливість швидкого перегляду документації для кожного елемента коду безпосередньо у вікні редактора;
- допомога в написанні коду, включаючи загальні шаблони, автоматичне додавання та перевірку версії інтерпретатора;
- потужний рефакторинг, який дозволяє швидко й масштабно вносити зміни до проєкту;
- інтерактивні консолі для Python, SSH, Django, баз даних та механізмів автодоповнення.

На основі аналізу доступних середовищ розробки було прийнято рішення обрати PyCharm, яке оптимально підходить для роботи з Python.

3.2 Розробка структури програми виявлення об'єктів

У магістерській роботі розроблено програму для обробки цифрових зображень, яка виділяє задані об'єкти та виконує їх розпізнавання. Програмний продукт створено мовою Python із застосуванням бібліотеки OpenCV [32], що суттєво спрощує процес розробки. Це досягається завдяки відсутності необхідності написання численних допоміжних функцій, які часто потрібні для розв'язання окремих задач програмування.

Розроблений продукт для виявлення об'єктів працює в режимі реального часу: цифрове зображення сцени надходить послідовно кадр за кадром з камери, обробляється за заданим алгоритмом, а результати відображаються через інтерфейс користувача на моніторі.

Також використано широкі можливості OpenCV для Python. OpenCV — це потужна бібліотека комп'ютерного зору з відкритим кодом, яка забезпечує засоби для обробки зображень та містить численні алгоритми загального призначення. Реалізована на C/C++, бібліотека також доступна для Java, Python, Lua, Ruby та інших мов програмування. OpenCV можна вільно застосовувати в комерційних і навчальних проєктах на умовах ліцензії BSD [32].

Програма виділення об'єктів працює в реальному часі, де зображення з камер спостереження обробляються послідовно, кадр за кадром, у визначеній

послідовності операцій. Результати обробки миттєво відображаються на екрані користувача. Для роботи програми необхідно завантажити файли з бібліотек OpenCV та NumPy у версії для Python (зокрема, пакет opencv-python). Крім того, для коректного функціонування програми підключається модель згорткової нейронної мережі Faster R-CNN, а також базу даних, необхідну для початкового налаштування і навчання мережі.

Програмний засіб для виділення заданих об'єктів, що створений у даній магістерській роботі, складається із ряду програмних модулів. Створена структура програмного продукту є результатом об'єднання основних модулів програмної системи для виконання процесу відеоспостереження та реалізації запропонованого підходу по виділенню заданих об'єктів. Для виявлення заданих об'єктів використовується послідовний пошук та аналіз виявлених об'єктів із використанням згорткової нейронної мережі. Програмна частина сформованої комп'ютерної системи відеоспостереження складається із таких основних частин: модуля, що виконує введення даних із зони відеоспостереження, модуля, на який покладається попередня обробка отриманих даних, модуля, що виконує розпізнавання заданих об'єктів та модуля виведення результатів роботи системи відеоспостереження, а також модуля налаштування нейронної мережі [7].

Модуль введення даних з відеоспостереження відповідає за отримання відеокадрів з камери та забезпечує їх подальше використання для аналізу. Його основне завдання — отримати послідовність кадрів для виявлення об'єктів, що змінили своє місце розташування. Цей модуль забезпечує взаємодію між апаратними й програмними компонентами системи відеоспостереження.

Модуль попередньої обробки зображення є ключовим елементом системи, оскільки від якості обробки та аналізу вхідних даних залежить ефективність усієї роботи системи. В цьому модулі реалізовано кілька функцій, які покращують якість зображень, зокрема через фільтрацію та попереднє оцінювання, а також готують дані для наступних етапів обробки. Частину цих функцій було взято з відкритих бібліотек. Після обробки зображень цим модулем, система отримує основні параметри для пошуку й виявлення об'єктів у зоні спостереження.

Модуль розпізнавання заданих об'єктів здійснює ідентифікацію завчасно визначених типів об'єктів. Як один із варіантів таких об'єктів вибираємо процес розпізнавання автомобілів, а при достатніх умовах відео спостереження та сформованій базі даних зображень автомобілів може виявляти у отриманому кадрі зображення типи автомобілів та виконати операцію їх розпізнавання. Робота модуля розпізнавання заданих об'єктів ґрунтується на згортковій нейронній мережі із використанням глибокого навчання типу Faster R-CNN [24]. Нейронною мережею на основі виділення зображення автомобілів формуються ознаки для її наступного розпізнавання. Це виконується на основі попереднього навчання цієї згорткової нейронної мережі по здійсненню дій по пошуку, виявленню і розпізнаванню автомобілів серед інших сторонніх об'єктів.

Модуль виведення даних здійснює процес виведення отриманої інформації про задані об'єкти. Якщо додатково модуль розпізнавання буде налаштований на пошук й розпізнавання типу автомобілів, то даний модуль буде надавати інформацію ще і про тип автомобілів, дані про яких розміщені у заделегідь сформованій базі даних системи відеоспостереження. Запропонований такий склад модулів системи відеоспостереження дозволяє виділяти і розпізнавати задані об'єкти.

3.3 Розробка програми виділення об'єктів зображення

У даній магістерській роботі розроблено програмний засіб для пошуку й виділення заданих об'єктів, якими у зоні спостереження є автомобілі. Сформована програма виконує процес пошуку зони із заданими об'єктами, знаходження у виділеній сцені автомобілів із використанням згорткової нейронної мережі та подальшого розпізнавання типу автомобіля.

Для налаштування програми спочатку потрібно імпортувати необхідні функції з бібліотеки OpenCV, а також завантажити дані щодо згорткової нейронної мережі Faster R-CNN. Це включає опис конфігурації мережі та ініціалізацію параметрів для її початкового налаштування.

Лістинг 3.1 — Фрагмент коду завантаження бібліотек

```
import numpy as np
import cv2
import imutils
from Faster_RCNN.config import Config
from Faster_RCNN.utils import some_function
```

Першим кроком роботи програми є отримання кадрів із відеопотоку. Для цього підключається відеокамера з використанням функції `cv2.VideoCapture("vtest.avi")`. Після цього здійснюється розподіл на сусідні кадри відео, які отримують назви `frame1` та `frame2`. Це необхідно для виявлення змін у розташуванні об'єктів між двома послідовними кадрами. Різницю між кадрами обчислюємо для виявлення таких змін.

Оскільки кадри потрібно виділяти періодично, формується цикл для обробки відеопотоку. Спочатку необхідно конвертувати зображення з моделі кольорів BGR, що використовується в OpenCV, до моделі RGB. Подальше оброблення буде зручніше виконувати в градаціях сірого, тому кожен з сусідніх кадрів конвертуємо в відтінки сірого кольору за допомогою методу `cv2.cvtColor()`, де вказується початкова модель кольорів (BGR).

Для зменшення впливу шумів і покращення якості зображення застосовуємо розмиття. Для цього використовуємо функцію Гауса, яка є в OpenCV і реалізується через `cv2.GaussianBlur`. Ця функція приймає кілька параметрів: розміри вхідного зображення та зображення після обробки, стандартні відхилення по осях x і y , а також тип обробки границь і розмір ядра функції Гауса.

Для виділення об'єктів використовуємо нейронну мережу Faster R-CNN. Для цього попередньо конвертуємо модель нейронної мережі із застосуванням бібліотеки навчання `keras` і потім трансформуємо модель нейронної мережі Faster R-CNN у режимі виводу.

Лістинг 3.2 — Фрагмент коду формування моделі мережі

```
python convert.py = Faster R-CNN.cfg Faster R-CNN.weights model_data/
Faster R-CNN.h5
model = FASTER R-CNN (mode="inference",
model_dir=MODEL_DIR,
config= Faster R-CNN Config())
```

Завантажуємо попередньо навчену модель і передаємо зображення в Faster R-CNN для отримання результатів. Процес для визначення координат виділених об'єктів та розпізнавання людей здійснюємо за допомогою знайденого раніше фрагмента зображення, використовуючи функцію `param_image_to_process`. По завершенню розпізнавання об'єктів відображаємо зображення з виділеними прямокутниками.

Лістинг 3.3 — Фрагмент коду виділення фрагменту зображення

```
height, width, depth = image_to_process.shape
blob = cv2.dnn.blobFromImage(image_to_process, 1 / 255, (608, 608), (0, 0,
0), swapRB=True, crop=False)
net.setInput(blob)
outs = net.forward(out_layers)
class_indexes, class_scores, boxes = ([] for i in range(3))
objects_count = 0
```

Запускаємо процес пошуку об'єктів на отриманому зображенні, використовуючи раніше виділені фрагменти зображення та вказуючи координати області навколо кожного об'єкта. Для об'єктів, що належать до визначеного класу (наприклад, силуети знайдених автомобілів), малюємо прямокутники навколо них.

Лістинг 3.4 — Фрагмент коду виділення класу об'єктів

```
if classes[class_index] in classes_to_look_for:
    objects_count += 1
```

```

        image_to_process = draw_object_bounding_box(image_to_process,
class_index, box)

    final_image = draw_object_count(image_to_process, objects_count)
    return final_image

```

Для виведення зображення виділеного автомобіля, створюємо рамку у потрібному вікні зображення. Додаємо функцію, яка дозволяє вказати координати меж знайдених об'єктів (автомобілів) на зображенні, використовуючи метод `apply_Faster_R-CNN_object_detection`. Для цього застосовуємо параметр `index`, який виводить клас об'єкта, виділений за допомогою `Faster R-CNN`. Функція `return` повертає результат у вигляді зображення з позначеними об'єктами сцени. Для встановлення рамок на зображеннях знайдених автомобілів фільтруємо результати та виводимо оброблене зображення відеокадру. Рамки для виділених об'єктів з'являються на кадрі, і підсумкове зображення виводиться на екран.

Лістинг 3.5 — Фрагмент коду формування рамки для об'єктів

```

cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 255, 0), 1)
cv2.imshow('Video', frame)

```

Для правильної роботи програми необхідно встановити середовище `.NET` на комп'ютері, підключити відеокамеру для спостереження з відповідними драйверами та налаштуваннями пристрою керування процесами на комп'ютері.

В результаті отримуємо області зображення, що містять виділені силуети автомобілів, які використовуються для подальшого прийняття рішень. Розроблений програмний код для виконання вищезазначених підходів до виділення об'єктів, зокрема автомобілів, що з'являються у зоні спостереження, наведений у додатку Б магістерської роботи.

3.4 Вибір апаратних засобів системи відеоспостереження

Основною метою комп'ютерної системи відеоспостереження є моніторинг стану певної території. Це завдання реалізується завдяки використанню

технічних засобів системи відеоспостереження, серед яких важливу роль відіграють відеокамери, програмне забезпечення для розпізнавання об'єктів і осіб, а також виконання відповідних дій залежно від ситуації. Для ефективного контролю над зоною спостереження необхідно правильно вибрати відповідне обладнання для відео реєстрації, що включає, перш за все, відеокамери. Ці камери повинні бути здатні працювати в різних умовах освітлення, а також витримувати зміни температури і вплив інших зовнішніх факторів.

Особливо важливим є вибір камер, здатних виявляти та розпізнавати об'єкти за низького рівня освітлення, що є критичним, особливо вночі. Для цього оптимально використовувати відеокамери з інфрачервоним підсвічуванням. Багато компаній виробляють такі камери, які спеціалізуються на забезпеченні якісного зображення при обмеженому освітленні такі, як EZVIZ, Hikvision, DahuaTechnology, GreenVision [33,34] та іншими, головною особливістю яких є робота у великому діапазоні зміни рівня освітленості. Іншими параметрами цих відео камер є кут огляду та роздільна здатність, тип використаного об'єктиву, кількість передач кадрів відео у секунду та протокол обміну даними. Це такі відео камери, як DahuaDH-IPC-26P, Foscam FI9903P, HikvisionDS-2CD2T43WD-I8 [33] та ряд інших. Камери типу Foscam FI9903P та DahuaDH-IPC-G26P обладнані інфрачервоним підсвічуванням на відстань до 20-30м, зв'язок із засобами обробки даних здійснюється по мережі Wi-Fi, тому за цими показниками вони нам не підходять. Ліпшими показниками для виконання задач відеоспостереження є камера типу HikvisionDS-2CD2T42WD-I8 [33], які мають необхідні засоби відео спостереження в інфрачервоному діапазоні випромінювання частот та можуть контролювати зону до 30 метрів.

Зв'язок із центральним сервером системи буде здійснюватися через медіаконвертер TP-LINK MC110CS [35]. Для підключення відеокамер до медіаконвертера використовуватиметься комутатор UTP3-SW08-TP120.

Для зберігання даних про стан зони спостереження використовуватиметься відеореєстратор. Ми обрали відеореєстратори відомої компанії Hikvision. З кількох моделей, таких як Hikvision DS-7732NI-I4/24P,

Hikvision DS-7608NI-E2, Hikvision DS-7732NI-K4 та Hikvision DS-7208HWI-SH, зупинили свій вибір на моделі DS-7732NI-K4 [27]. Це професійний цифровий відеореєстратор на 32 канали запису даних із відеокамер, що підтримує роздільну здатність до 8 мегапікселів. Модель має підключення до чотирьох інтерфейсів SATA і забезпечує вхідну пропускну здатність до 160 Мбіт. Відеореєстратор підтримує підключення сторонніх мережевих IP-камер. Для стиснення даних використовується стандарт компресії H.265/H.264/MPEG4.

Для зберігання даних про зону спостереження вибираємо сервер типу Dell PowerEdge R420 [36]. Цей тип серверу характеризується такими параметрами: він має два процесори типу Intel® Xeon® із серії E5-2470v2 (4 ядра), обладнаний 4 відсіками під диски типу SAS із 15000 оберт/хвилину та із форм-фактором 3"5 обсягом на 6Г та загальним обсягом для зберігання даних 12 ТБ для роботи із операційною системою Microsoft Windows Server 2012.

Підготовлена схема підключення засобів системи відеоспостереження наведена на демонстраційних плакатах до магістерської роботи.

3.5 Налаштування нейронної мережі для розпізнавання об'єктів

Процес навчання нейронних мереж для розпізнавання об'єктів можна поділити на основні етапи. Розглянемо їх більш детально.

Для налаштування та тренування нейронної мережі формуємо комп'ютерну систему на центральному процесорі Intel Core-i3 7th Gen (2,30 ГГц) з 8 ГБ ОП та на одній інтегрованій відеокарті Intel HD Graphics 620.

Першим етапом є збір набору даних, що складається зі зображень або відео, на яких зображені об'єкти, які необхідно розпізнати. Важливо, щоб цей набір був репрезентативним і включав достатню кількість прикладів для кожного класу об'єктів.

Потім для кожного зображення чи відеокадру проводиться анотація — процес маркування об'єктів і визначення їх розташування на зображенні. Зазвичай це робиться за допомогою прямокутників обрамлення (bounding boxes)

або масок сегментації. Ці анотації служать «еталоном» (ground truth) під час навчання моделі.

Після цього зібрані та анотовані дані ділять на три набори: навчальний, валідаційний і тестовий. Навчальний набір використовується для тренування моделі, валідаційний — для налаштування гіперпараметрів і контролю перенавчання, а тестовий — для оцінки загальної продуктивності моделі.

Перед навчанням модель може потребувати попередньої обробки даних, що допоможе поліпшити результати та прискорити процес. Попередня обробка може включати зміну розміру зображень, аугментацію (повороти, зсуви, зміну контрастності тощо) і нормалізацію пікселів.

Вибір архітектури нейронної мережі залежить від специфіки завдання і наявних даних. У випадку задач розпізнавання об'єктів найчастіше застосовують згорткові нейронні мережі (CNN). У нашому випадку обрана модель Faster R-CNN завдяки її ефективності в обробці зображень і здатності точно визначати об'єкти.

Перед самим початком тренування модель ініціалізується або випадковими вагами, або вагами попередньо навченої моделі. Використання попередньо навченої моделі може суттєво пришвидшити процес навчання та підвищити його ефективність, особливо якщо набір даних невеликий.

Під час навчання модель працює з навчальним набором, використовуючи алгоритм оптимізації (у нашому випадку — стохастичний градієнтний спуск, SGD) для мінімізації функції втрат. У процесі роботи вона прогнозує мітки об'єктів на зображеннях і порівнює ці прогнози з наявними анотаціями. Ваги мережі коригуються так, щоб зменшити різницю між передбаченнями і реальними даними.

На кожній ітерації тренування модель перевіряється на валідаційному наборі, щоб оцінити поточну похибку та виявити можливе перенавчання. Для цього використовується стратегія ранньої зупинки: навчання припиняється автоматично, якщо значення IoU (Intersection over Union) на валідаційному наборі залишається незмінним протягом 10 ітерацій.

Важливим аспектом тренування є налаштування гіперпараметрів, таких як швидкість навчання, розмір батчу і кількість епох. Ці параметри регулюють на основі валідаційних даних для досягнення оптимальної продуктивності та запобігання перенавчанню. У нашому випадку швидкість навчання встановлена на рівні 0,0001, а модель тренується протягом 100 епох.

Також використовується стратегія ReduceLROnPlateau, яка автоматично знижує швидкість навчання, якщо продуктивність моделі на валідаційних даних не покращується протягом певної кількості епох. Це дозволяє точніше налаштовувати модель і підвищує її ефективність на різних етапах навчання.

Після того, як завершиться тренування моделі, вона проходить оцінку на тестовому наборі даних, щоб визначити її загальну ефективність у розпізнаванні об'єктів. Для цього використовуються ключові метрики, такі як точність (precision), повнота (recall), F1-міра та Intersection over Union (IoU). Оцінка на тестовому наборі дозволяє зрозуміти, наскільки добре модель працює з новими даними, а також допомагає виявити можливі проблеми, такі як перенавчання або недонавчання.

Після тестування важливо проаналізувати отримані результати, щоб знайти слабкі місця моделі та визначити шляхи її вдосконалення. Це може включати зміну архітектури нейронної мережі, додавання регуляризації, розширення набору даних або застосування інших технік для підвищення продуктивності. У нашому випадку тестовий набір даних використовується для остаточної оцінки моделі та перевірки її готовності до реальних завдань.

Заключним етапом є інтеграція навченої моделі в реальну систему або застосунок для розпізнавання об'єктів у відеопотоці. Цей процес може вимагати оптимізації моделі для прискорення обробки даних, адаптації для роботи на різних платформах (мобільних пристроях чи вбудованих системах) та інтеграції з іншими компонентами — такими як бази даних, інтерфейси користувача або системи сповіщення.

Всі ці кроки можуть повторюватися і вдосконалюватися на різних етапах розробки, щоб досягти найкращих результатів і відповідати специфічним

вимогам конкретного застосування або викликам, які виникають у процесі роботи.

У даному розділі магістерської роботи були обрані мова та середовище програмування, сформована архітектура програми та розроблено програмний засіб для реалізації запропонованого підходу для оброблення зображення для виявлення заданих об'єктів у зоні відеоспостереження та здійснено вибір апаратних засобів для реалізації комп'ютерної системи відеоспостереження.

4 ПЕРЕВІРКА ПРАЦЕЗДАТНОСТІ ПРОГРАМНОГО ПРОДУКТУ

У даному розділі розглянуто процес навчання нейронної мережі, особливості формування бази даних для перевірки якості роботи програми та аналіз якості роботи створеного програмного продукту.

4.1 Навчання нейронної мережі

Для навчання моделі Faster R-CNN ми будемо використовувати датасет у форматі COCO [37]. Цей формат організовано у вигляді п'яти основних розділів, які містять повну інформацію про набір даних. COCO підтримує кілька типів анотацій, таких як: визначення об'єктів, ключові точки, сегментація об'єктів, паноптична сегментація, щільна поза та текстові підписи до зображень. Усі анотації зберігаються у файлах формату JSON.

Для розпізнавання об'єктів використовуються такі поля:

- info – загальна інформація про набір даних;
- licenses – відомості про ліцензії на зображення в наборі даних;
- images – перелік зображень у наборі;
- annotations – список анотацій, включно з рамками, які є у зображеннях;
- categories – перелік категорій міток.

Щоб зручно працювати з форматом COCO, можна використовувати клас CocoDetection із бібліотеки PyTorch, який значно спрощує обробку та підготовку даних.

Наступним етапом підготовки даних для моделі є розділення датасету на навчальну та валідаційну вибірки. Це необхідний процес, який дозволяє оцінити якість моделі на даних, що не використовувалися під час її тренування.

Спершу визначаємо розмір навчальної вибірки, який зазвичай складає 70% від загального обсягу датасету. Такий підхід гарантує, що модель отримає достатню кількість даних для навчання, залишаючи при цьому решту 30% для перевірки її ефективності на етапі валідації.

У нейронних мережах, особливо в задачах комп'ютерного зору, таких як Faster R-CNN, важливу роль відіграє backbone. Це компонент, що відповідає за

витягнення ознак із вхідного зображення. Завдяки цьому модель отримує структуровану інформацію, яка служить основою для подальшої обробки виявлення об'єктів і їх класифікації.

Зазвичай у ролі backbone використовується попередньо навчена модель, наприклад, VGG або ResNet. Такі мережі добре справляються з визначенням як простих, так і складних ознак, що значно підвищує ефективність навчання.

Використання попередньо навченого backbone має кілька важливих переваг. До них відносяться покращення точності моделі, ефективність навчання та покращення продуктивності в реальному часі.

Попередньо навчені мережі, такі як ResNet або VGG, здатні розпізнавати широкий спектр ознак, оскільки вже були навчені на великих датасетах (наприклад, ImageNet). Це приводить до покращення точності моделі і дозволяє новій моделі краще адаптуватися до складних задач аналізу нових зображень.

Використання готового backbone суттєво зменшує час і ресурси, потрібні для навчання моделі, що покращує ефективність навчання. Замість того щоб вчити мережу з нуля, основні обчислення спрямовуються на налаштування для вирішення конкретної задачі.

У задачах, де важливий швидкий відгук, наприклад, розпізнавання об'єктів для автономних транспортних засобів, використання попередньо навченого backbone дозволяє моделі швидше обробляти інформацію та забезпечувати високу точність навіть у ресурсно обмежених середовищах, що призводить до покращення продуктивності в реальному часі.

У цьому проєкті використовується ResNet50 як попередньо навчена модель. У порівнянні з більш складними мережами, такими як ResNet101, ResNet50 має менше параметрів і спрощену архітектуру, що знижує вимоги до обчислювальних ресурсів під час тренування та виконання.

Для демонстрації роботи модель інтегрована у систему автоматичного розпізнавання об'єктів на відео. Вона здатна виконувати ідентифікацію та класифікацію об'єктів у реальному часі. На GPU забезпечується швидка обробка кадрів, хоча на малопотужних відеокартах частота обробки складає близько 6–7

кадрів на секунду. Модель ефективно розпізнає об'єкти, вказуючи їхні класи та точність визначення. Приклад такого розпізнавання приведено на рисунку 4.1

Інтеграція даних із додаткових сенсорів, таких як лідара, радара або інших пристроїв, може суттєво підвищити ефективність розпізнавання об'єктів. Поєднання різномірної інформації надає моделі можливість отримувати більш повну і детальну картину про навколишнє середовище, що сприяє покращенню точності як виявлення, так і класифікації об'єктів.

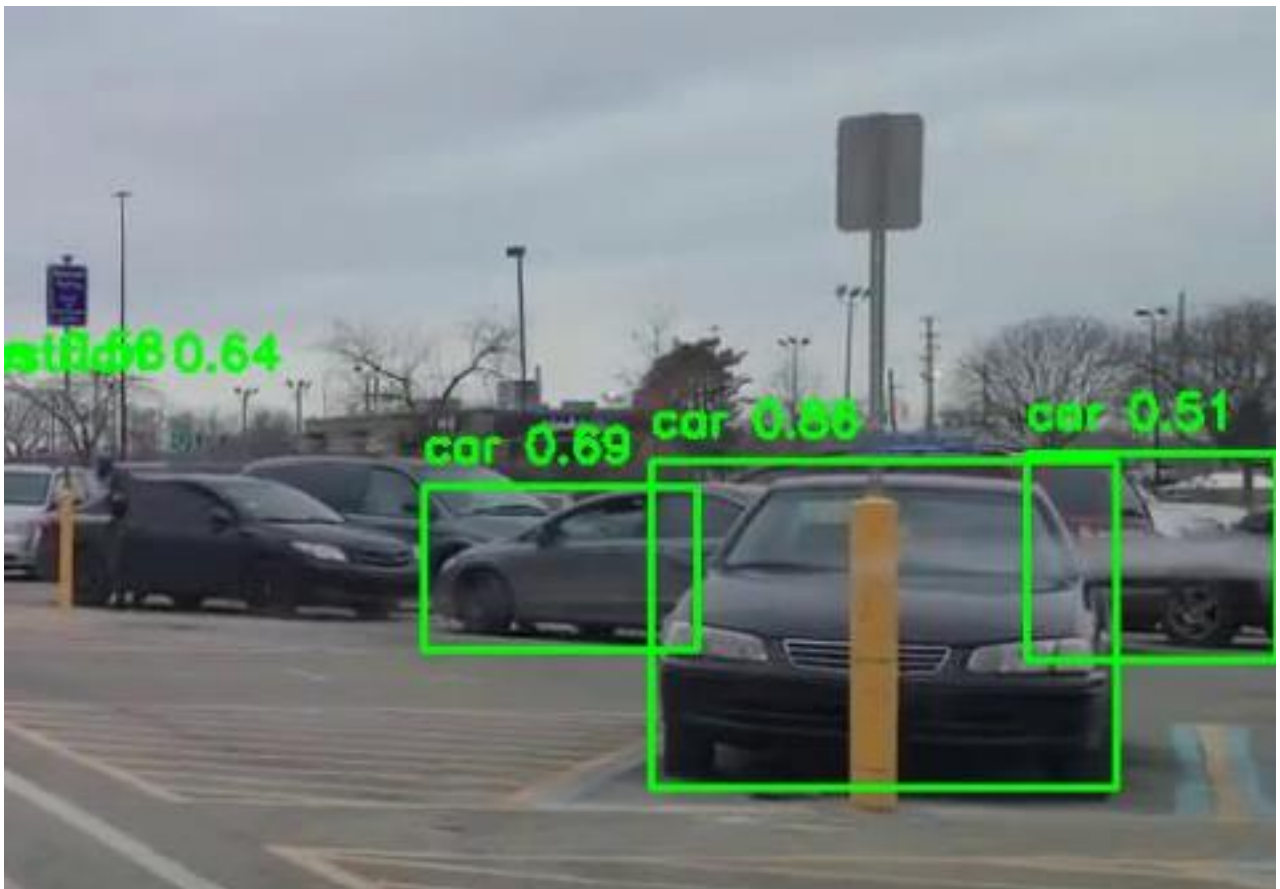


Рисунок 4.1 — Приклад розпізнавання автомобілів на стоянці

4.2 Використання датасету для навчання нейронної мережі

Датасет Udacity Self Driving Car Dataset був створений з метою вдосконалення методів навчання, що використовуються в розпізнаванні автономних транспортних засобів. Цей набір даних містить значну кількість анотованих зображень, отриманих за допомогою камер автономних автомобілів, і призначений для навчання алгоритмів комп'ютерного зору.

Усього датасет включає 30 000 зображень, знятих за різноманітних дорожніх і погодних умов. Обрана версія набору даних складається із зображень з роздільною здатністю 512x512 пікселів, що дозволяє досягти балансу між детальним аналізом і зменшенням складності навчання моделей.

Кожне зображення супроводжується анотаціями, які містять інформацію про такі об'єкти, як автомобілі, пішоходи, дорожні знаки та світлофори. Анотації точно описують місця розташування об'єктів на зображеннях, що дозволяє ефективно використовувати ці дані для тренування моделей глибокого навчання.

Зображення відображають широкий спектр дорожніх сценаріїв, включно з міськими дорогами, автомагістралями, а також різними часами доби та погодними умовами. Таким чином, даний датасет є ідеальним інструментом для навчання та проведення порівняльного аналізу моделей розпізнавання об'єктів.

Датасет складається із 97942 міток для 11 класів. Також у складі датасет є 1720 зображення без жодної мітки. Баланс класів має такі зображення: 64399 міток автомобілів, 10806 людей, 3623 грузовики, 1864 мотоциклістів та інші дані, що пов'язані із переміщенням автомобілів по дорозі [26].

Датасет містить значну кількість міток і широкий спектр класів, що забезпечує хорошу основу для навчання моделей глибокого навчання, таких як Faster R-CNN. Однак є й недоліки, які можуть вплинути на ефективність моделі. Одним із головних є нерівномірний баланс класів, через що модель може схилитися до кращого розпізнавання об'єктів із більш поширених класів, ігноруючи рідкісні. Також зображення без міток можуть створювати додаткові труднощі, якщо їх попередньо не опрацювати.

Серед потенційних проблем слід виділити якість анотацій і самих зображень, оскільки це прямо впливає на точність результатів. Налаштування параметрів моделі також потребує численних експериментів, особливо коли окремі класи представлені в недостатній кількості.

Для підвищення ефективності навчання варто застосовувати техніки аугментації даних, щоб збільшити різноманітність прикладів, і приділити увагу рідкісним класам, використовуючи вагові коефіцієнти. Регулярна перевірка на

валідаційній вибірці дозволить уникнути перенавчання та вдосконалити модель відповідно до поставлених цілей.

У контексті задачі розпізнавання об'єктів із використанням згорткових нейронних мереж виникають специфічні труднощі, пов'язані з характеристиками датасету, такими як обсяг і якість тренувальних даних, особливості об'єктів на зображеннях і можливість включення синтетичних даних для підвищення продуктивності.

Однією з ключових проблем, яка може виникнути при навчанні моделі, є обмежений обсяг тренувальних даних. Нестача зображень для навчання може стати причиною перенавчання, коли модель демонструє високу точність на тренувальних даних, але погано справляється з новими, невідомими зображеннями. Для вирішення цієї проблеми існує кілька підходів.

По-перше, слід використовувати методи аугментації даних. Це передбачає створення нових тренувальних прикладів шляхом трансформації існуючих зображень. Такі зміни можуть включати обертання, масштабування, зсуви, зміну яскравості та контрастності, віддзеркалення тощо. Аугментація дає змогу значно розширити набір тренувальних даних без необхідності збирання нових зображень, що суттєво підвищує ефективність навчання.

По-друге, варто розглянути використання синтетичних даних. Їх можна створювати за допомогою технологій комп'ютерної графіки або генеративно-змагальних мереж (GAN). Синтетичні дані здатні імітувати реальні сцени та включати об'єкти, подібні до тих, що зустрічаються в реальному світі. Це дозволяє збільшити обсяг тренувального набору й забезпечити більшу різноманітність зображень, що позитивно впливає на точність і надійність моделі.

По-третє, ефективним рішенням є використання попередньо натренованих моделей, таких як ResNet, VGG або MobileNet. Ці моделі, навчені на великих загальних датасетах, наприклад ImageNet, містять знання, які можуть бути адаптовані до специфічної задачі шляхом подальшого навчання на спеціалізованому наборі даних. Використання передтренованих моделей знижує

потребу в обчислювальних ресурсах і часі на тренування, а також покращує кінцеву точність результатів.

Застосування аугментації даних, синтетичних зображень та передтренованих моделей дозволяє ефективно вирішити проблему обмеженого набору даних і забезпечити високу точність і надійність моделі розпізнавання об'єктів навіть за умов обмежених ресурсів.

Для підготовки програми розпізнавання об'єктів була використана мова програмування Python, а для налаштування програми як основний фреймворк було використано PyTorch, який дозволяє ефективно працювати з нейронними мережами. Тренування нейронної мережі виконувалося на датасеті COCO, яка є достатній мірі є репрезентативною та містить широкий набір об'єктів для навчання і розпізнавання. У роботі було треновано модель нейронної мережі YOLOv8 nano. Для моделі YOLOv8 nano було виконано процес тренування протягом 100 епох із застосуванням високопродуктивного графічного процесора Intel HD Graphics 620, який забезпечив швидке і ефективне навчання.

Під час навчання контролювалися такі показники. Для оцінювання втрат з прив'язкою до об'єктів використовувались показники Train/Box Loss та Val/Box Loss відповідно для тренувального і валідаційного наборів даних (таблиця 4.1). Втрати зменшуються із часом, що вказує на поліпшення точності розпізнавання об'єктів моделлю нейронної мережі.

Як показники втрат класифікації для тренувального та валідаційного наборів використовувались Train/Cls Loss і Val/Cls Loss. Ці дві метрики характеризують здатність моделі YOLOv8 правильно класифікувати об'єкти. Інші показники, такі як Train/Dfl Loss і Val/Dfl Loss показують втрати моделі в розпізнаванні деталей об'єктів для тренувального та валідаційного наборів.

Показники Metrics/Precision(B) та Metrics/Recall(B) характеризують точність і повноту моделі для тренувального набору даних. Показник точності вказує на відсоток правильно розпізнаних об'єктів серед усіх ідентифікованих, а показник повноти вказує на відсоток правильно розпізнаних об'єктів серед усіх реальних об'єктів.

Таблиця 4.1 — Результати тренування за епохами

Порядк овий номер епохи	train/ box	train/ cls	train/ dfl	metrics/ precis	metrics/ recall	metrics/ mAP50	metrics/ mAP50 - 95
1	1.415	2.548	1.247	0.7811	0.2047	0.276	0.162
2	1.347	1.399	1.213	0.7464	0.2726	0.306	0.183
3	1.311	1.161	1.199	0.7132	0.2969	0.324	0.190
4	1.282	1.068	1.203	0.6913	0.3022	0.331	0.197
5	1.267	1.036	1.186	0.7247	0.3065	0.335	0.199
6	1.257	0.967	1.175	0.7193	0.3131	0.346	0.206
7	1.241	0.949	1.154	0.7384	0.3069	0.345	0.204
8	1.215	0.934	1.156	0.7295	0.31611	0.351	0.208
9	1.192	0.915	1.160	0.7193	0.3188	0.354	0.209
10	1.175	0.846	1.140	0.7615	0.31321	0.359	0.212

Також використовувались показники Metrics/mAP50(B) та Metrics/mAP50-95(B), які подають середні значення точності по множині порогів IoU у діапазоні від 0,5 до 0,95. Вони представляють собою загальні метрики для оцінювання якості роботи об'єктних детекторів, що враховують як точність, так і повноту по широкому діапазону порогів IoU. Отримані числові значення показників характеризують процес налаштування нейронної мережі і показують про достатній рівень її навчання.

4.3 Оцінка якості роботи програми

Для перевірки якості роботи запропонованого підходу із виділення заданих об'єктів у отриманих зображеннях у зоні відеоспостереження були використанні дані із відкритих джерел на ресурсі YouTube. Для здійснення тестування програмного продукту було завантажено близько 10 різних відеороликів із автомобілями, що рухаються, у різних ситуаціях. Для запропонованого способу

по виділенню заданих об'єктів у різних просторових позиціях для цих об'єктів були здійснені операції тестування для визначення результатів перевірки по підрахунку кількості вірних розпізнавань по відношенню до загальної кількості об'єктів у проведених дослідженнях. Перевірка роботи й тестування розробленого програмного продукту виділення заданих об'єктів зображення були виконані для різної кількості автомобілів, при різних рівнях освітлення цих автомобілів та їх розташування у зоні спостереження.

Перевірка виконувалась як для окремо сформованих статичних зображень, так і при русі автомобілів. Для цього було відібрано 10 відеороликів із автомобілями, що рухаються. Тривалість часу у відібраних відеофайлах була приблизно однаковою із відмінністю не більше ніж на декілька секунд. Такий підхід до вибору часу дозволив здійснити оцінку працездатності створеного програмного засобу. У даному випадку для контролю були відібрані такі відеофайли, у яких не було значних перепадів рівня освітлення за невеликий інтервал часу, тобто щоб не було таких різких скачків рівня освітленості, який є при включенні ліхтарів освітлення у темну пору суток. Кількість автомобілів, що переміщалися, була достатньою для перевірки критерію, як буде працювати та реагувати дана програма на значну кількість автомобілів у отриманому відеозображенні. Приклад розпізнавання автомобілів у русі приведено на рисунку 4.2.

Для тестування використовувались зображення із різними типами автомобілів та на різній відстані від камери спостереження. Проводилася перевірка роботи програного засобу із зображеннями автомобілів без використання фільтру підвищення різкості та із використанням фільтру. Значення ймовірності розпізнавання збільшувалося для зображень, де попередньо було використано фільтр підвищення різкості. Також підвищилася точність розпізнавання для зображень із використанням фільтру підвищення різкості (таблиця 4.2).

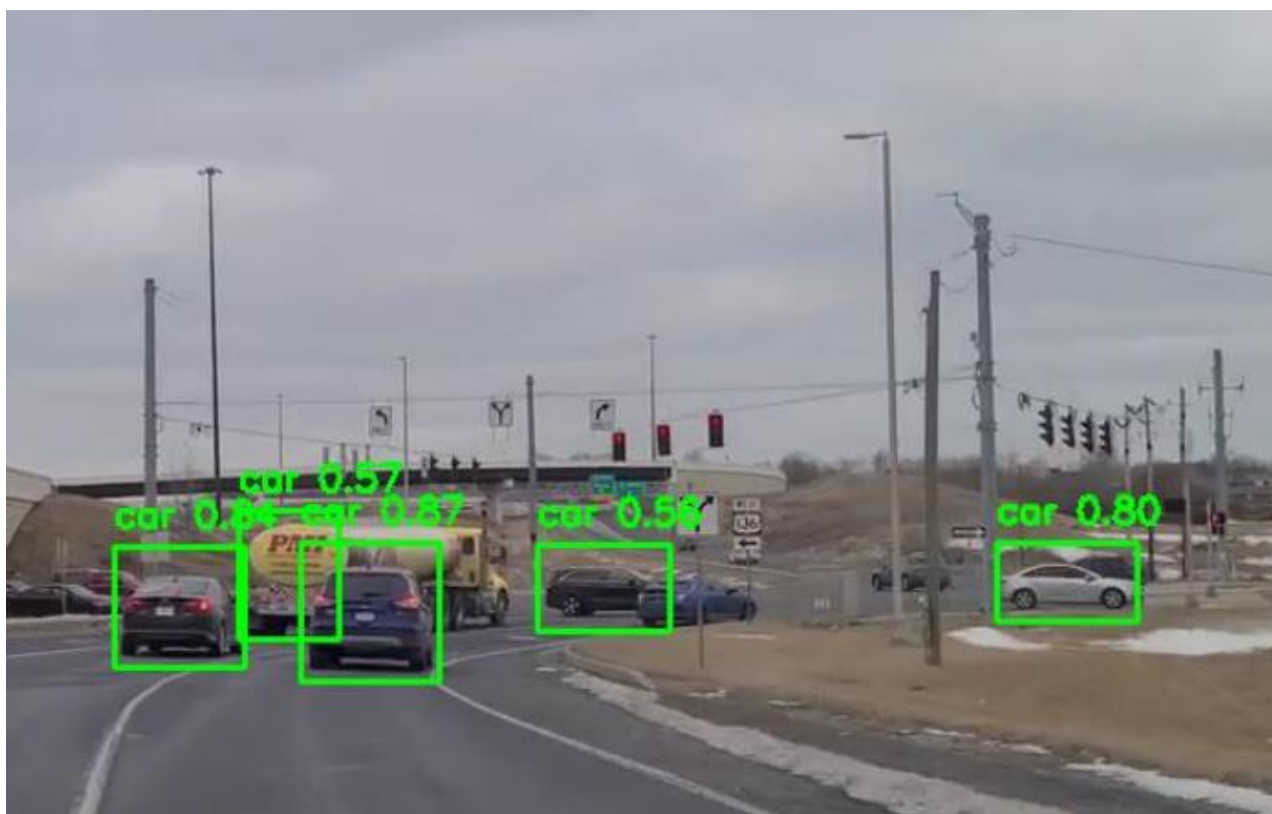


Рисунок 4.2 — Приклад розпізнавання автомобілів у русі

Це особливо помітно для випадків, коли розмір зображення автомобілів є незначним. Як видно із таблиці 4.2, кількість вірно розпізнаних автомобілів зросла 28 штук, що у відсотках становить 1,9%. Отже, запропонований підхід до розпізнавання заданих об'єктів, у якості яких були вибрані автомобілі, показав збільшення точності розпізнавання.

Розроблений програмний засіб виділення заданих об'єктів у відеозображеннях проходив перевірку при різних відстанях між камерою спостереження і вибраними об'єктами. Значення відсотку вірних розпізнавань заданих об'єктів визначалося із розрахунку відношення кількості отриманих вірних результатів розпізнавання об'єктів до сумарної кількості таких же об'єктів у зоні спостереження. Запропонований підхід по виділенню заданих об'єктів у зоні спостереження показав досить точні і адекватні результати, його точність збільшується зі зменшенням відстані між контрольованими об'єктами і камерою відеоспостереження.

Таблиця 4.2 — Кількість розпізнаних автомобілів двома способами розпізнавання

Порядковий номер файлу	Кількість розпізнаних об'єктів способом 1	Кількість розпізнаних об'єктів способом 2
1	140	142
2	179	181
3	162	165
4	93	94
5	169	171
6	142	146
7	88	90
8	153	156
9	181	187
10	98	101
Всього	1405	1433

На підставі проведених експериментів можна стверджувати, що використання фільтру підвищення різкості дозволило збільшити точність розпізнавання заданих об'єктів. Запропонований підхід по пошуку і виділенню заданих об'єктів у зоні відео спостереження можна використовувати у комп'ютерних системах для пошуку, виділення й розпізнавання заданих об'єктів у вибраній сцені відеоспостереження.

У даному розділі магістерської роботи описано перевірку працездатності розробленого програмного засобу, описано використання сформованого датасету та проведено експериментальні дослідження і виконана оцінка якості його роботи.

5 РОЗРАХУНОК ЕКОНОМІЧНОЇ ДОЦІЛЬНОСТІ СТВОРЕННЯ ПРОГРАМНОГО ЗАСОБУ ПОШУКУ І ВИДІЛЕННЯ ЗАДАНИХ ОБ'ЄКТІВ У ЗОНІ ВІДЕОСПОСТЕРЕЖЕННЯ

Метою економічної частини магістерської кваліфікаційної роботи є довести економічну доцільність та ефективність впровадження наукової розробки, для цього необхідно виконати такі етапи:

- оцінити комерційний потенціал розробки;
- спрогнозувати витрати на виконання наукової роботи та впровадження її результатів;
- спрогнозувати комерційний ефект від реалізації результатів розробки;
- розрахувати ефективність вкладених інвестицій та період їх окупності.

5.1 Оцінювання економічного потенціалу розробки

Метою проведення технологічного аудиту є оцінювання комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності.

Для проведення технологічного аудиту залучено трьох незалежних експертів. Кожен з експертів повинен ознайомитися з запропонованою розробкою, та заповнити таблицю, яка визначає рекомендовані критерії оцінювання комерційного потенціалу розробки та їх можливу оцінку в балах. Після виконання цього, підраховується середньоарифметична сума балів та визначається який рівень комерційного потенціалу має нова розробка.

Для проведення технологічного аудиту залучено трьох незалежних експертів. Кожен з експертів повинен ознайомитися з запропонованою розробкою, та заповнити таблицю, яка визначає рекомендовані критерії оцінювання комерційного потенціалу розробки та їх можливу оцінку в балах. Після виконання цього, підраховується середньоарифметична сума балів та визначається який рівень комерційного потенціалу має нова розробка.

Оцінювання комерційного потенціалу розробки здійснюється за критеріями, наведеними в таблиці 5.1 [38].

Таблиця 5.1 — Рекомендовані критерії оцінювання комерційного потенціалу розробки та їх можлива бальна оцінка

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Кри-терій	0	1	2	3	4
Технічна здійсненність концепції:					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність продукту в реальних умовах
Ринкові переваги (недоліки):					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкуренція немає

Продовження таблиці 5.1

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Кри- терій	0	1	2	3	4
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витратити значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. 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 менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Результати оцінювання комерційного потенціалу наведено в табл. 5.2.

Таблиця 5.2 — Результати оцінювання комерційного потенціалу розробки

Критерії	Експерти		
	Експерт 1	Експерт 2	Експерт 3
	Бали, виставлені експертами:		
1	3	4	4
2	4	3	2
3	3	3	4
4	3	3	3
5	3	4	4
6	3	3	2
7	1	2	1
8	2	1	1
9	2	2	3
10	3	3	2
11	3	4	4
12	3	3	2
Сума балів	СБ ₁ =30	СБ ₂ =31	СБ ₃ =32
Середньоарифметична сума балів $\overline{СБ}$	$\overline{СБ} = \frac{\sum_i СБ_i}{i} = \frac{30 + 31 + 32}{3} = 31$		

Отже, з отриманих даних таблиці 5.2 видно, що середньоарифметична сума балів дорівнює 31, тобто нова розробка має рівень комерційного потенціалу вище середнього.

5.2 Розрахунок кошторису витрат на розробку програм

Прогнозування витрат на виконання науково-дослідної чи дослідно-конструкторської роботи може складатися з таких етапів:

- розрахунок витрат, які безпосередньо стосуються виконавців даного розділу роботи.
- розрахунок загальних витрат на виконання даної роботи.
- прогнозування загальних витрат на виконання та впровадження результатів даної роботи.

Розрахунок витрат можна здійснити за такими статтями та формулами:

Основна заробітна плата із розробників (дослідників) $З_0$, якщо вони працюють в наукових установах бюджетної сфери [33]:

$$З_о = \frac{М}{Т_p} \cdot t_{грн.}, \quad (5.1)$$

де $М$ — місячний посадовий оклад конкретного розробника, наукового керівника;

$Т_p$ — кількість робочих днів у місяці, $Т_p = 22$ дні.

t — число робочих днів роботи розробника (дослідника).

Заробітна плата розробника:

$$З_p = \frac{33000}{22} \cdot 60 = 90000(\text{грн})$$

Заробітна плата наукового керівника проекту:

$$З_{нк} = \frac{40000}{22} \cdot 24 = 43636,36(\text{грн})$$

Витрати на оплату праці, основна заробітна плата:

$$З_о = З_p + З_{нк} = 90000 + 43636,36 = 133636,36(\text{грн})$$

Таблиця 4.3 — Розрахунки основної заробітної плати спеціалістів

Найменування посади виконавця	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи, t	Витрати на оплату праці, грн.
Розробник	33000	1500	60	90000
Науковий керівник проекту	40000	1818,18	24	43636,36
Всього:				133636,36

Додаткова заробітна плата $З_d$ всіх розробників та робітників, які брали участь у виконанні даної роботи, розраховується як $(10...12)\%$ від суми основної заробітної плати всіх розробників та робітників, тобто:

$$З_d = (0,1...0,12) \cdot З_о \quad (5.2)$$

Додаткова заробітна плата усіх робітників та розробників:

$$З_d = 0,12 \cdot 133636,36 = 16036,36(\text{грн})$$

Нарахування на заробітну плату $H_{ЗП}$ розробників та робітників, які брали участь у виконанні даної роботи, розраховуються за формулою:

$$H_{ЗП} = (З_0 + З_р + З_д) \cdot \frac{\beta}{100}, \quad (5.3)$$

де $З_0$ — основна заробітна плата розробників, грн.;

$З_д$ — додаткова заробітна плата розробників, грн.;

β — ставка єдиного внеску на загальнообов'язкове державне соціальне страхування, %.

З 01.01.2018 року ставка єдиного внеску на загальнообов'язкове державне соціальне страхування встановлено 22%.

Нарахування на заробітну плату:

$$H_{ЗП} = (133636,36 + 16036,36) \cdot \frac{22}{100} = 32927,99(\text{грн})$$

Амортизаційні відрахування обладнання можна розрахувати за формулою:

$$A = \frac{Ц \cdot H_a}{100} \cdot \frac{T}{12} \text{ грн} \quad (5.4)$$

де $Ц$ — загальна балансова вартість обладнання;

H_a — річна норма амортизаційних відрахувань. Для нашого випадку можна прийняти, що $H_a = 10\text{-}25\%$;

T — термін використання обладнання, $T = 2\text{-}4$ міс.

Витрати на матеріали M , що були використанні під час виконання роботи, розраховуються по кожному виду матеріалів за формулою:

$$M = \sum_1^n H_i \cdot Ц_i \cdot K_i - \sum_1^n B_i \cdot Ц_v \text{ грн.}, \quad (5.5)$$

- де H_i — витрати матеріалу i -го найменування;
 Π_i — вартість матеріалу i -го найменування, грн./кг.;
 K_i — коефіцієнт транспортних витрат, $K_i = (1,1 \dots 1,15)$;
 V_i — маса відходів матеріалу i -го найменування, кг;
 Π_v — ціна відходів матеріалу i -го найменування, грн/кг.

Таблиця 5.4 — Результати розрахунків амортизаційних відрахувань

Найменування обладнання, приміщень тощо	Балансова вартість, грн.	Норма амортизації, %	Термін використання, міс.	Величина амортизаційних відрахувань, грн.
1.Ноутбук	20000	15	2	500
2. Приміщення лабораторії	58000	15	2	1450
3. Камера.	350	25	2	14,6
4. Програмне забезпечення.	4400	20	2	146,6
5. Програма для розпізнавання об'єктів.	9819	10	2	163,6
Всього				A = 2274,8

Таблиця 5.5 — Вартість матеріалів, що були використані для розробки ПЗ

Найменування матеріалу	Одиниці виміру	Ціна, грн.	Витрачено	Вартість витрачених матеріалів, грн.
Папір	Шт.	100	1	100
Всього				100

Загальна вартість витрат матеріалів становить:

$$M = 100 \cdot 1,1 = 110 \text{ (грн)}$$

Використані для розробки нового технічного рішення комплектуючі наведено в табл. 5.6.

Витрати на комплектуючі вироби (B_K) розраховуються згідно з їх номенклатурою за формулою:

$$B_K = \sum_i H_{Ki} \cdot C_{Ki} \cdot K, \quad (5.6)$$

де H_{Ki} — кількість комплектуючих i -го найменування;

C_{Ki} — ціна комплектуючих i -го найменування;

K — коефіцієнт транспортних витрат.

Транспортні витрати на підприємствах складаються з постійних та змінних витрат, що беруть участь в процесі транспортування.

Таблиця 5.6 — Комплектуючі, що використовуються для розробки нового технічного рішення

Найменування комплектуючих	Кількість, шт.	Ціна за одиницю, грн.	Сума, грн.
1.Мікропроцесорна платформа. (2500 грн)	1	2500	2500
2,Відеокамера (1699 грн).	1	1699	1699
3.Лазер (600 грн).	1	600	600
4,Кроковий двигун (450 грн).	1	450	450
5.Електронні компоненти (300 грн).	1	300	300
Всього, грн.			5549

Витрати на силову електроенергію B_e , розраховується за формулою:

$$B_e = B \cdot \Pi \cdot \Phi \cdot K_{пгрн.}, \quad (5.7)$$

де B — вартість 1кВт-год. електроенергії, $B = 10,42$ грн/кВт [39];

Π — установлена потужність обладнання, $\Pi = 0.06$ кВт;

Φ — фактична кількість годин роботи обладнання, $\Phi=480$ год;

$K_{п}$ — коефіцієнт використання потужності, $K_{п} = 0.75$.

Витрати на силову енергію становлять:

$$B_e = 10.42 \cdot 0.06 \cdot 480 \cdot 0.75 = 225,072 \text{ (грн)}$$

Інші витрати I_v можна прийняти як (100...300%) від суми основної заробітної плати розробників, тобто:

$$B_{in} = (1...3) \cdot 3_o \quad (5.8)$$

Інші витрати становлять:

$$B_{in} = 2 \cdot 133636.36 = 267272,72 \text{ (грн)}$$

Сума усіх витрат становить:

$$B_{in} = 133636,36 + 16036,36 + 32927,99 + 2274,8 + 225,072 + 5549 + 110 + 267272,72 = 458032,30 \text{ (грн)}$$

Розрахунок загальних витрат на виконання роботи. Загальна вартість роботи визначається за B_{zag} формулою:

$$B_{zag} = \frac{B}{\alpha} \quad (5.9)$$

де α — частка витрат, які безпосередньо здійснює виконавець даного етапу роботи, у відн. одиницях $\alpha = 1$.

$$B_{\text{заг}} = \frac{458032,30}{1} = 458032,30 \text{ (грн)}$$

Прогнозування загальних витрат на виконання та впровадження результатів. Прогнозування загальних витрат ЗВ на виконання та впровадження результатів здійснюється за формулою:

$$ЗВ = \frac{B_{\text{заг}}}{\beta}, \quad (5.10)$$

де β — коефіцієнт, який характеризує етап (стадію) виконання даної роботи, на стадії розробки промислового зразка, $\beta \approx 0,7$.

$$ЗВ = \frac{458032,30}{0,7} = 654332,43 \text{ (грн)}$$

5.3 Розрахунок експлуатаційних витрат для нової розробки

Виконання наукової роботи та впровадження її результатів буде здійснюватися протягом одного року. Позитивні результати від впровадження розробки очікуються протягом трьох років від впровадження розробки. Одним із основних позитивних результатів є зростання величини прибутку. Зростання чистого прибутку забезпечить підприємству (організації) надходження додаткових коштів, які дозволять покращити фінансові результати діяльності.

Збільшення чистого прибутку підприємства $\Delta\Pi_i$ для кожного із років, протягом яких очікується отримання позитивних результатів від впровадження

розробки, розраховується за формулою:

$$\Delta\Pi_i = \sum_1^n (\Delta\Pi_{\text{я}} \cdot N + \Pi_{\text{я}} \cdot \Delta N)_i, \quad (5.11)$$

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;

$\Pi_{\text{я}}$ — 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 витрати матеріалів на розробку алгоритму зменшаться на 400 грн (що автоматично спричинить збільшення чистого прибутку підприємства на 400 грн), а кількість користувачів збільшиться: протягом першого року — на 250 користувачів, протягом другого року — на 200 користувачів, протягом третього року — на 150 користувачів.

Реалізація продукції до впровадження результатів наукової розробки складала 700 шт., а прибуток, що його отримувало підприємство (організація) на одиницю продукції до впровадження результатів наукової розробки — 500 грн.

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

Збільшення чистого продукту $\Delta\Pi_1$ протягом першого року складатиме:

$$\Delta\Pi_1 = 400 \cdot 700 + (500 + 400) \cdot 250 = 505000(\text{грн})$$

Протягом другого року:

$$\Delta\Pi_2 = 400 \cdot 700 + (500 + 400) \cdot (250 + 200) = 685000(\text{грн})$$

Протягом третього року:

$$\Delta\Pi_3 = 400 \cdot 700 + (500 + 400) \cdot (250 + 200 + 150) = 820000(\text{грн})$$

Чистий прибуток це прибуток після сплати податків. Прибуток також визначають як суму, на яку зріс власний капітал компанії за даний період у результаті діяльності цієї компанії.

5.4 Розрахунок ефективності вкладених інвестицій та періоду їх окупності

В даному підрозділі необхідно кількісно спрогнозувати, яку вигоду, зиск можна отримати у майбутньому від впровадження результатів виконаної роботи.

Розрахунок ефективності вкладених інвестицій передбачає проведення таких робіт.

Крок 1 — розрахуємо теперішню вартість інвестицій PV , що вкладаються в наукову розробку. Такою вартістю ми можемо вважати прогнозовану величину загальних витрат $ЗВ$ на виконання та впровадження результатів НДДКР, розраховану раніше, тобто будемо вважати, що $ЗВ = PV = 654332,43$ (грн).

Крок 2 — 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 нами раніше та становить:

$$\Delta\Pi_1 = 505000(\text{грн}), \Delta\Pi_2 = 685000(\text{грн}), \Delta\Pi_3 = 820000(\text{грн}).$$

Крок 3 — для спрощення подальших розрахунків необхідно побудувати вісь часу, на яку наносять всі платежі (інвестиції та прибутки), що мають місце під час виконання науково-дослідної роботи та впровадження її результатів.

Якщо загальні витрати ЗВ на виконання та впровадження результатів НДДКР (або теперішня вартість інвестицій PV) дорівнюють 654332,43 грн., а результати вкладених у наукову розробку інвестицій почнуть виявлятися вже вкінці другого року впровадження. То ці результати виявляться у тому, що у першому році підприємство отримає збільшення чистого прибутку на 505000 грн. відносно базового року, у другому році — збільшення чистого прибутку на 685000 грн. (відносно базового року), у третьому році — збільшення чистого прибутку на 820000 грн. (відносно базового року).

Тоді рисунок, що характеризує рух платежів (інвестицій та додаткових прибутків) буде мати вигляд, наведений на рис. 5.1.

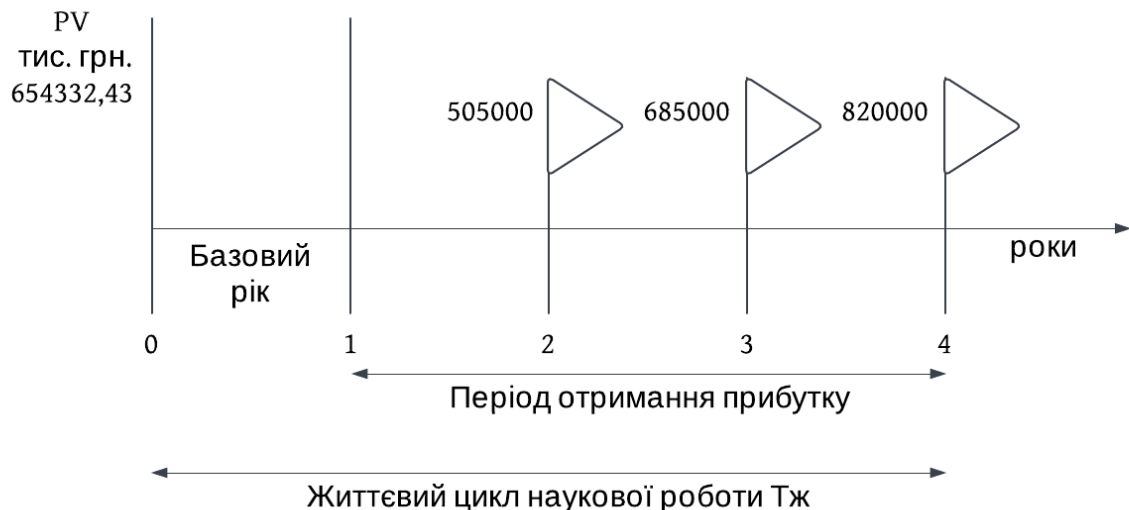


Рисунок 5.1 — Вісь часу з фіксацією платежів, що мають місце під час розробки та впровадження результатів НДДКР

Крок 4 — розрахуємо абсолютну ефективність вкладених інвестицій $E_{\text{абс}}$. Якщо $E_{\text{абс}} \leq 0$ то результати від проведення наукових досліджень та їх

впровадження буде збитковим і вкладати гроші в проведення цих досліджень ніхто не буде.

Для цього показника скористаємося формулою:

$$E_{abc} = (ПП - PV), \quad (5.12)$$

де ПП — приведена вартість всіх чистих прибутків, що їх отримає підприємство (організація) від реалізації результатів наукової розробки, грн.;

PV — теперішня вартість інвестицій $PV = 3B = 654332,43$ грн.

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

$$ПП = \sum_1^T \frac{\Delta\Pi_i}{(1+\tau)^t}, \quad (5.13)$$

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 0,1;

t — період часу (в роках) від моменту отримання чистого прибутку до точки „0”.

$$ПП = \frac{505000}{(1+0,1)^2} + \frac{685000}{(1+0,1)^3} + \frac{820000}{(1+0,1)^4} = 417355,37 + 514651,43 + 560075,27 = 1498082,07(\text{грн})$$

$$E_{абс} = (1498082,07 - 654332,43) = 843749,64(\text{грн})$$

Оскільки $E_{абс} > 0$, то вкладання коштів на виконання та впровадження результатів НДДКР може бути доцільним.

Крок 5 — розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій E_v . Для цього використаємо формулу:

$$E_v = \sqrt[T_{ж}]{1 + \frac{E_{абс}}{PV}} - 1, \quad (5.14)$$

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

PV — теперішня вартість інвестицій $PV = 3B$, грн;

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

Далі, розрахована величина E_v 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:

$$\tau = d + f \quad (5.15)$$

де d — середньозважена ставка за депозитними операціями в комерційних банках; в 2024 році в Україні $d = (0,15...0,17)$;

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

Якщо величина $E_v > \tau_{мін}$, то інвестор може бути зацікавлений у фінансуванні даної наукової розробки. В іншому випадку фінансування наукової розробки здійснюватися не буде.

Спочатку спрогнозуємо величину $\tau_{мін}$. Припустимо, що за даних умов $\tau_{мін} = 0,15 + 0,05 = 0,20$.

Тоді відносна (щорічна) ефективність вкладних інвестицій в проведення наукових досліджень та впровадження їх результатів складе:

$$E_B = \sqrt[4]{1 + \frac{843749,64}{654332,43}} - 1 = 0,22$$

Оскільки $E_B = 22\% > \tau_{\min} = 20\%$, то інвестор буде зацікавлений вкладати гроші в дану наукову розробку.

Крок 6 — розраховують термін окупності вкладених у реалізацію наукового проекту інвестицій. Термін окупності вкладених у реалізацію наукового проекту інвестицій T_{OK} можна розрахувати за формулою:

$$T_{OK} = \frac{1}{E_g} \quad (5.16)$$

Якщо $T_{OK} < 3...5$ -ти років, то фінансування даної наукової розробки в принципі є доцільним. В інших випадках потрібні додаткові розрахунки та обґрунтування.

Термін окупності становить:

$$T_{OK} = \frac{1}{0,22} = 4,55 \text{ (років)} \quad (5.17)$$

$T_{OK} < 5$ років, що свідчить про доцільність фінансування даної наукової розробки.

Рівень комерційного потенціалу нової розробки є вище середнього. Показники ефективності показують, що даний метод є доцільним і буде цікавий

для інвестора. Термін окупності розробленого проекту менше 5-ти років, що підтверджує доцільність вкладання коштів в дану розробку. Економічний ефект від впровадження розробленої програми очікується позитивним при невеликій вартості і складності розробки даного ПЗ, це зробить розпізнавання об'єктів у різних ситуаціях більш доступним у галузі безпеки та охоронній діяльності.

ВИСНОВКИ

Створення систем пошуку та розпізнавання заданих об'єктів із використанням засобів комп'ютерного зору є актуальною задачею сьогодення. Запропоновані рішення у цій області інформаційних технологій знаходять широке застосування в різних сферах діяльності людства: розпізнавання особи за зображенням обличчя, забезпечення захисту від недозволеного проникнення злоумисників на об'єкти, що охороняються, розпізнавання заданих об'єктів у робототехніці, розпізнавання людини за її ходою та інше. У цій магістерській роботі сформована послідовність та розроблені засоби пошуку і виділення заданих об'єктів у отриманих відео кадрах виділеної зони спостереження.

У першому розділі магістерської роботи був проведений аналітичний огляд особливостей функціонування систем комп'ютерного зору, виконаний аналіз методів та систем розпізнавання об'єктів та особливостей їх побудови, розглянуті підходи до формування ознак для виділення і розпізнавання об'єктів та запропоновано використати нейронні мережі глибокого навчання для розпізнавання заданих об'єктів у отриманій послідовності відео кадрів.

У другому розділі магістерської роботи вдосконалено метод пошуку заданих об'єктів у відео послідовності шляхом підкреслення їх контурів та подальшого розпізнавання із використанням нейронної мережі. Основним підходом до пошуку й виділення заданих об'єктів використано пошук області із заданими об'єктами, застосовано фільтри для підкреслення контурів об'єктів та використання згорткової нейронної мережі для розпізнавання заданих об'єктів у отриманому відео зображенні зони спостереження.

У третьому розділі магістерської роботи виконано вибір засобів та середовища розробки програмного продукту, створено програму для реалізації запропонованого підходу по виділенню заданих об'єктів при обробці відео зображення для пошуку об'єктів у вибраній сцені спостереження.

У четвертому розділі магістерської роботи описано перевірку працездатності розробленого програмного засобу та проведено експериментальні дослідження і виконана оцінка якості його роботи.

У п'ятому розділі магістерської роботи виконані економічні розрахунки із обґрунтування доцільності виконання нової роботи по виділенню заданих об'єктів у відео зображеннях, обраховані фінансові затрати на створення програмного засобу та визначено економічні переваги від впровадження запропонованого наукового рішення у вигляді завершеної комп'ютерної програми.

Створений програмний засіб може використовуватися у комп'ютерних системах відеоспостереження для пошуку і виділення заданих об'єктів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Довбиш А. С. Основи теорії розпізнавання образів : навч. посіб. : у 2 ч. / А. С. Довбиш, І. В. Шелехов. – Суми : Сумський державний університет, 2015. — Ч. 1. — 109 с.
2. Кутковецький В. Я. Розпізнавання образів : навчальний посібник / В. Я. Кутковецький. — Миколаїв : Вид-во ЧНУ ім. Петра Могили, 2017. — 420 с.
3. Гороховський О. І. Інтелектуальні системи. /О. І. Гороховський -Вінниця: ВНТУ, 2010. —193с.
4. Крістофор М. Б. Розпізнавання образів і машинне навчання — Київ: Кондор, 2008. — 489 с.
5. Гавриш Б. М. Алгоритми розпізнавання об'єктів на зображенні. / Б. М. Гавриш, О. В. Тимченко, М. П. Кляп //Поліграфія і видавнича справа. 2022 / 1 (83) — с. 47—58.
6. Zhao Z., Zheng P., Xu S., Wu S. Object detection with deep learning: A review. Computer Vision and Pattern Recognition. 2019. №1. P. 1 — 21.
7. Заяць В. М. Методи розпізнавання образів. Навч. посібник. / В. М. Заяць, Р. М. Камінський.- Львів, видав. Національного університету «Львівська політехніка», 2004. — 176 с.
8. Кирилюк О. О. Програмний засіб пошуку і виділення заданих об'єктів у зоні відеоспостереження. / О.О. Кирилюк, Т. Б. Мартинюк, М. А. Очуров // Матеріали LIV Всеукраїнської науково-технічної конференції підрозділів ВНТУ. [Електронний ресурс]. Режим доступу: <https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-2025/paper/view/22948/18923>
9. Meera S, Sharmikha Sree, Manasvinii S R, Haritha V. Object Detection for Visually Challenged Using Image Processing. // International Conference on Communication, Computing and Internet of Things (IC3IoT), 2022. — pp.1 — 8.
10. Herbert B. SURF: Speeded Up Robust Features / B. Herbert, Andreas Ess, Tinne Tuytelaars, Luc Van Gool // Computer Vision and Image Understanding (CVIU). 2008. — Vol. 110, No. 3. — P. 346— 359.

- 11.Марчук Д. Аналіз сучасних алгоритмів виявлення і розпізнавання об'єктів з відеопотоку для систем управління паркуванням в реальному часі. // Вісник Хмельницького національного університету, №3, 2023 (321) с. 17—23.
- 12.Кушнір Д. О. Методи пошуку та розпізнавання об'єктів у відеозображеннях на мобільній платформі IOS в реальному часі / Д. О. Кушнір, Я. С. Парамуд // Комп'ютерні системи та мережі. — Львів : Видавництво Львівської політехніки, 2019. — Том 1. — № 1. — С. 24—34.
- 13.Бабарика А. О. Обґрунтування показника вибору оптимального алгоритму виділення фону у відеопослідовностях з камер відеоспостереження відомчих систем відео спостереження. / А. О. Бабарика // Сучасні інформаційні технології у сфері безпеки та оборони .— 2019. — № 3. — с. 97—10.
- 14.Khan, N.,& Rehman, A. A review on moving object detection and tracking in video surveillance. Multimedia Tools and Applications, 2018, 77(22), 29835 — 29857. doi:10.1007/s11042-018-6979-9.
- 15.Принципи побудови нейронних мереж [Електронний ресурс] — Режим доступу: <https://itmaster.biz.ua/programming/vision/neural-networksprinciples.html>.
- 16.Субботін С. О. Нейронні мережі : теорія та практика: навч. посіб. / С. О. Субботін. — Житомир : Вид. О. О. Євенок, 2020. — 184 с.
- 17.Кушнір Н.О. Використання згорткових нейронних мереж у задачах розпізнавання та класифікації об'єктів зображень / Н.О. Кушнір, Т.М. Локтікова, А.В. Морозов, В.О. Юрченко // Технічна інженерія №1 (89), 2022. — С. 93—100.
- 18.Назаркевич М. А., Олексів Н. Т. Система розпізнавання об'єктів на основі моделі YOLO. Український журнал інформаційних технологій. 2024, т. 6, № 1. С. 120 — 126.
- 19.Олег Жеребукх, Ігор Фармага. Використання нейронних мереж для визначення об'єктів на зображенні Computer design systems. Theory and practice. Vol. 6, №1, 2024 с. 232 — 240.

20. System and method of processing information (images) aimed at tracking a person through multiple cameras. Патент US №9390332 B2, МПК G06Т 7/00, опубл. 12.07.2016.
21. Цивадиць Павло. Порівняння методів виявлення об'єктів в комп'ютерному зорі. / Павло Цивадиць, Тетяна Скрипник, Леонід Вознюк. // Вісник Хмельницького національного університету, №2, 2024 (333) С. 265 — 268.
22. Борківський, Б. П., & Теслюк, В. М. (2023). Використання нейромережевих засобів для розпізнавання об'єктів у мобільних системах з обходом перешкод. Scientific Bulletin of UNFU, 33(4), с. 84 — 89. <https://doi.org/10.36930/40330412>.
23. Марчук Дмитро. Аналіз сучасних алгоритмів виявлення і розпізнавання об'єктів з відеопотоку для систем управління паркуванням в реальному часі Вісник Хмельницького національного університету, №3, 2023 (321) с.17—23.
24. Дзюбаненко А. В. Локалізація об'єктів та вилучення основних ознак з відеопотоку в комп'ютеризованих системах телеспостереження/ А. В. Дзюбаненко // Проблеми інформатизації та управління, №4(56), 2016. — С. 34—37.
25. Синєглазов В. Глибокі нейронні мережі для вирішення завдань розпізнавання і класифікації зображення [Електронний ресурс] / В. Синєглазов, О. Чумаченко // «Інформаційні технології та комп'ютерне моделювання»: матеріали ст. Міжнар. наук.-практ. конф., м. Івано-Франківськ, 15—20 трав. 2017 р. — Івано-Франківськ, 2017. — С. 274 — 277.
26. Redmon, J. You only look once: Unified, real-time object detection. / J. Redmon, S. Divvala, R. Girshick & A. Farhadi // In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016. — pp. 779 — 788.
27. Diwan T. Object detection using YOLO: Challenges, architectural successors, datasets and applications. / T. Diwan, G. Anirudh, JV Tembhurne. // Multimedia Tools and Applications, 2023. — Springer — 26 p.
28. Shaoqing Ren. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks / Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun // IEEE

- Transactions on Pattern Analysis and Machine Intelligence. June 1 2017, № 6 (39), pp. 1137 —1149.
- 29.Zhan, L., Fang, Y., & Cao, Z. Moving object detection using deep convolutional neural networks. Journal of Visual Communication and Image Representation. 2018. №51, pp 160 — 170. doi: 10.1016/j.jvcir.2017.11.003.
 - 30.Guido van Rossum, Fred L. Drake, Jr. Python Software Foundation. — Springer, 2017. — 826 p.
 - 31.Середовище PyCharm. [Електронний ресурс] — Режим доступу: [https://uk.wikipedia.org › wiki › PyCharm](https://uk.wikipedia.org/wiki/PyCharm).
 - 32.OpenCV documentation [Електронний ресурс] — Режим доступу: http://docs.opencv.org/3.0beta/doc/py_tutorials/py_video/py_meanshift/py_meanshift.html?highlight=camshaft.
 - 33.Засоби фірми Hikvision [Електронний ресурс] — Режим доступу: <https://hikvision.org.ua/ua/node/1009>.
 - 34.Камери відеоспостереження. [Електронний ресурс] — Режим доступу: https://відеокамери.com.ua/videocamera/external_cameras.
 - 35.Медіаконвертери [Електронний ресурс] — Режим доступу: <https://lantorg.com/catalog/mediakonverter>.
 - 36.Сервери. [Електронний ресурс] — Режим доступу: <https://server-shop.ua/servers/>.
 - 37.COCO. [Електронний ресурс] — Режим доступу: <https://cocodataset.org/#home>
 - 38.Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт / Уклад. : В. О. Козловський, О. Й. Лесько, В. В. Кавецький. — Вінниця : ВНТУ, 2021. — 42 с.
 - 39.Тарифи на електроенергію [Електронний ресурс]. Режим доступу: <http://index.minfin.com.ua/tarif/electric.php>.

ДОДАТОК А

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

ЗАТВЕРДЖУЮ

Завідувач кафедри ОТ

проф., д.т.н. Азаров О.Д.

" "

2024 р.

ТЕХНІЧНЕ ЗАВДАННЯ

на виконання магістерської кваліфікаційної роботи

Програмний засіб пошуку і виділення заданих об'єктів у зоні

відеоспостереження

08-54.МКР.011.00.000 ПЗ

Науковий керівник: д.т.н., професор

_____ Мартинюк Т. Б.

Студент групи 1КІ-23м

_____ Кирилюк О.О.

Вінниця 2024

1 Підстава для виконання магістерської кваліфікаційної роботи (МКР)

1.1 Актуальність дослідження пов'язана із широким застосуванням засобів відеоспостереження у місцях скупчення великої кількості людей, надання доступу у банках, на кордоні, у боротьбі із злочинністю та інших сферах діяльності людей, що пов'язана із процесом розпізнавання особи за зображенням її обличчя.

1.2 Наказ про затвердження теми МКР.

2 Мета МКР і призначення розробки

2.1 Мета роботи — аналіз сучасного стану і тенденцій розвитку зарубіжних і вітчизняних розробок у сфері розпізнавання обличчя людини у засобах відеоспостереження і вдосконалення технології розпізнавання обличчя людини у засобах відеоспостереження.

2.2 Призначення розробки — створення програмного продукту для виконання процесу розпізнавання особи за зображенням її обличчя.

3 Вихідні дані для виконання МКР

3.1 Технічний опис сучасних методів розпізнавання людей в відеопотоці.

3.2 Середовище розробки ПЗ Visual Studio Code.

4 Вимоги до виконання МКР

4.1 Провести аналіз різних типів технологій виявлення і розпізнавання особи у відеозображенні.

4.2 Вивчити принцип роботи та особливості систем розпізнавання особи у відеозображенні.

4.3 Запропонувати метод розпізнавання особи у відеозображенні.

4.4 Розробити технологію виявлення і розпізнавання особи у відеозображенні.

5 Етапи МКР та очікувані результати

Етапи роботи та очікувані результати приведено в Таблиці А.1.

Таблиця А.1 — Етапи МКР

№ етап у	Назва етапу	Термін виконання		Очікувані результати
		початок	кінець	
1	Аналіз методів та засобів виділення особи у відеозображенні	20.09.2024р.	30.09.2024р.	Розділ 1
2	Розробка нейромережевої технології виявлення і розпізнавання особи у відеозображенні	01.10.2024р.	04.10.2024р.	Розділ 2
3	Розробка програми виявлення і розпізнавання особи у відеозображенні	05.10.2024р.	18.10.2024р.	Розділ 3
4	Перевірка працездатності програмного продукту	19.10.2024р.	01.11.2024р.	Розділ 4
5	Розрахунок економічної доцільності створення програми виявлення і розпізнавання особи у відеозображенні	02.11.2024р.	15.11.2024р.	Розділ 5
6	Апробація та впровадження результатів дослідження	16.11.2024р.	30.11.2024р.	Тези доповіді
7	Оформлення пояснювальної записки, графічного матеріалу і презентації	02.12.2024р.	10.12.2024р.	ПЗ, графічний матеріал
8	Підготовка і підпис супроводжуючих документів, нормоконтроль та тест на плагіат	10.12.2024р.	12.12.2024р.	Оформлені документи

6 Матеріали, що подаються до захисту МКР

До захисту подаються: пояснювальна записка МКР, графічні і ілюстративні матеріали, протокол попереднього захисту МКР на кафедрі, відзив наукового керівника, відзив опонента, протоколи складання державних екзаменів, анотації до МКР українською та іноземною мовами, нормоконтроль про відповідність оформлення МКР діючим вимогам.

7 Порядок контролю виконання та захисту МКР

Виконання етапів графічної та розрахункової документації МКР контролюється науковим керівником згідно зі встановленими термінами.

Захист МКР відбувається на засіданні Державної екзаменаційної комісії, затвердженою наказом ректора.

8 Вимоги до оформлення МКР

8.1 При оформлюванні МКР використовуються:

— ДСТУ 3008: 2015 «Звіти в сфері науки і техніки. Структура та правила оформлювання»;

— ДСТУ 8302: 2015 «Бібліографічні посилання. Загальні положення та правила складання»;

— ГОСТ 2.104-2006 «Єдина система конструкторської документації. Основні написи»;

— методичні вказівки до виконання магістерських кваліфікаційних робіт зі спеціальності 123 «Комп'ютерна інженерія»;

— документи на які посилаються у вище вказаних.

8.2 Порядок виконання МКР викладено в «Положення про кваліфікаційні роботи на другому (магістерському) рівні вищої освіти СУЯ ВНТУ-03.02.02-П.001.01:21».

ДОДАТОК Б

Методи розпізнавання об'єкта

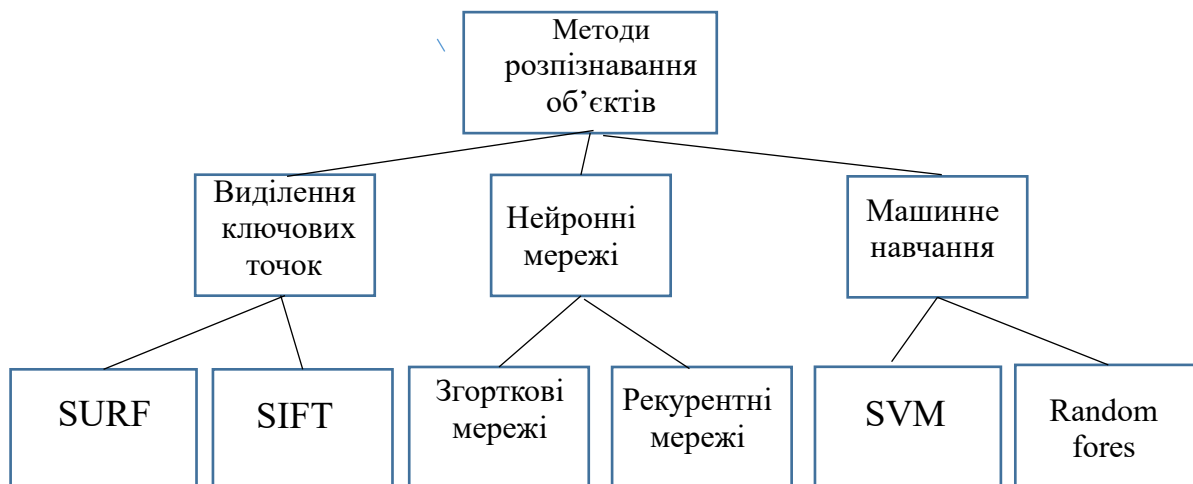


Рисунок Б.1 — Методи розпізнавання об'єкта

ДОДАТОК В

Типи нейронних мереж

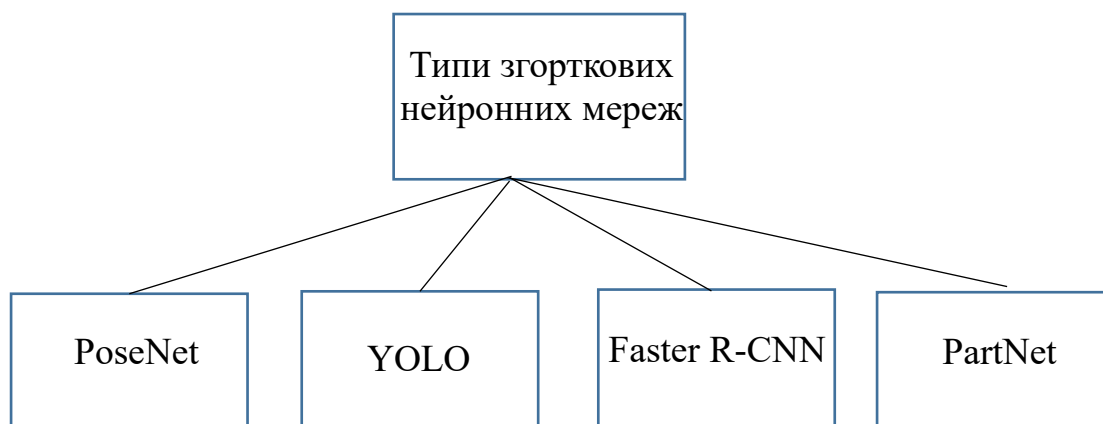


Рисунок В.1 — Типи нейронних мереж

ДОДАТОК Г

Структура програми розпізнавання об'єкта

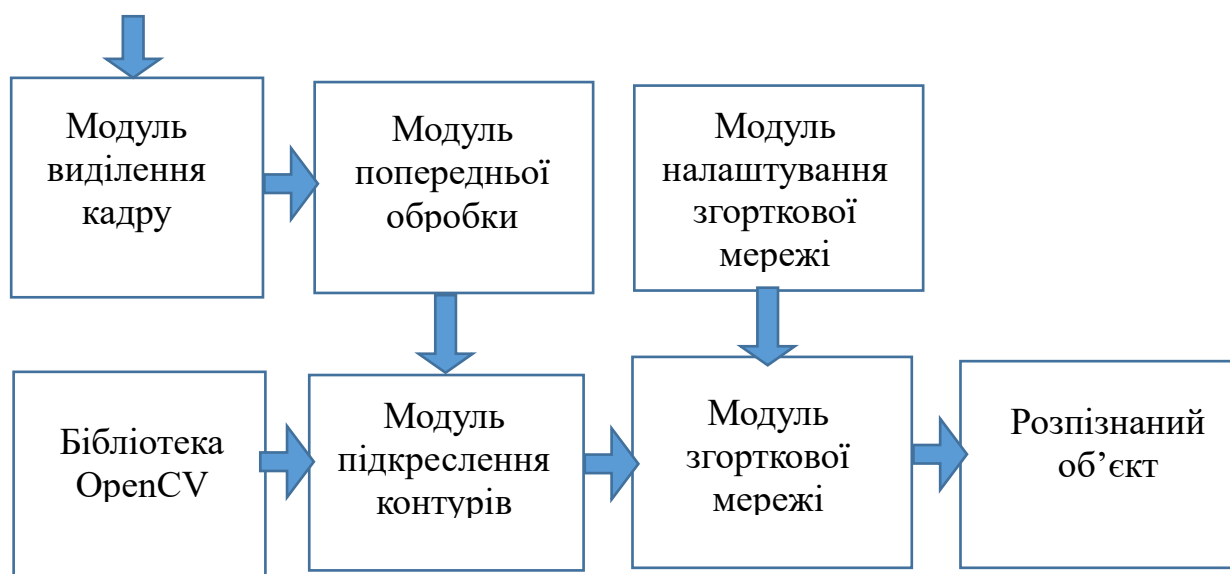


Рисунок Г.1 — Структура програми розпізнавання об'єкта

ДОДАТОК Д

Лістинг обробки відеокадру

```

import numpy as np
import cv2
import imutils
import dlib
import datetime
from Faster_RCNN.config import Config
from Faster_RCNN.utils import some_function
python convert.py = Faster R-CNN.cfg Faster R-CNN.weights model_data/
Faster R-CNN.h5
model = FASTER R-CNN (mode='inference',
model_dir=MODEL_DIR,
config= Faster R-CNN Config())
height, width, depth = image_to_process.shape
blob = cv2.dnn.blobFromImage(image_to_process, 1 / 255, (608, 608), (0, 0,
0), swapRB=True, crop=False)
net.setInput(blob)
outs = net.forward(out_layers)
class_indexes, class_scores, boxes = ([[] for i in range(3)])
objects_count = 0
if classes[class_index] in classes_to_look_for:
objects_count += 1
image_to_process = draw_object_bounding_box(image_to_process,
class_index, box)
final_image = draw_object_count(image_to_process, objects_count)
return final_image
cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 255, 0), 1)

```

```

cv2.imshow('Video', frame)
from imutils.video import VideoStream, FPS
from PIL import ImageGrab
from tkinter import *
run = True
frame = 0
def foo(event, x, y, flags, param):
    global run
    global frame
    if event == cv2.EVENT_LBUTTONDOWN:
        run = not run
        while run:
            frame += 1
            frame = cap.read()[1]
            cv2.imshow(window_name, frame)
            key = cv2.waitKey(5) & 0xFF
            if key == ord('v'):
                pass
            elif event == cv2.EVENT_RBUTTONDOWN:
                pass
        vs = VideoStream(src=0).start()
        detector = dlib.get_frontal_auto_detector()
        predictor = dlib.shape_predictor(
            while True:
                frame = vs.read()
                grayFrame = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
                autos = detector(grayFrame)
                for auto in autos:
                    cv2.putText(frame, 'The camera image revealed a

```

```

auto:&quot;.format(len(autos)), (10, 40), cv2.FONT_HERSHEY_SIMPLEX, 0.6,
(255, 0,
0),2)
cv2.circle(frame, (x, y), 1, (0, 0, 255), 2)
cv2.putText(frame, &quot;Press the ESC button to exit the program window&quot;,
(10,
20), cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 0), 2)
cv2.imshow(&quot;The result of the master&#39;s thesis program&quot;, frame)
key = cv2.waitKey(1)
if key == 27:
    break
tk = Tk()
tk.geometry(&#39;250x250+200+100&#39;)
txt = Text(tk)
txt.pack()
def event_info(event):
    txt.delete(&quot;1.0&quot;, END)
    screen = ImageGrab.grab()
    screen.save(&#39;screenshot.png&#39;, &#39;PNG&#39;)
    txt.insert(END, &#39;&#39;)
    txt.bind(&quot;&lt;space&gt;&quot;, event_info)
tk.mainloop()
input(&#39;Press &quot;Enter&quot; to exit&#39;)
def Load Autos():
    return known_ auto _encodings, known_ auto_names;
video_capture = cv2.VideoCapture(0)
known_ auto _encodings, known_ auto_names = LoadFaces()
while True:
    ret, frame = video_capture.read()
    rgb_frame = frame[:, :, :-1]

```

```

auto_locations = auto_recognition. auto_locations(rgb_frame)
auto_encodings = auto_recognition. auto_encodings(rgb_frame, auto
_locations)
for (top, right, bottom, left), auto_encoding in zip(auto_locations,
auto_encodings):
matches = auto_recognition.compare_ autos(known_ auto_encodings,
auto _encoding)
name = &quot;Unknown&quot;
auto_distances = auto_recognition. auto_distance(known_
auto_encodings, auto_encoding)
best_match_index = np.argmin(auto_distances)
if matches[best_match_index]:
name = known_ auto_names[best_match_index]
cv2.rectangle(frame, (left, top), (right, bottom), (0, 0, 255), 1)
cv2.rectangle(frame, (left, bottom - 25), (right, bottom), (0, 0, 255),
cv2.FILLED)
font = cv2.FONT_HERSHEY_COMPLEX_SMALL
cv2.putText(frame, name, (left + 6, bottom - 6), font, 0.7, (255, 255, 255),
cv2.imshow(&#39;The result of the my diploma program&#39;, frame)
if cv2.waitKey(1) &amp; 0xFF == ord(&#39;q&#39;):
break
video_capture.release()
cv2.destroyAllWindows()

```

ДОДАТОК Е

Лістинг налаштування мережі

```
import numpy as np
import tensorflow as tf
import vgg16
import rpn
import detector
import math_utils

class FasterRCNNModel(tf.keras.Model):
    def __init__(self, num_classes, allow_edge_proposals, custom_roi_pool,
        activate_class_outputs, l2 = 0, dropout_probability = 0):
        super().__init__()
        self._num_classes = num_classes
        self._activate_class_outputs = activate_class_outputs
        self._stage1_feature_extractor = vgg16.FeatureExtractor(l2 = l2)
        self._stage2_region_proposal_network = rpn.RegionProposalNetwork(
            max_proposals_pre_nms_train = 12000,
            max_proposals_post_nms_train = 2000,
            max_proposals_pre_nms_infer = 6000,
            max_proposals_post_nms_infer = 300,
            l2 = l2,
            allow_edge_proposals = allow_edge_proposals
        )
        self._stage3_detector_network = detector.DetectorNetwork(
            num_classes = num_classes,
            custom_roi_pool = custom_roi_pool,
            activate_class_outputs = activate_class_outputs,
            l2 = l2,
            dropout_probability = dropout_probability
        )
```



```

def call(self, inputs, training=False):
    input_image = inputs[0] )
    anchor_map = inputs[1] )
    anchor_valid_map = inputs[2] )
    if training:
        gt_rpn_map = inputs[3] )
        gt_box_class_idxes_map = inputs[4]
        gt_box_corners_map = inputs[5] )
        feature_map = self._stage1_feature_extractor(input_image = input_image,
training
            = training)
        rpn_scores, rpn_box_deltas, proposals =
self._stage2_region_proposal_network(
    inputs = [
        input_image,
        feature_map,
        anchor_map,
        anchor_valid_map
    ],
    training = training
)
    if training:
        proposals, gt_classes, gt_box_deltas = self._label_proposals(
proposals = proposals,
        gt_box_class_idxes = gt_box_class_idxes_map[0],
        gt_box_corners = gt_box_corners_map[0],
        min_background_iou_threshold = 0.0,
        min_object_iou_threshold = 0.5
        )
        proposals, gt_classes, gt_box_deltas = self._sample_proposals(

```

```

proposals = proposals,
gt_classes = gt_classes,
gt_box_deltas = gt_box_deltas,
max_proposals = 128,
positive_fraction = 0.25
)
gt_classes = tf.expand_dims(gt_classes, axis = 0)
gt_box_deltas = tf.expand_dims(gt_box_deltas, axis = 0)
proposals = tf.stop_gradient(proposals)
gt_classes = tf.stop_gradient(gt_classes)
gt_box_deltas = tf.stop_gradient(gt_box_deltas)
detector_classes, detector_box_deltas = self._stage3_detector_network(
inputs = [
input_image,
feature_map,
proposals
],
training = training
)
if training:
rpn_class_loss = self._stage2_region_proposal_network.class_loss(y_predicted
=
rpn_scores, gt_rpn_map = gt_rpn_map)
rpn_regression_loss =
self._stage2_region_proposal_network.regression_loss(y_predicted =
rpn_box_deltas, gt_rpn_map = gt_rpn_map)
detector_class_loss = self._stage3_detector_network.class_loss(y_predicted =
detector_classes, y_true = gt_classes, from_logits = not
self._activate_class_outputs)
detector_regression_loss =

```

```

        self._stage3_detector_network.regression_loss(y_predicted =
detector_box_deltas,
        y_true = gt_box_deltas)
        self.add_loss(rpn_class_loss)
        self.add_loss(rpn_regression_loss)
        self.add_loss(detector_class_loss)
        self.add_loss(detector_regression_loss)
        self.add_metric(rpn_class_loss, name = &quot;rpn_class_loss&quot;);
        self.add_metric(rpn_regression_loss, name =
&quot;rpn_regression_loss&quot;);
        self.add_metric(detector_class_loss, name = &quot;detector_class_loss&quot;);
        self.add_metric(detector_regression_loss, name =
&quot;detector_regression_loss&quot;);
    else:
        rpn_class_loss = float(&quot;inf&quot;);
        rpn_regression_loss = float(&quot;inf&quot;);
        detector_class_loss = float(&quot;inf&quot;);
        detector_regression_loss = float(&quot;inf&quot;);
    return [
        rpn_scores,
        rpn_box_deltas,
        detector_classes,
        detector_box_deltas,
        proposals,
        rpn_class_loss,
        rpn_regression_loss,
        detector_class_loss,
        detector_regression_loss
    ]

def predict_on_batch(self, x, score_threshold):

```

```

_, _, detector_classes, detector_box_deltas, proposals, _, _, _, _ =
super().predict_on_batch(x = x)
scored_boxes_by_class_index = self._predictions_to_scored_boxes(
input_image = x[0],
classes = detector_classes,
box_deltas = detector_box_deltas,
proposals = proposals,
score_threshold = score_threshold
)
return scored_boxes_by_class_index
def load_imagenet_weights(self):
keras_model = tf.keras.applications.VGG16(weights = &quot;imagenet&quot;)
for keras_layer in keras_model.layers:
weights = keras_layer.get_weights()
if len(weights) > 0:
vgg16_layers = self._stage1_feature_extractor.layers +
self._stage3_detector_network.layers
our_layer = [ layer for layer in vgg16_layers if layer.name ==
keras_layer.name
]
if len(our_layer) > 0:
print(&quot;Loading VGG-16 ImageNet weights into layer: %s&quot; %
our_layer[0].name)
our_layer[0].set_weights(weights)
def _predictions_to_scored_boxes(self, input_image, classes, box_deltas,
proposals,
score_threshold):
input_image = np.squeeze(input_image, axis = 0)
classes = np.squeeze(classes, axis = 0)
box_deltas = np.squeeze(box_deltas, axis = 0)

```

```

if not self._activate_class_outputs:
    classes = tf.nn.softmax(classes, axis = 1).numpy()
    proposal_anchors = np.empty(proposals.shape)
    proposal_anchors[:,0] = 0.5 * (proposals[:,0] + proposals[:,2])
    proposal_anchors[:,1] = 0.5 * (proposals[:,1] + proposals[:,3])
    proposal_anchors[:,2:4] = proposals[:,2:4] - proposals[:,0:2]
    boxes_and_scores_by_class_idx = {}
    for class_idx in range(1, classes.shape[1]):
        box_delta_idx = (class_idx - 1) * 4
        box_delta_params = box_deltas[:, (box_delta_idx + 0) : (box_delta_idx + 4)]
        proposal_boxes_this_class = math_utils.convert_deltas_to_boxes(
            box_deltas = box_delta_params,
            anchors = proposal_anchors,
            box_delta_means = [0.0, 0.0, 0.0, 0.0],
            box_delta_stds = [0.1, 0.1, 0.2, 0.2]
        )
        proposal_boxes_this_class[:,0::2] = np.clip(proposal_boxes_this_class[:,0::2], 0,
            input_image.shape[0] - 1)
        proposal_boxes_this_class[:,1::2] = np.clip(proposal_boxes_this_class[:,1::2], 0,
            input_image.shape[1] - 1)
        scores_this_class = classes[:,class_idx]
        sufficiently_scoring_idxs = np.where(scores_this_class >=
score_threshold)[0]
        proposal_boxes_this_class =
        proposal_boxes_this_class[sufficiently_scoring_idxs]
        scores_this_class = scores_this_class[sufficiently_scoring_idxs]
        boxes_and_scores_by_class_idx[class_idx] = (proposal_boxes_this_class,
            scores_this_class)
        scored_boxes_by_class_idx = {}
        for class_idx, (boxes, scores) in boxes_and_scores_by_class_idx.items():

```

```

idxs = tf.image.non_max_suppression(
boxes = boxes,
scores = scores,
max_output_size = proposals.shape[0],
iou_threshold = 0.3
)
idxs = idxs.numpy()
boxes = boxes[idxs]
scores = np.expand_dims(scores[idxs], axis = 0)
scored_boxes = np.hstack([ boxes, scores.T ])
scored_boxes_by_class_idx[class_idx] = scored_boxes
return scored_boxes_by_class_idx

def _label_proposals(self, proposals, gt_box_class_idx, gt_box_corners,
min_background_iou_threshold, min_object_iou_threshold):
proposals = tf.concat([ proposals, gt_box_corners ], axis = 0)
ious = math_utils.tf_intersection_over_union(boxes1 = proposals, boxes2 =
gt_box_corners)
best_ious = tf.math.reduce_max(ious, axis = 1)
box_idx = tf.math.argmax(ious, axis = 1)
gt_box_class_idx = tf.gather(gt_box_class_idx, indices = box_idx)
gt_box_corners = tf.gather(gt_box_corners, indices = box_idx)
idxs = tf.where(best_ious >= min_background_iou_threshold)
proposals = tf.gather_nd(proposals, indices = idxs)
best_ious = tf.gather_nd(best_ious, indices = idxs)
gt_box_class_idx = tf.gather_nd(gt_box_class_idx, indices = idxs)
gt_box_corners = tf.gather_nd(gt_box_corners, indices = idxs)
retain_mask = tf.cast(best_ious >= min_object_iou_threshold, dtype =
gt_box_class_idx.dtype)
gt_box_class_idx = gt_box_class_idx * retain_mask
num_classes = self._num_classes

```

```

gt_classes = tf.one_hot(indices = gt_box_class_idx, depth = num_classes)
proposal_centers = 0.5 * (proposals[:,0:2] + proposals[:,2:4])
proposal_sides = proposals[:,2:4] - proposals[:,0:2]
gt_box_centers = 0.5 * (gt_box_corners[:,0:2] + gt_box_corners[:,2:4])
gt_box_sides = gt_box_corners[:,2:4] - gt_box_corners[:,0:2]
detector_box_delta_means = tf.constant([0, 0, 0, 0], dtype = tf.float32)
detector_box_delta_stds = tf.constant([0.1, 0.1, 0.2, 0.2], dtype = tf.float32)
tyx = (gt_box_centers - proposal_centers) / proposal_sides
thw = tf.math.log(gt_box_sides / proposal_sides)
box_delta_targets = tf.concat([ tyx, thw ], axis = 1)
box_delta_targets = (box_delta_targets - detector_box_delta_means) /
detector_
box_delta_stds
gt_box_deltas_mask = tf.repeat(gt_classes, repeats = 4, axis = 1)[:,:4:]
gt_box_deltas_values = tf.tile(box_delta_targets, multiples = [1, num_classes - 1])
gt_box_deltas_mask = tf.expand_dims(gt_box_deltas_mask, axis = 0)
gt_box_deltas_values = tf.expand_dims(gt_box_deltas_values, axis = 0)
gt_box_deltas = tf.concat([ gt_box_deltas_mask, gt_box_deltas_values ], axis = 0)
gt_box_deltas = tf.transpose(gt_box_deltas, perm = [ 1, 0, 2])
return proposals, gt_classes, gt_box_deltas

```

ДОДАТОК Ж

ПРОТОКОЛ ПЕРЕВІРКИ НАВЧАЛЬНОЇ (КВАЛІФІКАЦІЙНОЇ) РОБОТИ

Назва роботи: Програмний засіб пошуку і виділення заданих об'єктів у зоні відеоспостереження

Тип роботи: Магістерська кваліфікаційна робота

(кваліфікаційна робота, курсовий проект (робота), реферат, аналітичний огляд, інше (зазначити))

Підрозділ кафедра обчислювальної техніки
(кафедра, факультет (інститут), навчальна група)

Керівник Мартинюк Т.Б., проф. каф. ОТ
(прізвище, ініціали, посада)

Показники звіту подібності

Turnitin	
Оригінальність	91%
Загальна схожість	9%

Аналіз звіту подібності (відмітити потрібне)

- ☒ Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
- ☐ Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її автора. Роботу направити на доопрацювання.
- ☐ Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Заявляю, що ознайомлений (-на) з повним звітом подібності, який був згенерований Системою щодо роботи (додається)

Автор _____ Кирилюк О.О.
(підпис) (прізвище, ініціали)

Опис прийнятого рішення

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

Експерт _____
(за потреби) (підпис) (прізвище, ініціали, посада)