Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Автоматизації та комп'ютерно-інтегрованих технологій
(повна назва)

Кафедра Комп'ютерно-інтегрованих технологій, автоматизації та мехатроніки
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Розроблення бортової багатозонової системи комп`ютерного зору із функціями розпізнавання та ідентифікації
(тема)

Виконав:
студент 2 курсу, групи КІТПВм-21-1
Каплін О.П.
(прізвище, ініціали)

Спеціальність 151 Автоматизація та комп'ютерно-інтегровані технології
(код і повна назва спеціальності)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Комп'ютерно-інтегровані технологічні процеси і виробництва
(повна назва освітньої програми)

Керівник проф. Цимбал О. М.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри КІТАМ _____ Невлюдов І. Ш.
(підпис) (прізвище, ініціали)

2022 р.

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Факультет Автоматизації та комп'ютерно-інтегрованих технологій

Кафедра КІТАМ

Рівень вищої освіти другий (магістерський)

Спеціальність 151 Автоматизація та комп'ютерно-інтегровані технології
(код і повна назва)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Комп'ютерно-інтегровані технологічні процеси і виробництва
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)
«____»_____ 20 ___ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Капліну Олександру Павловичу
(прізвище, ім'я, по батькові)

1. Тема роботи «Розроблення бортової багатозонової системи комп`ютерного зору із функціями розпізнавання та ідентифікації»
затверджена наказом університету від « 07 » грудня 2022 р. № 1464 Ст
2. Термін подання студентом роботи до екзаменаційної комісії «13» грудня 2022 р.
3. Вихідні дані до роботи Бібліотека комп'ютерного зору OpenCV. Мова програмування Python. Нейронна мережа YOLO. Середовище програмування Microsoft Visual Studio Code. Застосунок для створення набору даних із зображень Label Image.
4. Перелік питань, що потрібно опрацювати в роботі
4.1 Аналіз існуючих систем комп'ютерного зору в області виявлення вибухонебезпечних об'єктів;
4.2 Розробка архітектури бортової багатозонової системи;
4.3 Розробка бортової багатозонової системи;
4.4 Перевірка достовірності роботи бортової багатозонової системи;
4.5 Охорона праці.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п. 5 включається до завдання за рішенням випускової кафедри) демонстраційний матеріал, представлений у форматі презентації PowerPoint (*.ppt) на аркушах формату А4(12-16 сторінок)

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання та аналіз завдання	05.11.22	Виконано
2	Аналіз існуючих систем комп'ютерного зору в	06.11.22	Виконано
	області виявлення вибухонебезпечних об'єктів		
3	Розробка архітектури систем	10.11.22	Виконано
4	Розробка бортової багатозонової системи	14.11.22	Виконано
5	Перевірка достовірності роботи бортової	15.11.22	Виконано
	багатозонової системи		
6	Оформлення пояснювальної записки	04.12.22	Виконано
7	Підготовка презентації	05.12.22	Виконано
8	Подання закінченої роботи науковому керівникові	06.12.22	Виконано
9	Подання роботи на рецензування	07.12.22	Виконано
10	Попередній захист	08.12.22	Виконано
11	Подання роботи до екзаменаційної комісії	13.12.22	Виконано

Дата видачі завдання _____

Студент _____ Каплін О.П.
(підпис)

Керівник роботи _____ проф. Цимбал О. М.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 87 с., 3 табл., 45 рис., 1 дод., 32 джерела.

КОМП’ЮТЕРНИЙ ЗІР, ГЛИБИННІ НЕЙРОННІ МЕРЕЖІ, РОЗПІЗНАВАННЯ ОБРАЗІВ, ІДЕНТИФІКАЦІЯ, БАГАТОЗОНОВІСТЬ.

Об'єкт дослідження – процеси інтелектуального керування мобільними робототехнічними платформами.

Предмет дослідження – реалізація багатозонової бортової системи комп'ютерного зору.

Мета кваліфікаційної роботи – дослідження застосування методів комп'ютерного зору у завданнях ідентифікації вибухонебезпечних предметів.

Методи дослідження – аналіз існуючих підходів до задач комп’ютерного зору, аналіз релевантних джерел, аналіз існуючих бортових багатозонових систем комп`ютерного зору.

Область застосування – роботизовані системи гуманітарного розмінування.

Проаналізовано існуючі системи комп'ютерного зору в області виявлення вибухонебезпечних об’єктів. Розроблено архітектуру бортової багатозонової системи. Розроблено бортову багатозонову систему. Проведено експерименти з перевірки роботи бортової багатозонової системи.

Результати кваліфікаційної роботи опубліковані у матеріалах 2-х міжнародних конференціях.

ABSTRACT

Explanatory note: 87 p., 3 tab., 45 fig., 1 add, 32 sources.

COMPUTER VISION, DEEP NEURAL NETWORKS, PATTERN RECOGNITION, IDENTIFICATION, ON-BOARD, MULTI-ZONE.

The object of research is processes of intelligent control of mobile robotic platforms.

The subject of research is the implementation of a multi-zone on-board computer vision system.

The purpose of the qualification work is to study the usage of computer vision methods in the tasks of identifying explosive objects.

Research methods – analysis of existing approaches to computer vision problems, analysis of relevant sources, analysis of existing on-board multi-zone computer vision systems.

Field of application – robotic systems of humanitarian demining.

The existing computer vision systems in the field of detection of explosive objects were analysed. The architecture of the on-board multi-zone system has been developed. An on-board multi-zone system has been developed. Experiments were conducted to check the operation of the on-board multi-zone system.

The results of the qualification work were published in the materials of 2 international conferences.

ЗМІСТ

Вступ 8
1 Аналіз існуючих систем комп’ютерного зору в області виявлення вибухонебезпечних об’єктів 10
1.1 Аналіз сучасних систем комп’ютерного зору 10
1.2 Аналіз існуючих бортових систем комп’ютерного зору 27
1.3 Аналіз систем комп’ютерного зору для пошуку вибухонебезпечних об’єктів 32
1.4 Висновки 39
2 Розробка архітектури бортової багатозонової системи комп`ютерного зору 40
2.1 Основні характеристики бібліотеки OpenCV 40
2.2 Засоби навчання системи комп’ютерного зору на основні YOLO 44
2.3 Розробка архітектури системи комп`ютерного зору 60
2.4 Висновки 62
3 Розробка бортової багатозонової системи комп`ютерного зору 63
3.1 Опис використаних технологій 63
3.2 Апаратна реалізація 65
3.3 Програмна реалізація 66
3.4 Експерименти з перевірки роботи розробленої системи 74
3.5 Висновки 79
4 Охорона праці 80
4.1 Загальні положення 80
4.2 Вимоги безпеки перед початком роботи 81

4.3 Вимоги безпеки під час виконання роботи 81
4.4 Вимоги до безпеки в аварійних ситуаціях 82
Висновки 83
Перелік джерел посилань 84
Додаток А Презентація 88

ВСТУП

В наші дні людина намагається автоматизувати як змога більше процесів при виконанні різного роду робіт. Це зумовлено тим, що є робота, яка потребує дуже високої точності або взагалі є небезпечною.

Існує дуже велика кількість систем, які допомагають автоматизувати процеси. Кожна система вирішує різні задачі. Деякі системи використовують у виробництві для автоматизації виробничих процесів, тим самим зменшуючи кількість людей, які приймають активну участь у виробничих процесах. Як наслідок зменшується вірогідність травмування людини. Інші системи автоматизації можуть врятувати життя людини сповістивши про небезпеку.

Системи комп`ютерного зору наразі набувають дуже великої популярності. Вони допомагають автоматизувати різні процеси. Наприклад, система виявлення браку продукції на виробництві. Також системи комп`ютерного зору використовуються у різних сферах, а саме: медицина, виробництво, громадська безпека, тощо.

Наразі в Україні триває війна і після її завершення буде велика кількість снарядів, що не розірвалися. Процес їх пошуку та розмінування дуже небезпечний і саме тому цей процес необхідно автоматизувати. Для того, щоб автоматизувати цей процес можна використовувати бортову багатозонову систему комп’ютерного зору із функціями розпізнавання та ідентифікації вибухонебезпечних об’єктів. Таким чином людина буде дистанційно виконувати пошук підозрілих об’єктів. Це може зберегти багато життів [1]. Саме тому тематика роботи, пов’язана з розробкою засобів комп’ютерного зору, орієнтованих на виявлення та ідентифікацію вибухонебезпечних предметів є актуальним завдання.

Метою кваліфікаційної роботи є розробка бортової багатозонової системи комп'ютерного зору із функціями розпізнавання та ідентифікації.

Об'єкт дослідження – процеси інтелектуального керування мобільними робототехнічними платформами.

Предмет дослідження – реалізація багатозонової бортової системи комп'ютерного зору.

Для досягнення мети планується розв’язати наступні завдання:

– провести аналіз існуючих системи комп'ютерного зору в області виявлення вибухонебезпечних об’єктів;

– розробити архітектуру бортової багатозонової системи;

– розробити бортову багатозонову систему;

– перевірити достовірність роботи бортової багатозонової системи;

– розглянути основні положення з охорони праці для інженера-програміста.

Результати кваліфікаційної роботи апробовані у 2-х міжнародних конференціях.

Кваліфікаційна робота оформлена згідно з вимогами ДСТУ 3008:2015 [2], а також з рекомендаціями з підготовки і оформлення кваліфікаційної роботи здобувачами другого (магістерського) рівня вищої освіти [3].

1 АНАЛІЗ ІСНУЮЧИХ СИСТЕМ КОМП’ЮТЕРНОГО ЗОРУ В ОБЛАСТІ ВИЯВЛЕННЯ ВИБУХОНЕБЕЗПЕЧНИХ ОБ’ЄКТІВ

1.1 Аналіз сучасних систем комп’ютерного зору

Комп’ютерний зір – це галузь штучного інтелекту, пов’язана з аналізом, класифікацією і розпізнаванням зображень і відео [4].

До завдань комп’ютерного зору можна віднести:

– розпізнавання – одне з базових та першорядних завдань у обробці зображень, комп'ютерному та машинному зору. Воно допомагає класифікувати і ідентифікувати об'єкти, що характеризуються певним набором властивостей та ознак;

– відновлення зображень – це видалення шуму з використанням різних методів, наприклад, розмиття за допомогою фільтрів на основі машинного навчання (шум датчика, розмитість рухомого об'єкта і т. д.);

– аналіз руху – завдання використовує комп'ютерний зір для оцінки швидкості руху об'єктів у відео. Також застосовується для оцінки рухів, у яких послідовність відеоданих обробляється для знаходження швидкості кожної точки зображення чи 3D сцени;

– відновлення чи реконструкція сцени – допомагає відтворити тривимірну модель зображення або сцени, що вводиться за допомогою зображень чи відео. Найчастіше моделлю служить набір точок тривимірного простору;

– обробка та аналіз зображення – завдання зосереджено на перетворення одного 2D-зображення в інше. Реалізується за допомогою піксельних операцій, таких як підвищення контрастності або поворот зображення;

– високорівнева обробка – невеликий набір даних. Вона використовує різні методи для вилучення інформації з сигналів в цілому, наприклад, набір точок або

ділянка зображення, в якому імовірно знаходиться певний об'єкт, що цікавить частина даних.

Процес пошуку об’єктів на зображенні складається з двох етапів, а саме, розпізнавання та ідентифікація об’єктів сцени.

Розпізнавання – процес розмітки сцени, що представляє собою проекцію тривимірного робочого простору на площину об'єктива, реєструючого пристрою – цифрову камеру або ультразвукової модуль [5].

Ідентифікації об'єктів – процес тісно пов'язаний з розпізнаванням, що полягає в об'єднанні всієї отриманої інформації від процесу розпізнавання в єдине ціле з метою класифікувати об'єкт [5].

Загальна послідовність дій при розпізнаванні та ідентифікації виглядає так [6]:

– попередня обробка зображення – згладжування, фільтрація перешкод, підвищення контрасту;

– бінаризація зображення і виділення контурів об’єктів;

– початкова фільтрація контурів по периметру, площі, коефіцієнту форми;

– приведення контурів до єдиної довжині, згладжування;

– перебір всіх знайдених контурів, пошук шаблону, максимально схожого на даний контур.

Для розпізнавання об’єктів на зображенні застосовують дві стратегії: моделювання фону і моделювання об’єкта. Вибір стратегії залежить від умов отримання зображення. Моделювання фону застосовне тільки для «ідеальних» умов зйомки. Моделювання об’єкта – більш загальний підхід. До моделювання об’єкта для пошуку об’єкта на зображенні можна застосовувати різні підходи. Вибір конкретного метода залежить від багатьох умов. Найпростішими методами виділення об’єкта на зображенні є кольорові фільтри. Такі методи застосовуються, якщо об’єкт суттєво виділяється на фоні. Виділення країв та контурний аналіз будуть корисними у випадках, якщо об’єкт досить складний, але добре виділяється.

Це дає змогу перейти від роботи з зображенням до роботи з об’єктами на цьому зображенні. Далі можна перевірити наявність на зображенні певних геометричних форм. Метод співставлення зі шаблоном полягає у пошуку на зображенні ділянок, які співпадають з зображенням шуканого об’єкта. Якщо зображення об’єкта повернуте чи масштабоване відносно шаблону, то цей метод неефективний. Для таких випадків краще підійдуть методи засновані на так званих особливих точках. Особливі точки – це особливі характеристики об’єкта. Вони дозволяють зіставити об’єкт сам з собою або зі схожими класами об’єктів. Існує кілька способів виділяти особливі точки. Деякі способи виділяють особливі точки на сусідніх кадрах, деякі - через великі проміжки часу та при різному освітленні, деякі дозволяють знайти особливі точки, навіть при повороті зображення. Найскладнішими випадками розпізнавання є пошук об’єктів певного класу. В таких випадках задачу виявлення і розпізнавання можна вирішити за допомогою побудови класифікатора на основі машинного навчання, який складається з метода виділення особливостей та власне класифікатора. Для того, щоб навчити алгоритми ідентифікувати об’єкти необхідно використовувати набори розмічених даних – датасети. Вони містять відмінності одного об’єкту від іншого. Слід зауважити, що точність розпізнавання та ідентифікації об’єктів залежить від розміру датасету. Чим більше об’єм датасету, тим точніше працює система комп’ютерного зору. Методи виділення особливостей залежать від поставленої задачі. Для одного класу задач це може бути навчання на позитивних і негативних датасетах, які містять у собі велику кількість зображень, для інших – це виділення кластерів дескрипторів особливих точок і створення, так би мовити, словника дескрипторів [7].

Існує велика кількість підходів для розпізнавання об'єктів. У наші часи методи машинного навчання та глибокого навчання набули популярності в якості підходів для розпізнавання об'єктів.

Технології глибокого навчання активно використовуються для вирішення задач із розпізнавання об'єктів. Існують різні архітектури глибокого навчання до

яких можна віднести: згорткові мережі, генеративні змагальні мережі. Наприклад згорткові нейронні мережі. Згорткова нейронна мережа (англ. Convolutional neural network, CNN) – спеціальна архітектура штучних нейронних мереж, націлена на ефективне розпізнавання образів, входить до складу технологій глибокого навчання [8]. Завдяки високій швидкості розпізнавання нейронних мереж цього типу було покращено більшість завдань комп'ютерного зору, як уже існуючих, так і нових. Для того, щоб навчити згорткову мережу розпізнавати об'єкти необхідно використовувати набори позначених зображень. Мережі цього типу будуть аналізувати набори зображень і вивчати особливості різних об’єктів за якими потім і буде їх розпізнавати.

Згорткові мережі часто застосовують для вирішення задач класифікації зображень. У цих задачах кожне зображення містить лише один об’єкт і задача полягає в тому, щоб ідентифікувати його.

Для того, щоб розпізнавати кілька об’єктів та їх розташування на зображенні створюються системи виявлення об’єктів, такі як YOLO (you only look once або ви дивитеся лише один раз), SSD (single-shot detector або одиночний детектор) і R-CNN (Region-based Convolutional Neural Network або регіональна згорткова нейронна мережа), які не тільки класифікують зображення, але й можуть розпізнавати та ідентифікувати кілька об’єктів на зображеннях. Такі системи глибокого навчання можуть переглядати зображення, розбивати його на більш дрібні області та позначати кожну область класом, щоб змінну кількість об’єктів у даному зображенні можна було локалізувати та позначити.

Ідея згорткових нейронних мереж полягає в чергуванні згорткових шарів і субдискретизуючих шарів. Структура мережі – односпрямована (без зворотних зв’язків), багатошарова. Модель згорткової мережі складається з трьох типів шарів: згорткові шари, субдискретизуючі верстви і прошарки «звичайної» нейронної мережі – персептрона.

На рисунку 1.1 наведена структура згорткової нейронної мережі.

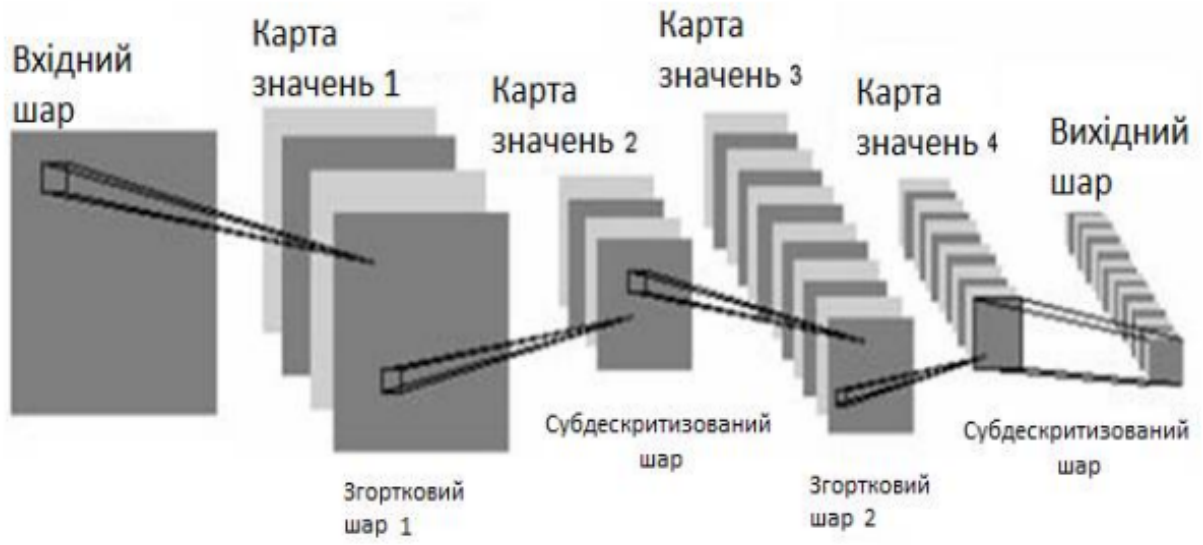

Рисунок 1.1 – Структура згорткової нейронної мережі

Робота згорткової нейронної мережі забезпечують два основні елементи, а саме:

– фільтри (визначники ознак);

– карти ознак.

Фільтр – це невелика матриця, що представляє ознаку, яку необхідно знайти на вихідному зображенні. За допомогою верхнього фільтра визначаються частини вихідного зображення з вертикальними лініями, нижній фільтр служить для визначення частин зображення з горизонтальними лініями.

Безпосередньо процес визначення заснований на операції згортки фільтром оригінального зображення. Результати згортки, які визначають місце розташування ознак вихідного зображення, називаються картами ознак.

Мета процесу згортки – зменшити розмірність карти ознак до такої міри, щоб з повним набором ознак могла працювати мережа прямого поширення (в більшості випадків багатошаровий персептрон) [9].

До недоліків згорткових мереж можна віднести [9]:

– доволі висока складність архітектури;

– повнозв’язність;

– фіксована площа вікна шару згортки.

GAN (generative adversarial network або генеративна змагальна мережа) – це тип архітектури нейронної мережі для генеративного моделювання. Існує багато застосувань GAN, пов'язаних із створенням нових моделей [10]:

– створення нових анімаційних персон;

– створення нових логотипів;

– генерація нових покемонів;

– створення моделей нового одягу;

– створення зразків нових меблів.

StackGAN є одним із варіантів архітектури GAN, який може використовувати текстовий опис об’єкта для створення зображення об’єкта з високою роздільною здатністю, що відповідає цьому опису. Це не просто пошук зображень у базі даних. Ці зображення ніколи раніше не існували і є повністю уявними [11].

GAN базуються на ідеї змагального навчання. Архітектура GAN в основному складається з двох нейронних мереж, які конкурують одна з одною:

– генератор намагається перетворити випадковий шум у спостереження, які виглядають так, ніби вони були взяті з вихідного набору даних;

– дискримінатор намагається передбачити, чи це зображення з оригінального набору даних, чи є однією з підробок генератора.

GAN виконує такі кроки:

– генератор приймає випадкові числа та повертає зображення;

– генероване зображення подається в дискримінатор разом із реальним зразком зображення;

– дискримінатор порівнює справжні зображення і підроблені та повертає ймовірності числа від 0 до 1, де 1 означає прогноз автентичності та 0 означає підробку.

На рисунку 1.2 наведена схема генеративної змагальної мережі.

Рисунок 1.2 – Схема генеративної змагальної мережі

Існує декілька основних технологій розпізнавання об'єктів на зображенні, а саме: R-CNN, SSD, та YOLO.

Як правило, процес виявлення об’єктів складається з чотирьох компонентів:

– регіональна пропозиція – алгоритм або модель глибокого навчання, яка використовується для генерації цікавих регіонів для подальшої обробки системою. Це регіони, які можуть містити об’єкт. Результатом є велика кількість обмежувальних рамок, кожна з яких має оцінку об’єктності. Рамки з великими балами об’єктності потім передаються вздовж мережевих шарів для подальшої обробки;

– виділення функцій і передбачення мережі – візуальні функції виділяються для кожної з обмежувальних рамок. Їх оцінюють, і за візуальними ознаками (наприклад, компонентом класифікації об’єктів) визначають, чи є об’єкти в пропозиціях і які саме;

– немаксимальне придушення – на цьому етапі модель, знайшла кілька обмежувальних рамок для одного об’єкта. Немаксимальне придушення допомагає уникнути повторного виявлення одного й того самого екземпляра, об’єднуючи рамки, що перекриваються, в одну обмежувальну рамку для кожного об’єкта;

– метрики оцінки. Подібно до метрик точності, прецизійності та запам’ятовування в задачах класифікації зображень, системи виявлення об’єктів мають власні метрики для оцінки ефективності виявлення такі як середня точність, крива точності-відкликання і перетин через об’єднання.

Для оцінки продуктивності детектора об’єктів використовують два основних показника оцінки: кількість кадрів за секунду та середню точність.

Найпоширенішим показником, який використовується для вимірювання швидкості виявлення, є кількість кадрів за секунду. Наприклад, R-CNN працює зі швидкістю лише 7 кадрів на секунду, тоді як SSD працює зі швидкістю 59 кадрів на секунду.

Найпоширенішим показником оцінки, який використовується в задачах розпізнавання об’єктів, є середня точність. Це відсотки від 0 до 100, і вищі значення зазвичай кращі, але його значення відрізняються від показника точності, який використовується в класифікації.

Цей показник оцінює перекриття між двома обмежувальними рамками: обмежувальною рамкою істинності на землі і прогнозованою обмежувальною рамкою. Застосовуючи це, ми можемо визначити, чи є виявлення дійсним (справжній позитивний) чи ні (хибний позитивний).

Підхід SSD базується на згортковій мережі прямого зв’язку, яка створює колекцію обмежувальних прямокутників фіксованого розміру та оцінює присутність екземплярів класу об’єктів у цих коробках, після чого виконується крок немаксимального придушення для створення остаточних виявлень.

Архітектура моделі SSD складається з трьох основних частин:

– базова мережа для отримання карт об’єктів – стандартна попередньо навчена мережа, яка використовується для високоякісної класифікації зображень, яка скорочується перед усіма шарами класифікації;

– багато масштабовані функціональні шари – серія згорткових фільтрів додається після базової мережі. Розміри цих шарів поступово зменшуються, що дозволяє прогнозувати виявлення в різних масштабах;

– немаксимальне придушення – використовується для усунення блоків, що перекриваються і збереження лише одного блоку для кожного виявленого об’єкта.

Сімейство YOLO – це серія скрізних моделей глибокого навчання, розроблених для швидкого виявлення об’єктів, і це була одна з перших спроб створити швидкий детектор об’єктів у реальному часі. Це один із найшвидших алгоритмів виявлення об’єктів. Хоча точність моделей близька, але не така добра, як в R-CNN, вони популярні для виявлення об’єктів через їхню швидкість виявлення, яка часто демонструється у відео в реальному часі або вхідних даних камери.

У YOLO застосовується інший підхід, ніж в попередніх мережах. YOLO не проходить етап регіональної пропозиції, як R-CNN. Натомість він передбачає лише обмежену кількість обмежувальних рамок, розбиваючи вхідні дані на сітку комірок. Кожна клітинка безпосередньо передбачає обмежувальну рамку та класифікацію об’єкту. Результат є велика кількість обмежувальних рамок-кандидатів, які об’єднуються в остаточний прогноз.

Більшість додатків для глибокого навчання використовують підхід передачі навчання - процес, який передбачає точне налаштування перевіреної моделі. Процес починається з існуючої мережі, наприклад VGGNet, AlexNet або GoogleNet, і до неї подаються нові дані, що містять невідомі раніше класи. Цей метод менш трудомісткий і може забезпечити швидший результат, оскільки модель вже пройшла навчання на тисячах або мільйонах зображень.

Існує два підходи до розпізнавання об'єктів за допомогою глибокого навчання.

Навчання моделі з нуля: для того, щоб навчити глибоку мережу з нуля, треба зібрати дуже великий набір міток даних та спроектувати мережеву архітектуру, яка вивчить функції та побудувати модель. Результати можуть бути вражаючими, але такий підхід вимагає великої кількості даних для навчання мережі, і потрібно налаштувати шари та ваги нейронної мережі.

Щоб виконати розпізнавання об'єктів за допомогою стандартного підходу до машинного навчання, використовуються колекції зображень або відео та вибираються відповідні функції для кожного зображення. Наприклад, алгоритм вилучення можливостей може витягнути крайові або кутові функції, які можна використовувати для розмежування класів у даних.

Ці функції додаються до моделі машинного навчання, яка розділить ці особливості на їх окремі категорії, а потім використовувати цю інформацію при аналізі та класифікації нових об'єктів.

Скоріш за все, необхідно буде використовувати різноманітні алгоритми машинного навчання та методи вилучення функцій, які пропонують безліч комбінацій для створення точної моделі розпізнавання об'єктів.

Використання машинного навчання для розпізнавання об'єктів пропонує гнучкість вибору найкращого поєднання функцій та класифікаторів для навчання. За допомогою цього можна досягти точних результатів з мінімальними даними.

Визначення найкращого підходу до розпізнавання об'єктів залежить від програми та проблеми, яку необхідно вирішити. У багатьох випадках машинне навчання може бути ефективною технікою, особливо якщо відомо, які особливості чи характеристики зображення найкраще використовувати для диференціації класів об’єктів.

Основним моментом, який слід пам’ятати при виборі між машинним та глибоким навчанням це можливість використовувати потужний комп’ютер та

великий обсяг зображень для тренування. Якщо такої можливості нема, то найкращим вибором може стати підхід до машинного навчання. Технології глибокого навчання, як правило, краще працюють із більшою кількістю зображень, а графічний процесор допомагає скоротити час, необхідний для тренування моделі [12].

У загальному випадку система комп’ютерного зору складається з камери та обчислювального блоку, який обробляє зображення з камери. На рисунку 1.3 наведена загальна структура системи комп`ютерного зору.

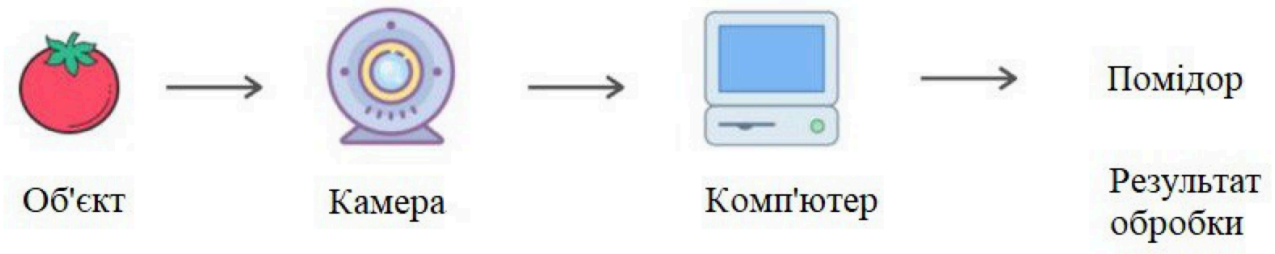

Рисунок 1.3 – Загальна структура системи комп`ютерного зору

Існує велика кількість систем комп`ютерного зору. Прикладами таких систем можуть бути:

– системи призначені для керування процесами;

– системи призначені для відеоспостереження;

– системи призначені для обробки та систематизації інформації;

– системи для моделювання об’єктів;

– системи моделювання антропомашинної взаємодії.

Системи комп`ютерного зору використовуються для вирішення різних задач.

Безпілотний автомобіль є гарним прикладом використання таких систем. Безпілотні автомобілі оснащені одною чи декількома камерами з кожної сторони автівки. Інформація про зображення, отримана від бортових камер транспортного

засобу, передається до блоку обробки зображень, де комп'ютер витягує інформацію про навколишнє середовище автівки та надсилає отриману інформацію до блоку керування транспортним засобом, який спираючись на цю інформацію керує автівкою. Прикладом такої автівки може бути Tesla. На рисунку 1.4 наведений процес керування автівкою за допомогою автопілота.

Рисунок 1.4 – Процес керування автівкою за допомогою автопілота

Також для автівок існують допоміжні системи. Прикладом такої системи може бути система утримання автівки у полосі. Принцип роботи цієї системи схожий на систему автопілоту. Автівка обладнана камерами, а зображення з них обробляє вже обчислювальний блок. Розпізнавання полоси виконується у декілька етапів, а саме:

а) отримується зображення спереду автівки;

б) зображення перетворюється в градації сірого;

в) робиться розмиття зображення за Гаусом;

г) робиться пошук країв за допомогою функції Кенні;

ґ) робиться сегментація зображення;

д) виконується пошук ліній на зображенні;

е) оригінал зображення та зображення із результатом пошуку полоси на дорозі об`єднуються.

На рисунку 1.5 наведені етапі розпізнавання полоси на дорозі системою утримання автівки у полосі.

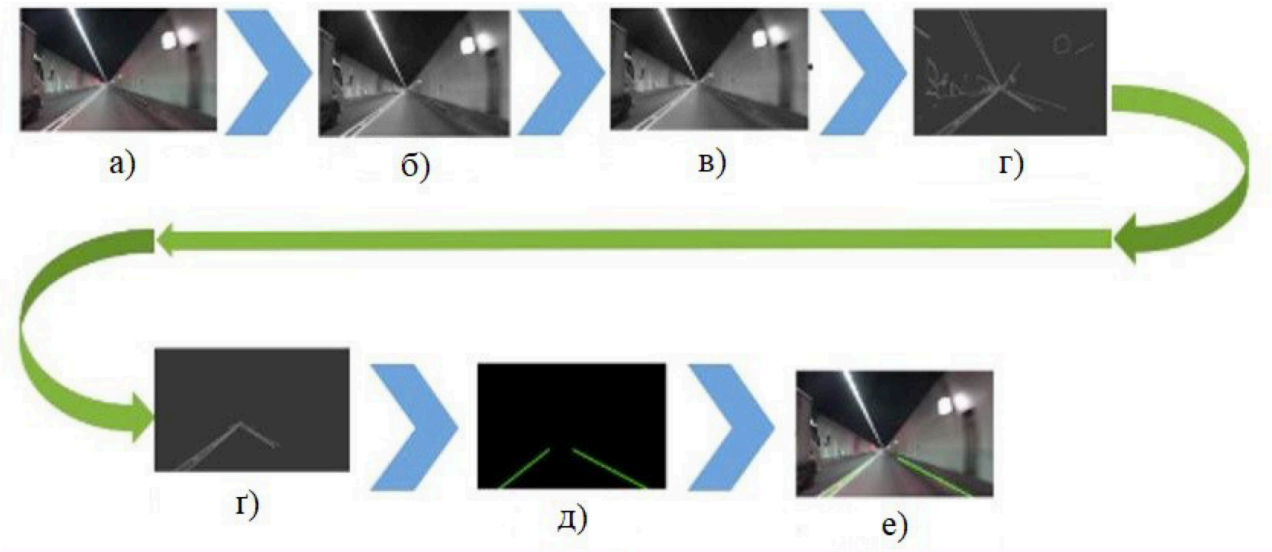

Рисунок 1.5 – Етапи розпізнавання полоси на дорозі системою утримання автівки у полосі

В промисловій галузі методи комп’ютерного зору використовуються для автоматизації процесів на виробництві. На будь-якому виробництві існує процес перевірки якості продукції. Для того, щоб автоматизувати цей процес використовуються системі візуального огляду на основі штучного інтелекту. Візуальна перевірка на основі штучного інтелекту передбачає використання машинного навчання для автоматичної перевірки якості продукції шляхом аналізу зображень і відеоданих. Технології штучного інтелекту та комп’ютерного зору дозволяють виробникам автоматизувати виявлення дефектів продукції, заощаджуючи час і гроші, покращуючи контроль якості. Прикладом може бути сервіс Amazon SageMaker Edge. Amazon SageMaker Edge – це сервіс машинного навчання, який допомагає виявляти дефекти продукту за допомогою комп’ютерного бачення для автоматизації процесу перевірки якості на виробничих лініях, не потребуючи досвіду у роботі із машинним навчанням. [13]. На рисунку 1.6 наведено результат роботи цієї системи.

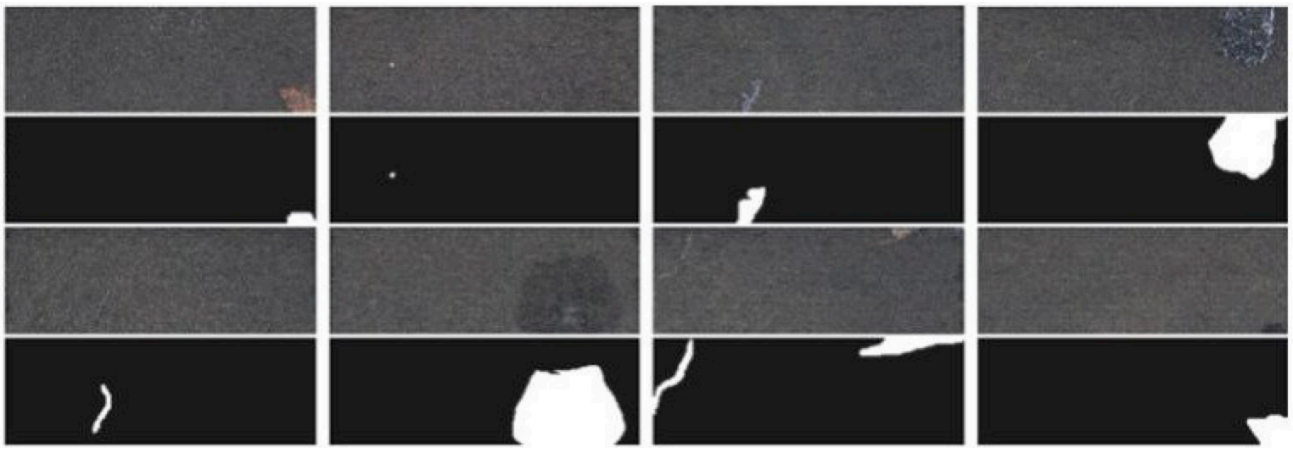

Рисунок 1.6 – Результат пошуку дефектів за допомогою Amazon SageMaker Edge

У ритейлі системи комп’ютерного зору використовуються для перевірки активності клієнтів. Крім цього системи комп’ютерного зору на основі штучного інтелекту можуть ідентифікувати незадоволених покупців або вітати постійних клієнтів. Це також може сприяти програмі утримання клієнтів. Завдання систем комп’ютерного зору в роздрібній торгівлі також включають оцінку стратегій розміщення товарів і відстеження запасів. Такі гіганти електронної комерції, як Amazon, також масово впроваджують технологію візуального пошуку для відображення добре оптимізованого вмісту.

У медицині класифікація зображень і виявлення шаблонів лежать в основі медичних програмних систем. Вони допомагають лікарям у діагностиці небезпечних станів. Значний прорив у комп’ютерному зорі також дозволяє використовувати дані медичних зображень. За останні кілька років спостерігається зростання застосування методів комп'ютерного зору до статичних медичних зображень. Прикладом такої системи може бути InnerEye від Microsoft. InnerEye — це дослідницький проект від Microsoft Health Futures, який використовує сучасну технологію машинного навчання для створення інноваційних інструментів для автоматичного кількісного аналізу тривимірних медичних зображень [14]. InnerEye

використовує тривимірні радіологічні зображення, доступні для всіх медичних установ. На рисунку 1.7 наведено результат роботи InnerEye.

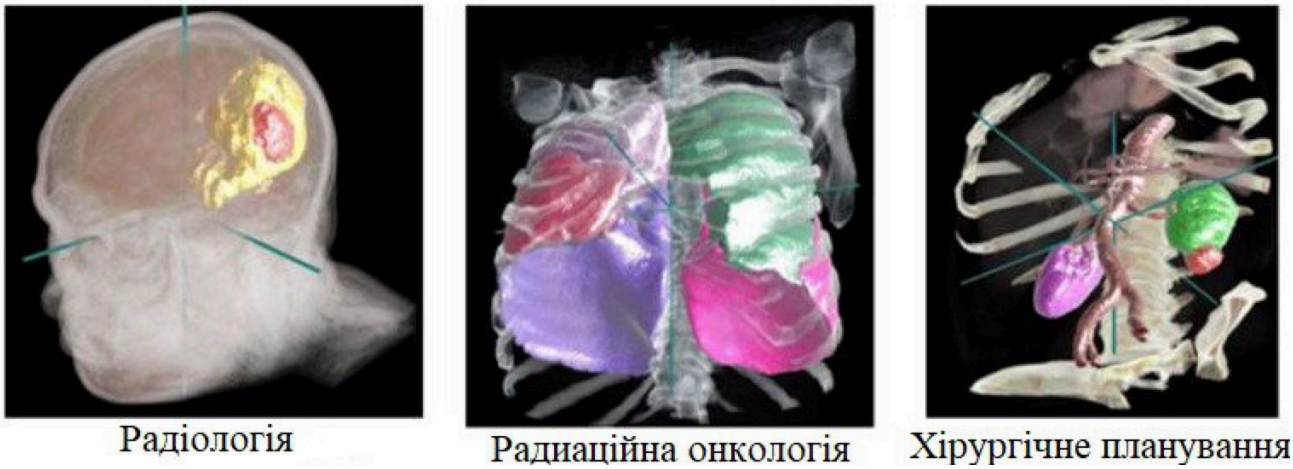

Рисунок 1.7 – Результат роботи InnerEye

Також комп’ютерний зір та програми глибокого навчання можуть використовуватися для виявлення пухлин мозку. Медичні працівники можуть використовувати програми комп’ютерного зору, щоб зробити процес виявлення менш трудомістким і виснажливим. У сфері охорони здоров’я методи комп’ютерного зору, такі як згорткові нейронні мережі Mask-R, можуть допомогти виявити пухлини мозку, тим самим значно зменшуючи ймовірність людської помилки.

Пандемія Covid-19 стала серйозною проблемою для сфери охорони здоров’я в усьому світі. Оскільки країни в усьому світі борються з цією хворобою, комп’ютерний зір може значно сприяти вирішенню цієї проблеми. Завдяки швидкому технологічному прогресу програми комп’ютерного зору можуть допомогти в діагностиці, контролі, лікуванні та профілактиці Covid-19. Цифрові рентгенівські зображення грудної клітки в поєднанні з програмами комп’ютерного зору, такими як COVID-Net, можуть легко виявляти захворювання у пацієнтів.

Прототип програми, розроблений Darwin AI, показав результати з точністю близько 92,4% у діагностиці ковіду.

У галузі громадської безпеки яскравим прикладом використання методів комп'ютерного зору є виявлення та розпізнавання обличчя. Розпізнавання обличь з використанням методів комп’ютерного зору може виявляти та запобігати злочинам. Алгоритм комп’ютерного зору допомагає зафіксувати обличчя після чого зображення надсилається в систему для подальшого аналізу. Поєднання алгоритмів аналізу та розпізнавання – це те, що фактично становить систему розпізнавання обличь.

Ще існують технології розпізнавання радужної оболонки ока. Ці технології використовують комерційні і урядові установи для різних цілей: від систем контролю доступу до організації робочого часу. Лабораторія ASSA ABLOY Future Lab займається дослідженням нової області застосування цієї біометричної технології – управління ідентифікаційними даними, а також іншими варіантами застосувань з метою забезпечення безпеки. Крім технологій розпізнавання радужної оболонки ока є ще технології розпізнавання сітківки ока. Різниця між цими технологіями полягає в тому, що при розпізнаванні райдужної оболонки по суті фіксується малюнок текстури райдужки, при скануванні сітківки ока захоплюється зображення сітки кровоносних судин всередині ока. На відміну від сітківки, райдужну оболонку можна побачити неозброєним поглядом, тому набагато простіше отримати якісне зображення райдужної оболонки. Сітківка, в свою чергу, складається з фоторецепторів клітин, розташованих на задній стінці ока, і її не можна побачити.

SmartGate ще один приклад використання системи розпізнавання обличь. SmartGate – це автоматизована система прикордонного контролю, яку використовують прикордонні сили Австралії та митна служба Нової Зеландії. Ця система дозволяє збільшити швидкість проходження імміграційного контролю і підвищує безпеку подорожей. Перевірка відбувається наступним чином. Система

розпізнає обличчя людини і потім порівнює його із тим, що зберігається у чіпі біометричного паспорту.

Системи розпізнавання обличчя також можуть використовуватися для підтвердження особи. Такі системи використовуються для розблокування пристрою або як спосіб підтвердження фінансових операцій. Прикладом такої системи може бути Face ID від Apple. Ця система проектує інфрачервоні точки на обличчя користувача, після чого створюється шаблон обличчя. Цей шаблон відправляється на обробку центральному процесору пристрою для подальшого порівняння отриманого шаблону і шаблону, який зберігається в пристрої.

Методи комп'ютерного зору в поєднанні із глибоким навчанням використовується для синтезу зображення людини. Ця методика отримала назву deepfake. Ця методика полягає у поєднанні та накладенні одних зображень та відео на інші зображення або відеоролики. Існує кілька методів створення deepfake, але найпопулярніший є використання нейронних мереж із залученням автокодерів, які використовують техніку зміни обличчя. Програми, які спеціалізуються на deepfake можуть використовуватися для різних цілей. Наприклад у процесі створення фільмів. До прикладів deepfake програм можна віднести Zao, FaceApp, DeepFace Lab.

У військовій галузі активно використовують системи комп'ютерного зору для виконання різноманітних задач. Наприклад виявлення транспортних засобів, ворожих солдатів. Крім цього методи комп'ютерного зору використовуються в системах наведення ракет. Така система наведення може працювати наступним чином. Ракета досягає певної області без використання системи комп'ютерного зору. Після перетину попередньо заданої області активується система наведення ракети, яка використовуючи методи комп'ютерного зору виконує наведення на ціль.

Ще одним прикладом застосуванням систем комп'ютерного зору може бути безпілотний літальний апарат. Безпілотні літальні апарати, які оснащені системою комп'ютерного зору можуть виконувати різні задачі. Літальний апарат може

використовувати камери для розвідки. Іншим варіантом використання безпілотного літального апарату може бути пошук та знищення ворога. Такі літальні апарати працюють за принципом ракет, який наведено вище.

Також системи комп'ютерного зору можуть виступати у ролі допоміжної системи. Прикладом такої системи може бути шолом пілота військового літака. Такі шоломи допомагають легше та точніше прицілюватися або сповіщують пілота про небезпеку. Прикладом такої системи може бути X-Sight. Система X-Sight забезпечує пілотам чітке, кольорове широке поле зору в режимі реального часу в будь-який час і за будь-яких погодних умов. З радіусом дії 2000 м система включає функції нічного бачення, які може вмикати та вимикати пілот, а також ідентифікацію цілей, що переміщуються, і індикацію цілей, що рухаються, щоб попередити пілотів про можливі наближення загрози [15].

1.2 Аналіз існуючих бортових систем комп'ютерного зору

Depth Sensing Kit flexx2 – це 3D-камера глибини з USB підключенням для розробки, досліджень і масового виробництва. Вона оснащена датчиком зображення Time-of-Flight із роздільною здатністю тридцять вісім тисяч пікселів, гнучкими робочими діапазонами, гнучкою частотою кадрів, а також зниженим глибинним шумом та інфрачервоним нічним баченням.

Ця камера має наступні характеристики:

а) розмір 71,9 мм х 19,2 мм х 10,6 мм;

б) роздільна здатність 224 х 172;

в) датчик ToF IRS2381C;

г) освітлення 940 нм 1 Вт VCSEL (LC1);

ґ) частота кадрів до 60 кадрів в секунду (3D кадри);

д) поле зору 56 х 44 FOV;

е) діапазон 0,1 – 4 м;

є) інтерфейс USB3 Type-C;

ж) точність <= 1% відстані (0,5–4 м @ 5 кадрів/с) <= 2 % відстані (0,1–1 м @ 45 кадрів/с);

з) Software Royale SDK на основі C/C++ підтримує Matlab, OpenCV, ROS 1, ROS 2.

На рисунку 1.8 наведено зовнішній вигляд камери.

Рисунок 1.8 – Зовнішній вигляд камери Depth Sensing Kit flexx2

Intel RealSense Depth Camera D405 – стереокамера короткого радіусу дії, що забезпечує субміліметрову точність для потреб комп'ютерного бачення на близькій відстані. D405 працює в ідеальному діапазоні від 7 см до 50 см з мінімальним виявленням об'єкта до 0,1 мм на відстані 7 см. Камеру можна використовувати як для автоматизації виробництва так і для медицини.

Камера має наступні технічні характеристики:

а) захищена від вологи та пилу;

б) технологія датчика зображення Global Shutter;

в) робочий діапазон від 7 см до 50 см;

г) технологія глибини Stereoscopic;

ґ) мінімальна дистанція глибини приблизно 7 см @ 480p;

д) точність глибини менше 2% при 50 см;

е) поле зору глибини 87° × 58°;

є) вихідна роздільна здатність потоку глибини до 1280 × 800;

ж) частота кадрів вихідного потоку глибини до 90 fps;

з) роздільна здатність кадру RGB 1280 × 800;

и) частота кадрів RGB 90 fps;

і) сенсорна технологія RGB Lef Imager RGB with ISP;

ї) механізм кріплення одна точка кріплення з різьбою 1/4-20 UNC або дві точки кріплення з різьбою;

й) датчик FOV RGB 87° × 58°;

к) роздільна здатність датчика RGB до 1280 x 800;

л) модуль камери Intel RealSense Module D401;

м) плата відеопроцесора Intel RealSense Vision Processor D4 Board v4;

н) форм-фактор Camera Peripheral;

о) габарити 42 мм × 42 мм × 23 мм;

п) роз'єми USB C та USB 3.1.

На рисунку 1.9 наведено зовнішній вигляд камери.

Рисунок 1.9 – Зовнішній вигляд камери

Intel RealSense Depth Camera D435 призначена для вимірювання глибини зображень шляхом проставлення кожному пікселю зображення свого значення відстані від камери до об'єкта зйомки. Камера працює за технологією стереоглубіни, тобто визначає відстань на основі відмінності в зображеннях з двох камер-приймачів, що працюють в інфрачервоному діапазоні і спрямованих в одному напрямку. Володіючи камерою з широким кутом огляду, даний пристрій є ідеальним варіантом для додатків автоматизованої навігації і розпізнавання об'єктів. D435 є частиною лінійки D400 камер Intel RealSense, яка використовує новітнє апаратне і програмне забезпечення Intel для визначення глибини (процесори машинного зору, модулі під ключ, камери, SDK і бібліотеки комп'ютерного зору).

Для роботи з камерою виробник надає Intel RealSense SDK 2.0 з можливістю самокалібрування камери, що дозволяє проводити калібрування камери менш ніж за 15 секунд без необхідності використання спеціалізованих додатків і апаратури [16].

Камера має наступні технічні характеристики:

а) захищена від вологи та пилу;

б) технологія сенсора зображення Global Shutter з розміром пікселя 3μm × 3μm;

в) максимальна дальність 10 м;

г) технологія визначення глибини Active IR Stereo;

ґ) мінімальна відстань 0,105 м;

д) кут огляду глибини зображення 86° × 57° (±3°);

е) вихідна роздільна здатність до 1280x720 90fps;

є) роздільна здатність RGB-сенсора 1920x1080;

ж) частота RGB-сенсора 30fps;

з) кут огляду RGB-сенсора 69,4° × 42,5° × 77° (±3°);

и) модуль камери Intel RealSense Module D430 та RGB Camera;

і) плата відео обробника Intel RealSense Vision Processor D4;

ї) інтерфейси USB 3.1;

й) габаритні розміри 90 мм x 25 мм x 25 мм;

к) механізми кріплення;

1) 2 отвори під M3 гвинт;

2) кріплення для трипода.

На рисунку 1.10 Наведена камера Intel RealSense D435.

Рисунок 1.10 – Intel RealSense Depth Camera D435

SOLOSHOT3 – камера, яка дозволяє записувати відео у високій якості та має вбудовані алгоритми відстеження об’єктів. Однією з її переваг є якісний зум (збільшення зображення у 65 разів не втрачаючи якості) та автоматичне відстежування заданого об’єкту у кадрі. Для того, щоб отримати дані з камери, використовується зручний інтерфейс, написаний на мові програмування Python. Проте, цілісної системи для роботи з даними з камер та відображення немає. Крім того, виробник не рекомендує використовувати цю камеру для зйомок в приміщеннях та місцях великого скупчення людей, що є суттєвим обмеженням в можливих сферах використання [17].

Камера має наступні технічні характеристики:

а) запис відео 4k @ 30 fps 1080p @ 30, 60, 120 fps 720p @ 240 fps;

б) формат відео mp4;

в) роздільна здатність 4000 на 3000;

г) формат фото jpeg;

ґ) оптичний зум 65х;

д) стабілізація матриці оптична;

е) фокусна відстань;

1) від 0,3 метрів до нескінченності (широкий кут);

2) від 6 метрів до нескінченності (телефото);

є) горизонтальне поле зору;

1) 72,6 (широкий кут);

2) 1,3 (телефото);

ж) вага 373 грами.

На рисунку 1.11 Наведена камера SOLOSHOT3.

Рисунок 1.11 – SOLOSHOT3

1.3 Аналіз систем комп’ютерного зору для пошуку вибухонебезпечних об’єктів

ANDROS – це клас дистанційно керованих роботів-знешкоджувачів, які використовуються американськими військовими, а також спецназом і

правоохоронними групами. Сімейство роботів ANDROS розроблено компанією REMOTEC, яка є дочірньою компанією оборонного гіганта Northrop Grumman. Доволі великою перевагою є те, що зображення з камер можна відправляти на обробку до віддалених серверів, які мають доступ до баз даних вибухонебезпечних предметів. Отже після розпізнавання вибухівки можна провести аналіз і визначити тип вибухонебезпечного пристрою та як його знешкодити.

Andros FX – це гусеничний робот, здатний протистояти широкому спектру загроз, включаючи саморобні вибухові пристрої, які перевозяться на транспортних засобах. На рисунку 1.12 наведений зовнішній вигляд роботу.

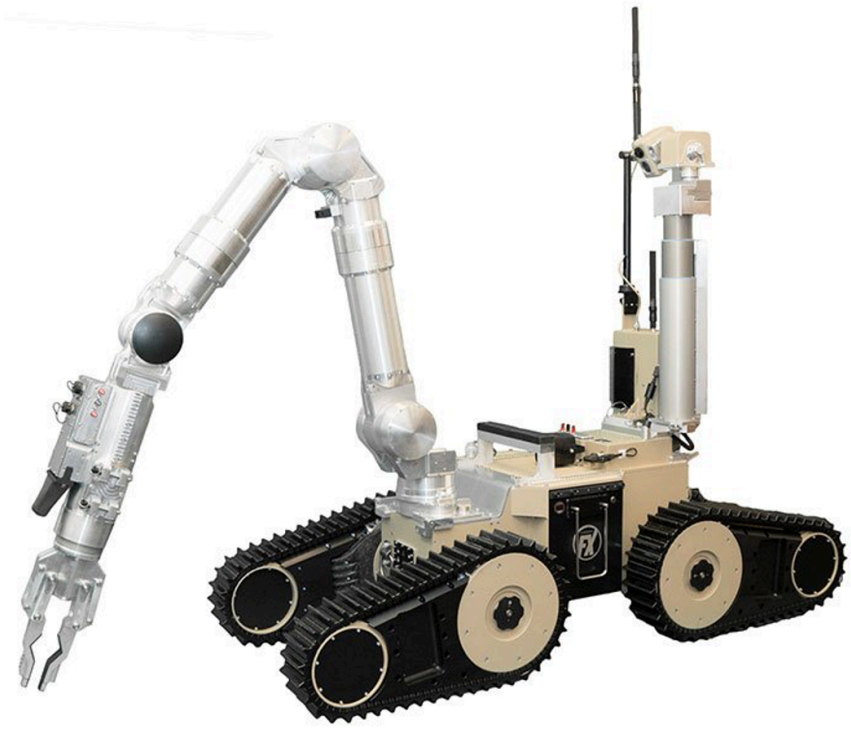

Рисунок 1.12 – Andros FX

Andros FX має наступні характеристики [18]:

а) камери;

1) спостереження;

– кольорова камера з можливістю перейти в режим інфрачервоного зображення;

- можливість повороту на 360°;
- нахил +/-90°;
- автоматичний або ручний фокус;
- автоматична або ручна діафрагма;
- світлодіодні ліхтарі з дистанційним перемиканням від білого світла до інфрачервоного;

2) на маніпуляторі;
   - фіксована кольорова камера з світлодіодним підсвічуванням із дистанційним перемиканням з білого світла на інфрачервоне;
3) спереду та позаду;
   - кольорова камера, яка має можливість перемикатися на чорно-біле зображення в умовах слабкого освітлення з фіксованим фокусом;
4) на захваті маніпулятора
   - чорно-біла з фіксованим фокусом;
5) додаткова теплова камера;
   - встановлюється в панорамну/нахильну щоглу або корпусі маніпулятора;
   - розмір зображення 320 на 240;
   - діапазон робочих температур від -40 °С до +80 °С;
6) додаткова інструментальна камера;
   - чорно-біла з фіксованим фокусом;
   - автоматична діафрагма;
   - додаткова лінза з лазерним покращенням;

б) програмне забезпечення;
1) можливість використовувати дев’ять заводських налаштувань та додати ще дев’ять власних налаштувань;
2) на екрані оператора можна відображати зображення з однієї або усіх чотирьох камер. Режим однієї камери пропонує три режими перегляду

камери «картинка в картинці». Картинки можна переміщувати та змінювати їх розмір;

3) положення нахилу робота та нахилу автомобіля відображаються на екрані чим допомагають забезпечити впевнені маневри на будь-якій місцевості.

Andros Titus невеликий, ефективний робот, що легко розгортається, представник сімейства Andros. Він надає розширені можливості для вирішення різних типів завдань для задоволення широкого кола потреб користувача.

Andros Titus забезпечує продуктивність, міцність та надійність, очікувані від більшої машини Andros при менших розмірах.

На рисунку 1.13 наведений зовнішній вигляд роботу.

Рисунок 1.13 – Andros Titus

Andros Titus має наступні характеристики [19]:

а) камери;

1) спостереження;

– кольорова камера із функцією нічного бачення;

- масштабування 216 до 1;
- можливість повороту на 360°;
- нахил на +/-90º;
- автоматичний або ручний фокус;
- автоматична або ручна діафрагма;

2) на маніпуляторі;
   - кольорова камера з фіксованим фокусом;
   - автоматична діафрагма;
   - вбудоване світлодіодне кільце білого світла;

3) спереду;
   - кольорова камера з фіксованим фокусом;
   - автоматична діафрагма;
   - вбудовані світлодіоди білого світла;

4) позаду;
   - кольорова камера з фіксованим фокусом;
   - автоматична діафрагма;

б) програмне забезпечення;

1) вікно перегляду;
   - «картинка в картинці»;
   - «плаваючі картинки»;
   - можна змінювати розмір вікон перегляду.

На рисунку 1.14 наведено зовнішній вигляд інтерфейсу програми для керування роботом Andros Titus.

Рисунок 1.14 – Інтерфейс програми для керування роботом Andros Titus

Andros F6B - найуніверсальніший і найпотужніший робот у своєму сегменті на ринку. Швидкість та маневреність роблять його привабливим для вирішення широкого кола завдань.

На рисунку 1.15 наведено зовнішній вигляд роботу.

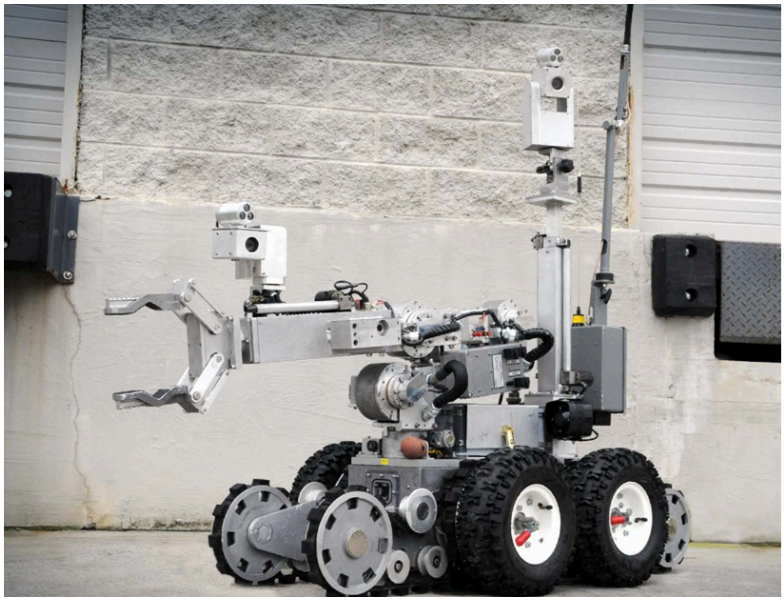

Рисунок 1.15 – Andros F6B

Andros F6B має наступні характеристики [20]:

а) камери;

1) спостереження;

– кольорова камера із функцією нічного бачення;
– масштабування 216 до 1;
– нахил 180º;
– автоматичний або ручний фокус;
– автоматична або ручна діафрагма;
– світлодіодні ліхтарі з дистанційним перемиканням з білого на інфрачервоне світло;
– моторизований розширювач камери;

2) на маніпуляторі;

– кольорова камера з автоматичним/ручним фокусуванням;
– автоматична/ручна діафрагма;
– ручне регулювання нахилу;
– світлодіодне підсвічування з дистанційним перемиканням з білого на інфрачервоне;
– ручне масштабування 40 до 1;

3) спереду;

– фіксований фокус при слабкому освітленні;
– автоматична діафрагма;

б) програмне забезпечення;

1) вікно перегляду;

– «картинка в картинці»;
– «плаваючі картинки»;
– можна змінювати розмір вікон перегляду.

1.4 Висновки

У першому розділі кваліфікаційної роботи магістранта було проведено огляд сучасних систем комп'ютерного зору, існуючих бортових систем комп'ютерного зору, систем комп'ютерного зору для пошуку вибухонебезпечних об'єктів. В ході аналізу виявилося, що методи комп'ютерного зору широко використовуються у різних сферах нашого життя для рішення різного роду задач. Одним із популярних методів розпізнавання та ідентифікації об'єктів є використання нейронної мережі та глибоко навчання. Одним із цікавих варіантів такої нейронної мережі є YOLO, бо вона дозволяє швидко виконувати пошук об'єктів на зображенні. Також аналіз систем комп'ютерного зору для пошуку вибухонебезпечних об'єктів показав, що всі системи мають декілька камер, зображення з яких виводиться в окремих вікнах між якими можна переключатися. Крім цього розглянуті системи для пошуку вибухонебезпечних об'єктів не мають функції розпізнавання та ідентифікації вибухонебезпечних об'єктів. Розроблювана система буде мати можливість розпізнавати та ідентифікувати вибухонебезпечні об'єкти. Більш того, зображення за камер буде склеюватися в одне панорамне, завдяки чому користувачу не потрібно буде перемикатися між камерами.

2 РОЗРОБКА АРХІТЕКТУРИ БОРТОВОЇ БАГАТОЗОНОВОЇ СИСТЕМИ КОМП`ЮТЕРНОГО ЗОРУ

2.1 Основні характеристики бібліотеки OpenCV

OpenCV – це величезна бібліотека з відкритим вихідним кодом для комп’ютерного зору, машинного навчання та обробки зображень і тепер вона відіграє важливу роль у роботі в режимі реального часу, що дуже важливо в сучасних системах [21]. У 1999 році Гері Брадскі, працюючи в корпорації Intel, запустив OpenCV з надією прискорити комп’ютерне зір і штучний інтелект, забезпечивши надійну інфраструктуру для всіх, хто працює в цій галузі. Бібліотека написана на C і C++ і працює на Linux, Windows і Mac OS. Ведеться активна розробка інтерфейсів для Python, Java, MATLAB та інших мов, включаючи можливість використання бібліотеки на Android і iOS для мобільних додатків. Протягом багатьох років OpenCV отримував більшу частину підтримки від Intel і Google.

OpenCV було розроблено для ефективності обчислень і з акцентом на програмах які працюють у режимі реального часу. Бібліотека написана на оптимізованому C++ коді і може використовувати переваги багатоядерних процесорів. Якщо потрібна подальша автоматична оптимізація на архітектурах Intel, є можливість використовувати бібліотеки Intel Integrated Performance Primitives, які складаються з низькорівневих оптимізованих процедур у багатьох різних алгоритмічних областях. OpenCV автоматично використовує відповідну бібліотеку Intel Integrated Performance Primitives під час виконання, якщо бібліотеку встановлено. Починаючи з OpenCV 3.0, Intel надала безкоштовну підмножину Intel Integrated Performance Primitives, яка вбудована в OpenCV і прискорює її за замовчуванням.

Однією з цілей OpenCV є створення простої у використанні інфраструктури, яка допомагає людям швидко створювати досить складні програми комп’ютерного

зору. Бібліотека складається з 2500 оптимізованих алгоритмів, до яких входять як класичні, так і сучасні алгоритми комп'ютерного зору та машинного навчання [22]. Ці алгоритми охоплюють багато областей комп’ютерного зору, включаючи перевірку продукції на заводі, медичні зображення, безпеку, інтерфейс користувача, калібрування камери, стереобачення та робототехніку. OpenCV містить бібліотеку машинного навчання загального призначення. Ця бібліотека зосереджена на статистичному розпізнаванні образів і кластеризації. Модуль ML дуже корисний для завдань розпізнавання образів, які є основними завданнями OpenCV, але він достатньо загальний, щоб використовувати його для будь-якої проблеми машинного навчання.

Ліцензію з відкритим вихідним кодом для OpenCV структуровано таким чином, щоб була можливість створити комерційний продукт, використовуючи OpenCV повністю або частково. Не зобов’язані відкривати вихідний код свого продукту чи повертати вдосконалення у суспільне надбання. Існує велика спільнота користувачів, до якої входять представники великих компаній (IBM, Microsoft, Intel, SONY, Siemens і Google, якщо назвати лише деякі) і дослідницьких центрів (таких як Стенфорд, Массачусетський технологічний інститут, Кембридж та INRIA). Існує форум Yahoo Groups, де користувачі можуть ставити запитання та обговорювати. OpenCV популярний у всьому світі, з великими спільнотами користувачів у Китаї, Японії, Європі та Ізраїлі. З моменту випуску альфа-версії в січні 1999 року OpenCV використовувався в багатьох програмах, продуктах і дослідженнях. Ці програми включають зшивання зображень у супутникових і веб-картах, вирівнювання сканованих зображень, зменшення шуму на медичних зображеннях, аналіз об’єктів, системи безпеки та виявлення вторгнень, системи автоматичного моніторингу та безпеки, системи перевірки виробництва, калібрування камер, військові застосування та безпілотні літальні апарати, наземні та підводні апарати. Його навіть використовували для розпізнавання звуку та музики, де методи комп`ютерного зору застосовуються до зображень звукової спектрограми. OpenCV був ключовою

частиною системи бачення в роботі зі Стенфорда «Стенлі», який виграв перегони пустельних роботів DARPA Grand Challenge.

OpenCV спрямований на надання основних інструментів, необхідних для вирішення проблем комп'ютерного зору. У деяких випадках функціональних можливостей високого рівня в бібліотеці буде достатньо для вирішення більш складних проблем комп'ютерного зору. Навіть якщо це не так, базові компоненти в бібліотеці є достатньо повними, щоб можна було створити власний комплекс майже для будь-якої проблеми комп'ютерного зору. В останньому випадку є кілька випробуваних методів використання бібліотеки. Усі вони починаються з вирішення проблеми, використовуючи якомога більше доступних компонентів бібліотеки. Як правило, після того, як розробили першу чернетку рішення, можна побачити, де є недоліки, а потім виправити ці недоліки за допомогою власного коду.

OpenCV отримує багато внесків користувачів, і централізована розробка значною мірою вийшла за межі Intel. Із появою багатоядерних процесорів і багатьох нових програм комп'ютерного зору цінність OpenCV почала зростати. Подібним чином швидке зростання у сфері робототехніки спонукало до значного використання та розвитку бібліотеки. Після того, як OpenCV стала бібліотекою з відкритим вихідним кодом, кілька років активно розроблялася в Willow Garage, а тепер її підтримує фонд OpenCV. Сьогодні OpenCV активно розробляється фондом, а також кількома державними та приватними установами.

OpenCV складається з декількох шарів. Верхній шар складається з ОС, на якій працює OpenCV. Наступний шар це мови програмування та зразки програм, які використовують OpenCV. Нижче наведено шар у якому надано код у opencv_contrib, який містить переважно функції вищого рівня. Далі ядро OpenCV, а останній шар – різні апаратні оптимізації на рівні апаратного прискорення. На рисунку 2.1 наведені усі вище зазначені шари з яких складається OpenCV.

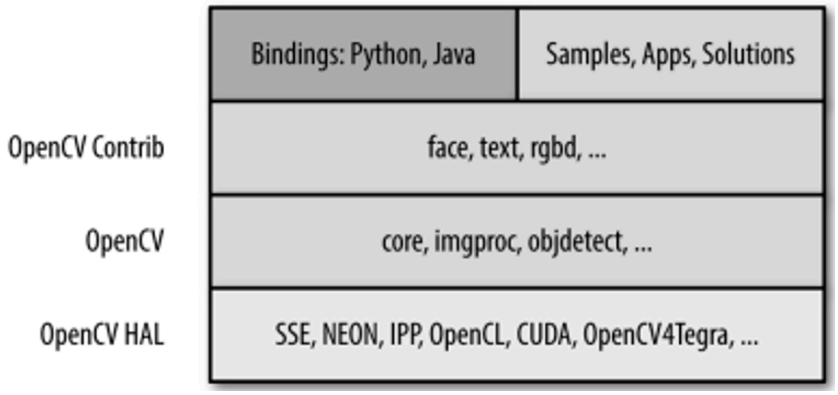

Рисунок 2.1 – Структура OpenCV

На рисунку 2.2 наведений графік розвитку бібліотеки OpenCV.

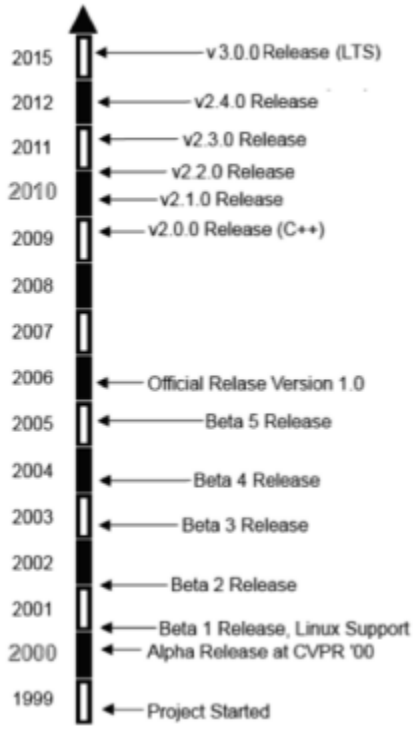

Рисунок 2.2 – Графік розвитку бібліотеки OpenCV

Бібліотека OpenCV використовується у різних сферах для вирішення різних задач, а саме:

– аналіз та обробка зображень;

– системи з розпізнавання обличчя;

– ідентифікації об'єктів;

– розпізнавання жестів на відео;

– побудова 3D моделей об'єктів;

– створення 3D хмар точок зі стерео камер;

– склеювання зображень між собою, для створення зображень всієї сцени з високою роздільною здатністю;

– система взаємодії людини з комп'ютером;

– пошуку схожих зображень із бази даних;

– стеження за рухом очей;

– аналіз руху;

– ідентифікація об'єктів;

– сегментація зображення;

– тренінг відео;

– розпізнавання елементів сцени і додавання маркерів для створення доповненої реальності.

Отже, OpenCV є дуже потужною бібліотекою комп’ютерного зору, за допомогою якої можна вирішувати велику кількість задач комп’ютерного зору. Бортова багатозонова система комп’ютерного зору повинна мати можливість отримувати зображення з камер та обробляти їх певним чином. Для вирішення цих задач необхідно використовувати методи комп’ютерного зору, а бібліотека OpenCV надає дуже велику їх кількість. Тож цю бібліотеку доцільно використовувати для реалізації бортової багатозонової системи комп’ютерного зору

2.2 Засоби навчання системи комп’ютерного зору на основні YOLO

Однією із важливих областей комп'ютерного зору є виявлення об`єктів. Для вирішення задач із виявлення об`єктів використовуються різні моделі машинного навчання і глибокого навчання.

В епоху глибокого навчання виявлення об’єктів можна поділити на дві категорії: двоступеневі детектори та одноступеневі детектори, де перші роблять «грубий», «неточний» процес визначення, а другі – завершують в один крок [23]. Двоступеневі детектори в основному зосереджені на вибірковій стратегії пропозицій регіону через складну архітектуру. Раніше дуже активно використовувалися для вирішення різного роду задач. Нещодавно з`явилися одноступеневі детектори, які зосереджуються на всіх пропозиціях просторової області для можливого виявлення об’єктів за допомогою відносно простішої архітектури за один кадр. Завдяки використанню базових алгоритмів та одноступеневості вони стали набагато кращими ніж двоступеневі детектори об`єктів і активно використовуються у різних контекстах.

Продуктивність детекторів об`єктів оцінюється через точність та час висновку. Як правило, точність виявлення двоступеневих детекторів перевершує одноступеневі детектори об’єктів. Однак час висновку одноступінчастих детекторів кращий порівняно з аналогами. З появою YOLO та його вдосконалених наступників точність виявлення значно покращилась і вона іноді краща, ніж у двоступеневих детекторів. YOLO використовується в різних сферах через можливість робити швидкі висновки, а не через точність виявлення. В якості прикладу можна порівняти YOLO та Fast-RCNN. Точність виявлення у випадку з YOLO становить 63,4 відсотки і 70 відсотків для Fast-RCNN, однак час надання висновку приблизно в 300 разів більше у YOLO.

Класифікація зображень – це завдання класифікації зображення або об’єкта на зображенні за однією з попередньо визначених категорій. Для вирішення цієї проблеми зазвичай використовують машинне навчання або алгоритмів глибокого навчання. У цих випадках моделі навчаються на великому наборі розмічених даних або датасетах. Якщо мова йде про машинне навчання то для вирішення задач класифікації використовуються моделі машинного навчання, які включають в себе ANN, SVM, дерева рішень і KNN. У випадку із глибоким навчанням CNN та його архітектурні наступники домінують над іншими моделями для вирішення задач класифікації.

Локалізація об’єкта – це завдання визначення положення об’єкта або кількох об’єктів на зображенні за допомогою прямокутної рамки навколо об’єкта, відомої як обмежувальна рамка. Однак сегментація зображення – це процес поділу зображення на кілька сегментів, де сегмент може містити повний об’єкт або частину об’єкта. Сегментація зображення у більшості випадків використовується для визначення місцезнаходження об’єктів, ліній і кривих, межі об’єкта або сегмента зображення. Виявлення об`єкта складається з класифікації, локалізації та сегментації. Це завдання правильної класифікації та ефективної локалізації одного чи кількох об’єктів на зображенні, як правило, за допомогою керованих алгоритмів, які мають достатньо великий навчальний набір з мітками. На рисунку 2.3 наведено класифікацію, локалізацію та сегментацію для одного та кількох об`єктів на зображенні в контексті виявлення об`єктів.

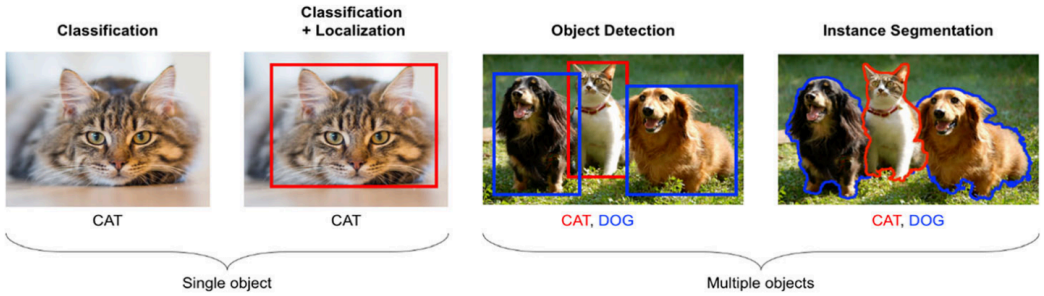

Рисунок 2.3 – Класифікація, локалізація та сегментація для одного та кількох об`єктів на зображенні

В наші дні існує п`ять версій YOLO. Кожна версія більш оптимізована в порівнянні з попередником. Автори YOLO розглядають проблему виявлення об’єктів як проблему регресії а не класифікації. Згорткова нейронна мережа передбачає обмежувальні рамки, а також імовірності класів для всіх об’єктів на зображенні. Через те, що цей алгоритм визначає об’єкти та їх розташування за допомогою обмежувальних рамок, дивлячись на зображення лише один раз, вони назвали його You Only Look Once. Згорткові нейронні мережі дуже добре працюють із

зображеннями для виділення функцій через те, що функції низького рівня ефективно поширюються від початкових згорткових шарів до наступних згорткових шарів у глибокій згортковій нейронній мережі. У цьому випадку проблема полягає в точній ідентифікації кількох об’єктів разом із їх точним розташуванням на одному зображенні. Спільне використання параметрів і фільтрів є двома важливими функціями згорткової нейронної мережі, які надають здатність ефективно впоратися з проблемою виявлення об’єктів.

У процесі виявлення об’єктів зображення ділиться на комірки сітки розміром S × S, кожна комірка сітки передбачає B обмежувальних прямокутників разом із їхніми розташуванням та розмірами, ймовірністю об’єкта в базовій сітці та ймовірностями умовного класу. В основу концепції виявлення об’єкта будь-якою коміркою сітки закладено те, що центр об’єкта повинен знаходитися всередині цієї комірки сітки. Комірка сітки відповідальна за виявлення конкретного об’єкта за допомогою будь-якої відповідної обмежувальної рамки.

Для сітки передбачаються параметри для однієї обмежувальної рамки, де перші п’ять параметрів є специфічними для обмежувальної рамки. Решта розподіляються між усіма обмежувальними рамками однієї сітки, незалежно від кількості обмежувальних рамок. Параметри, які передбачаються для обмежувальних рамок описуються наступною формулою (2.5):

$$p = p_c, b_x, b_y, b_w, b_h, p(c_1), p(\mathrm{c}_2), \ldots, p(\mathrm{c}_n), \tag{2.5}$$

де p_c – ймовірність утримання об’єкта в сітці базовою обмежувальною рамкою;

(b_x, b_y) – координати центру передбаченої обмежувальної рамки;

(b_w, b_h) – прогнозований розмір обмежувальної рамки;

$p(c_i)$ – умовний клас ймовірності того, що об’єкт належить до цього класу для даного p_c.

На рисунку 2.4 наведено схему передбачення вихідного тензора, коли вхідне зображення розділено на сітки 19 × 19 і передбачено чотири обмежувальні прямокутники для кожної сітки, де ймовірності класу розподіляються між усіма обмежувальними рамками для конкретної сітки.

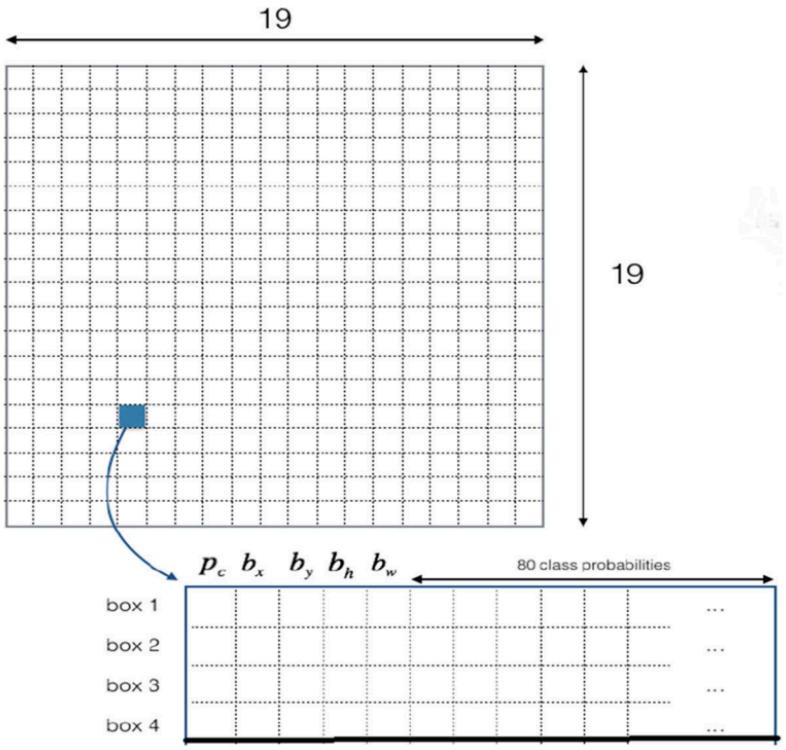

Рисунок 2.4 – Схема передбачення вихідного тензора

Оцінка достовірності обчислюється для кожної обмежувальної рамки на сітку шляхом множення p_c на перетин через об'єднання між основною правдою та прогнозованою обмежувальною рамкою. Якщо об'єкт не існує в комірці сітки, оцінка достовірності буде нульовою. На наступному кроці для кожної обмежувальної рамки всіх клітинок сітки обчислюється оцінка класу. Ця оцінка для певного класу задає як ймовірність появи класу в цьому полі, так і те, наскільки добре передбачене поле відповідає об'єкту.

Як правило, ці обмежувальні рамки відрізняються за розміром, враховуючи різні форми для захоплення різних об'єктів, відомі як опорні рамки. Об'єкт на

зображенні має розпізнаватися обмежувальною рамкою так, щоб центр об’єкта знаходився в цій обмежувальній рамці. Однак може існувати можливість розміщення центрів кількох об’єктів в одній обмежувальній рамці. Автори використали інший термін прив’язки для позначення обмежувальних рамок, що відповідають одній клітинці сітки. Блоки прив’язки — це лише набір із кількох стандартних обмежувальних рамок, вибраних після аналізу набору даних і базових об’єктів у наборі даних. Ці вибрані прив’язки мають представляти більшість класів/категорій, враховуючи різні комбінації ширини та висоти, такі як квадрат, вертикальний або горизонтальний прямокутник тощо, щоб відповідати пропорціям і масштабу всіх об’єктів, присутніх у наборі даних.

Передбачити той самий об’єкт також можуть і сусідні комірки сітки, тобто передбачати обмежувальні рамки, що перекриваються, для того самого об’єкта. Це означає, що буде кілька передбачень, через те, що сусідні комірки сітки можуть вважати, що центр об’єкта знаходиться всередині них. На рисунку 2.5 наведено прогнозування кількох обмежувальних прямокутників для об’єкта.

Рисунок 2.5 – Прогнозування кількох обмежувальних прямокутників для об’єкта

На рисунку 2.6 наведено високе та низьке перекриття між прогнозованим прямокутником та істиною.

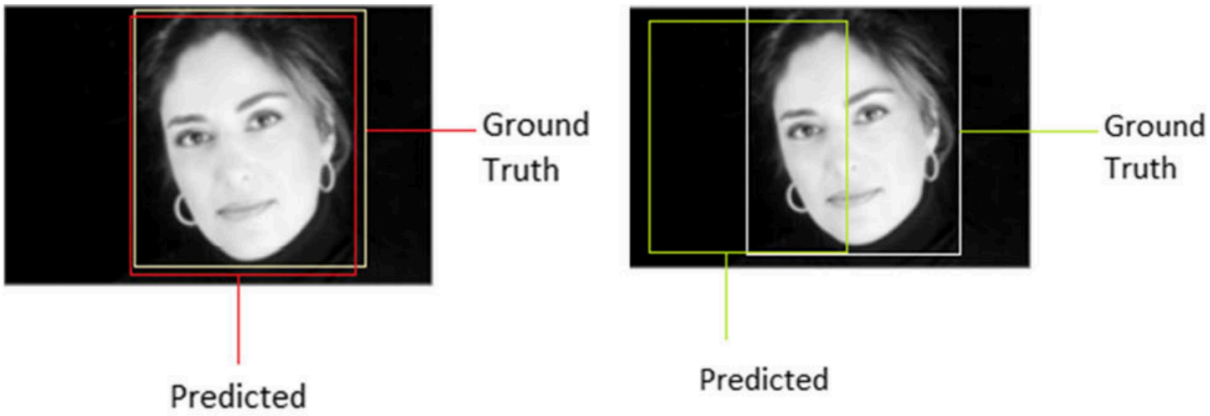

Рисунок 2.6 – Високе та низьке перекриття між прогнозованим прямокутником та істиною

На цьому етапі кожна обмежувальна рамка всіх сіток буде мати бал класу, координати рамки та категорією результату класифікації. Після цього буде отримана загальна кількість передбачених обмежувальних рамок. Якщо оцінка класу буде нижче за попередньо визначене порогове значення, то ці обмежувальні рамки не будуть використовуватися далі. В загальному випадку цей поріг дорівнює 0,5 але це значення може змінюватися в залежності від набору даних і його характеристик. Низький бал може бути у тому випадку, якщо низька ймовірність будь-якої конкретної категорії класу, яка максимізує бал класу або низька ймовірність вмісту об’єкта в цій сітці.

Після того, як кількість обмежувальних рамок зменшилась в наслідок фільтрації за пороговим значенням їх кількість все одно дуже велика. Для того, щоб на наступних етапах зменшити кількість обмежувальних рамок використовується немаксимальне придушення. Приклад результату використання цього критерію наведено на рисунку 2.7.

Рисунок 2.7 – Приклад результату використання немаксимального придушення

Немаксимальне придушення базується на концепції перетину через об’єднання. Об’єднання можна обчислити для двох блоків або іншими словами множин. В першу чергу обираються обмежувальні рамки із максимальним балом класу. Усі інші рамки, що перетинаються з обраною обмежувальною рамкою, будуть відкинуті, якщо обчислене значення перетину через об’єднання перевищує певне попередньо визначене порогове значення. Ці кроки будуть виконуватися доки не залишиться жодної обмежувальної рамки з нижчими оцінками достовірності, ніж вибрана обмежувальна рамка. На рисунку 2.8 наведено перетин та об’єднання двох обмежувальних рамок.

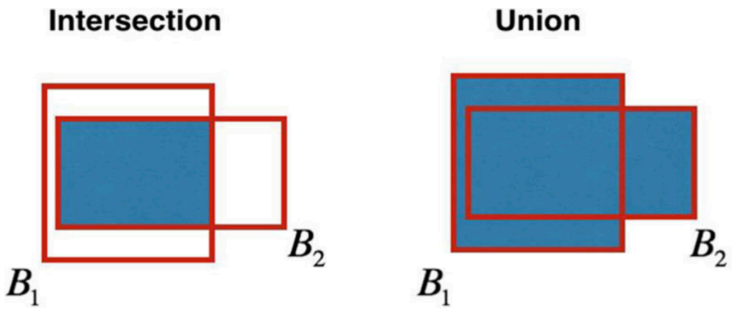

Рисунок 2.8 – Перетин та об’єднання двох обмежувальних рамок

Перетин двох рамок обчислюється за наступною формулою (2.6):

$$B = \frac{B_1 \cap B_2}{B_1 \cup B_2}, \tag{2.6}$$

де B – обмежувальна рамка, яка отримана в результаті об’єднання двох рамок;

B_1, B_2 – це обмежувальні рамки.

YOLO 1. Архітектура YOLO дуже схожа на архітектуру GoogLeNet. У YOLO початкові модулі GoogLeNet замінені згорткою 1×1, а потім згортковими фільтрами (3×3), тільки перший згортковий рівень має фільтр 7×7. YOLO має 24 шари згортки, за якими слідують 2 повністю з’єднані шари, як наведено на рисунку 2.9.

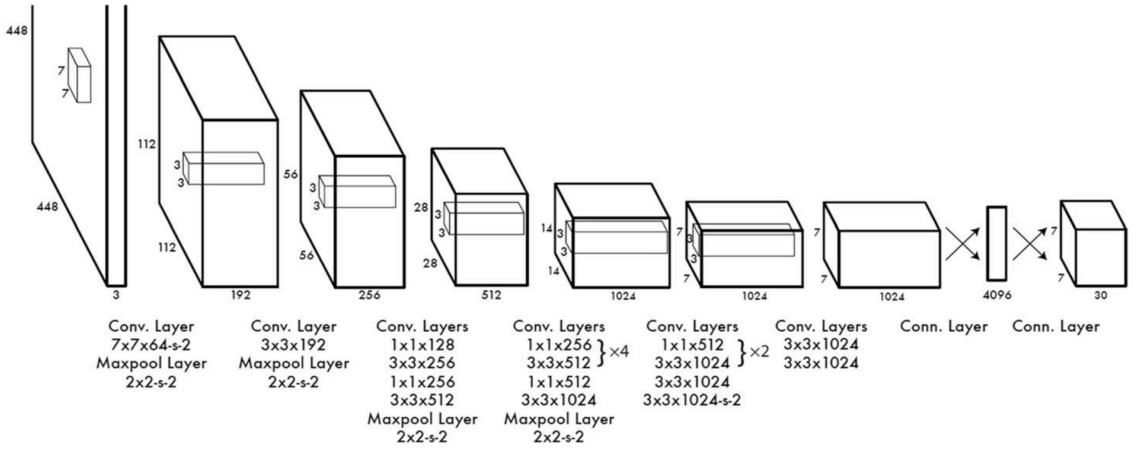

Рисунок 2.9 – Загальна структура YOLO 1

З цих 24 шарів згортки лише за чотирма згортковими шарами слідують шари максимального об’єднання. Алгоритм використовує згортку 1×1 і об’єднання глобальних середніх, які є основними перевагами цієї версії.

До недоліків YOLO 1 можна віднести велику похибку локалізації у порівнянні з двоступеневими детекторами об'єктів.

У разі необхідності швидкого виявлення об'єктів можна використовувати YOLO з меншою складністю моделі, відомий як Fast YOLO. Ця версія має лише 9

згортоких шарів із порівняно меншими фільтрами в цих шарах. Також, якщо необхідно виконувати виявлення об'єктів у реальному часі в системі без графічного процесора, то можна використовувати Yolo-lite.

YOLO 2. Ця версія YOLO являється наступником YOLO 1. Архітектура YOLO 2 використовує структуру Darknet-19, що складається з 19 згорткових шарів і 5 максимальних шарів об’єднання, як наведено на рис. 2.10.

Type	Filters	Size/Stride	Output
Convolutional	32	3×3	224×224
Maxpool		$2 \times 2/2$	112×112
Convolutional	64	3×3	112×112
Maxpool		$2 \times 2/2$	56×56
Convolutional	128	3×3	56×56
Convolutional	64	1×1	56×56
Convolutional	128	3×3	56×56
Maxpool		$2 \times 2/2$	28×28
Convolutional	256	3×3	28×28
Convolutional	128	1×1	28×28
Convolutional	256	3×3	28×28
Maxpool		$2 \times 2/2$	14×14
Convolutional	512	3×3	14×14
Convolutional	256	1×1	14×14
Convolutional	512	3×3	14×14
Convolutional	256	1×1	14×14
Convolutional	512	3×3	14×14
Maxpool		$2 \times 2/2$	7×7
Convolutional	1024	3×3	7×7
Convolutional	512	1×1	7×7
Convolutional	1024	3×3	7×7
Convolutional	512	1×1	7×7
Convolutional	1024	3×3	7×7
Convolutional	1000	1×1	7×7
Avgpool		Global	1000
Softmax			

Рисунок 2.10 – Структура Darknet-19

Використовується багато згорток 1×1, щоб досягти зменшення вибірки по всій глибині вхідного об’єму. YOLO 2 використовує глобальне середнє об’єднання та згортку 1×1. Також ця версія має нові методи оптимізації, такі як пакетна нормалізація, класифікатор високої роздільної здатності, згортка з блоками

прив’язки, кластери розмірів, пряме передбачення розташування, багатомасштабоване тренування.

В ході навчання нейронних мереж ваги ініціалізуються випадковим чином. В наслідок цього, вагові матриці послідовних шарів можуть мати різний розподіл. Один розподіл може бути змінено для прийняття змін іншого розподілу ваг.

Під час навчання нейронних мереж ваги ініціалізуються випадковим чином. Як наслідок, вагові матриці послідовних шарів можуть мати різний розподіл, один розподіл може бути змінено для прийняття змін іншого розподілу ваг. Зміни такого роду в параметрах прихованих шарів можуть призвести до непотрібних зрушень у більш глибоких прихованих шарах. Як наслідок виникає проблема внутрішнього коваріативного зсува. Для зменшення внутрішнього коваріативного зсуву виходи кожного прихованого шару нормалізуються за допомогою пакетної нормалізації для узгодженого розподілу вагових матриць між різними шарами.

Класифікатор високої роздільної здатності у базовій версії YOLO було навчено на зображеннях розміром 224×224 для класифікації та збільшення роздільної здатності до 448×448 для виявлення. Модель повинна була вивчити класифікацію і прийняти нове рішення. У цій версії модель навчалася на зображеннях розміром 448×448 для класифікації. Пізніше модель була налаштована для завдань виявлення об’єктів і пропонувати кращі обмежувальні прямокутники для вхідних даних із високою роздільною здатністю. За допомогою класифікатора з високою роздільною здатністю спостерігається приблизне збільшення mAP на 4%.

Повністю пов’язаний шар видалено з базової версії, оскільки він безпосередньо передбачав координати обмежувальної рамки. Замість цього відбувається прогнозування обмежувальних рамок замість. Крім того, передбачення зсуву робить модель простішою та швидкою для вивчення. Спостерігається приблизне збільшення запам’ятовування на 7% при використанні згортки з опорними блоками, однак також помічено зниження mAP на 0,3%.

У YOLO було дві проблеми. По-перше, вручну підібрані попередні параметри, які було вирішено за допомогою кластеризації k-середніх. По-друге, нестабільність моделі під час передбачення обмежувальної рамки. Випадкова ініціалізація вважається найбільшою перешкодою для прогнозування справжніх зсувів. Замість цього ми прогнозуємо координати розташування відносно розташування клітинок сітки. Ця додаткова функція збільшила mAP на 5%.

YOLO легко прогнозує великі об’єкти за допомогою карти об’єктів 13×13. Однак, об`єкти невеликого розміру будуть розпізнаватися не ефективно. Використовуючи ідею пропуску з’єднань у ResNet, функції вищої роздільної здатності об’єднуються з функціями нижчої роздільної здатності в послідовних каналах. На рисунку 2.11 наведено пропуск з’єднань у ResNet.

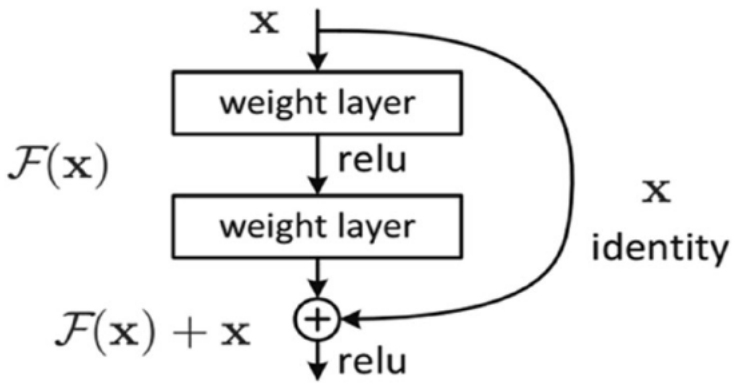

Рисунок 2.11 – Пропуск з’єднань у ResNet

Виконується зміна форми, тобто карта об’єктів розміром 26 × 26 × 512 перетворюється на карту об’єктів розміром 13 × 13 × 2048, а потім об’єднується з моделлю, як наслідок, прогнозований розмір результату становить 13 × 13 × 1024. Фінальна форма тензора буде 13 × 13 × 3072, для яких були застосовані фільтри. Додавання цього рівня збільшило mAP на 1%.

Після видалення повністю пов’язаних шарів YOLO може працювати з будь-яким виміром у діапазоні від 320 × 320 до 608 × 608. Модель випадковим чином

обирає вимір, який є кратним 32 на кожні 10 епох навчання. Завдяки цьому модель має гнучкість для прогнозування за різними параметрами

YOLO 3. Перша версія YOLO не мала рішення помилки локалізації, а друга версія не виявила об’єктів меншого розміру. У новій версії YOLO 3 надається більш ефективний спосіб виявлення об’єктів із покращеною продуктивністю під час навчання та оцінки на наборі даних COCO. Ця версія YOLO краще працює з об`єктами невеликого розміру ніж попередники але через це страждає отримання точних результатів для об’єктів середнього та великого розміру.

Архітектура YOLO 3 базується на структурі Darkent-53. Darkent-53 - це мережа, яка використовує згортковий фільтр 3×3 і 1×1 разом із деякими скороченими з’єднаннями. Ця мережа передбачає 53 згорткових шарів, що робить її значно потужнішою. Вона удвічі швидша, ніж ReNet-152 при цьому не страждає продуктивність.

Основна загальна архітектура YOLO 3 наведена на рис 2.12 [24].

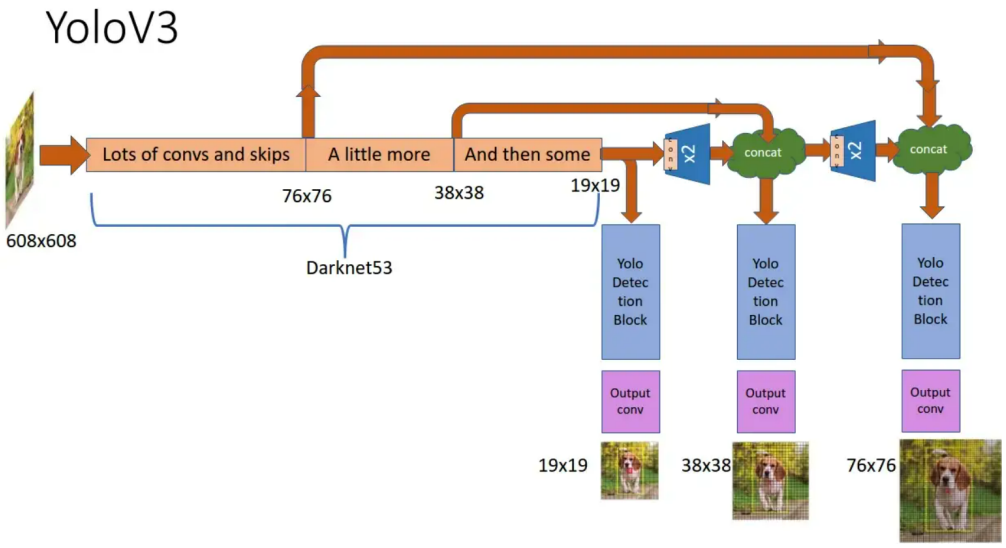

Рисунок 2.12 – Загальна архітектура YOLO 3

YOLO 3 використовує залишкові блоки, пропуск з’єднань і підвищення вибірки. Ця версія YOLO також використовує згортку 1×1 на картах об’єктів для

виявлення об’єктів. Вона створює карти у трьох різних масштабах. Зокрема, вона зменшує вибірку вхідних даних у різних масштабах із коефіцієнтом 32, 16 і 8.

YOLO 4. Ця версія представляє нові методи, демонструє різноманітні складні та потужні техніки. Ця версія швидша та точніша за всі попередні версії. Модель була спрямована на створення детектора об'єктів для виробничих систем. Автори розділили постановку проблеми та підходи на кілька сегментів, а потім запропонували різноманітні методології, які найкраще відповідають наявним ресурсам.

По-перше, надається вхідне зображення розміром H × W, де H означає висоту, а W означає ширину.

По-друге, серія згорток, які витягують функції через потужні мережі, такі як VGG16, Darknet53, ResNet50 та інші варіанти, які вони назвали магістральними. Для вилучення функцій у різних масштабах використовуються мережі агрегації шляхів та інші варіанти, які є композицією зав’язків між шляхами «знизу вгору» та «зверху вниз». Нарешті, передбачення опорних блоків з використанням одноступеневих детекторів, таких як YOLO, які мають щільні шари, або використання двоступеневих детекторів, таких як Faster R-CNN.

Що стосується архітектури, автори порівняли CSPResNeXt50, CSPDarknet53 і EfficientNetB3 для побудови архітектури YOLO 4. CSPDarknet53, мережа з 29 шарами згортки з фільтрами 3 × 3 і приблизно 27,6 мільйонами параметрів, обрана як магістраль, яка перевершила інші архітектури, де CSP означає Cross-stage partial connections. Мережі CSPNet допомагають у багатій комбінації градієнтів з мінімальними витратами на обчислення.

Для експериментів використовується попередньо навчена модель ImageNet для класифікації з базовим набором даних як COCO. Автори використовували Spatial Pyramid Pooling, який також використовувався у RCNN. Тут нейрона мережа, за якою слідують повністю пов’язані рівні, має обмеження на вхідний і вихідний обсяг. Однак ця версія забезпечує обробку будь-якого введення незалежно від розміру чи форми.

Spatial Pyramid Pooling розміщується між нейронною мережею і повністю підключеним рівнем, який відображає будь-який вхідний розмір на вихідний фіксований розмір. Це дає можливість виявляти об'єкти різного розміру, крім того, використовується генетичний алгоритм для налаштування гіперпараметрів [25].

YOLO 5. Ця версія заснована на попередніх моделях YOLO. Швидкість і точність виявлення значно покращені.

Загалом, алгоритм виявлення об`єкту розділений на 4 модулі, а саме вхідний рівень, опорну мережу, мережу Нека та головний вихідний рівень. Наступний аналіз удосконалень у YOLO 5 засновано на чотирьох модулях:

– рівень введення – YOLO 5 використовує метод покращення даних Mosaic для підвищення швидкості навчання моделі та точності мережі на етапі навчання моделі. Водночас запропоновано адаптивний розрахунок прив’язки та методи адаптивного масштабування зображення;

– еталонна мережа;

– мережа Нека;

– головний вихідний рівень – YOLO 5 успадковує механізм прив’язки вихідного рівня від YOLO 4.

Сімейство моделей YOLO складається з трьох основних архітектурних блоків:

– хребта;

– шиї;

– голови.

Хребет YOLO 5 використовує CSPDarknet як магістраль для виділення функцій із зображень, що складаються з перехресних часткових мереж. Шия YOLO 5 використовує PANet для створення мережі пірамід функцій для виконання агрегації функцій і їх передачі в голову для прогнозування. Голова YOLO 5 генерує прогнози з блоків прив’язки для виявлення об’єктів.

На рисунку 2.13 приведена архітектура нейронної мережі YOLO 5 [26].

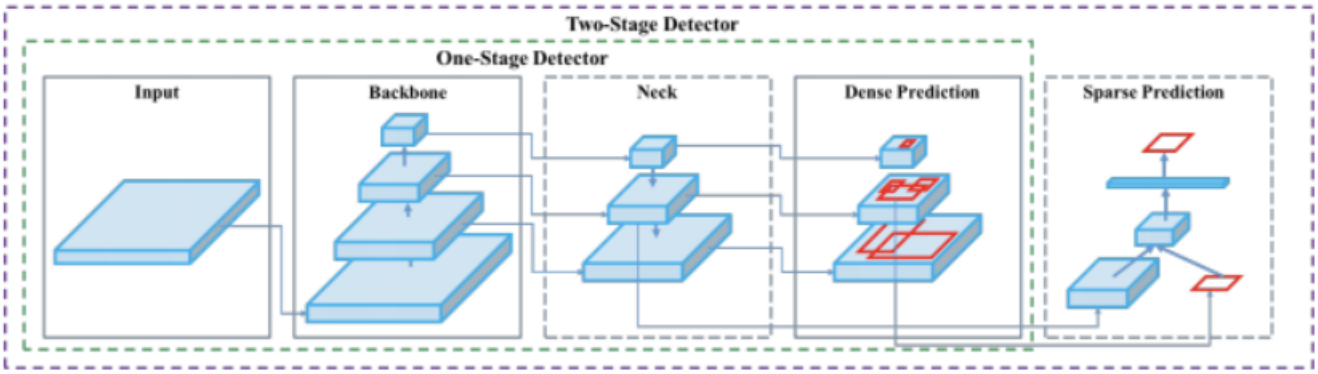

Рисунок 2.13 – Архітектура нейронної мережі YOLO 5

Окрім цього YOLO 5 використовує наведені нижче варіанти для навчання:

– активація та оптимізація;

– функція втрати.

YOLO 5 має кілька різновидів попередньо навчених моделей:

– YOLO 5n;

– YOLO 5s;

– YOLO 5m;

– YOLO 5l;

– YOLO 5x.

Різниця між ними полягає в компромісі між розміром моделі та часом висновку. Версія легкої моделі YOLO 5s займає лише 14 мб, але не дуже точна. З іншого боку, є YOLO 5x, розмір якої значно більше ніж у YOLO 5s і становить 166 мб, але це найточніша версія свого сімейства.

Переваги Yolo 5:

– приблизно на 88% менше, ніж YOLO 4 (27 мб проти 244 мб);

– приблизно на 180% швидше, ніж YOLO 4 (140 кадрів за секунду проти 50);

– приблизно така ж точність, як в YOLO 4 для одного и того ж завдання (0,895 mAP проти 0,892 mAP).

Недоліки Yolo 5 [27]:

– YOLO накладає сильні просторові обмеження для прогнозування обмежувального поля, оскільки кожна комірка сітки передбачає лише два поля і може мати лише один клас. Це просторове обмеження обмежує кількість об'єктів поблизу, які може передбачити наша модель.

Бортова багатозонова система комп’ютерного зору повинна мати можливість розпізнавати та ідентифікувати вибухонебезпечні об’єкти. Для вирішення цієї задачі можна використовувати нейронні мережі, які зараз є дуже популярними. Сімейство нейронних мереж YOLO, а саме YOLO 5 є хорошим варіантом для вирішення задач розпізнавання та ідентифікації у бортовій багатозоновій системі комп’ютерного зору. Це обумовлено тим, що в YOLO 5 значно покращені показники в порівнянні з попередніми нейронними мережами цього сімейства Крім цього передбачається, що система, що розроблюється бортова і може бути встановлена на роботі з не дуже потужним процесором та без потужного графічного процесора. YOLO 5 має п’ять типів моделей і можна буде обрати ту модель нейронної мережі, яка більше всього підходить за технічними характеристиками.

2.3 Розробка архітектури системи комп`ютерного зору

Для того, щоб описати систему комп`ютерного зору буде використовуватися теорія множин. Довільна сукупність об’єктів нашої інтуїції чи інтелекту, які можна відрізнити один від іншого і які складають єдине ціле, називається множиною. Об’єкти, які утворюють множину, називають її елементами [28]. Множина системи комп`ютерного зору, буде складатися з n камер, m позицій та k типів оптики. Для того, щоб описати систему комп`ютерного зору необхідно спочатку описати підмножини, які входять до системи комп`ютерного зору.

Підмножина камер для системи комп`ютерного зору описує формула (2.1):

$$K = \{ K_1, K_2, K_3, \ldots, K_n \}, \quad (2.1)$$

де K_1-K_n – це камери.

Підмножина оптики для системи комп`ютерного зору описує формула (2.2):

$$O = \{ O_1, O_1, O_1, \ldots, O_k \}, \quad (2.2)$$

де O_1-O_k – це оптика.

Підмножина розташування камер з оптикою для системи комп`ютерного зору описує формула (2.3):

$$P = \{ P_1, P_1, P_1, \ldots, P_m \}, \quad (2.3)$$

де P_1-P_m – це розташування камер з оптикою.

Після опису підмножин можна описати множину системи комп’ютерного зору. Множину системи комп’ютерного зору описує формула (2.4):

$$\text{СКЗ} = K \cup O \cup P, \quad (2.4)$$

де *К* – це підмножина камер;

О – це підмножина оптики;

Р – це підмножина розташування камер з оптикою.

Структура системи комп’ютерного зору може бути різноманітною. Одним із варіантів розташування камер з оптикою може бути передня, задня, ліва, права частини корпусу. На рисунку 2.14 можна побачити розташування блоків камер з оптикою (К1 – К4 та О1 – О4 відповідно) [29].

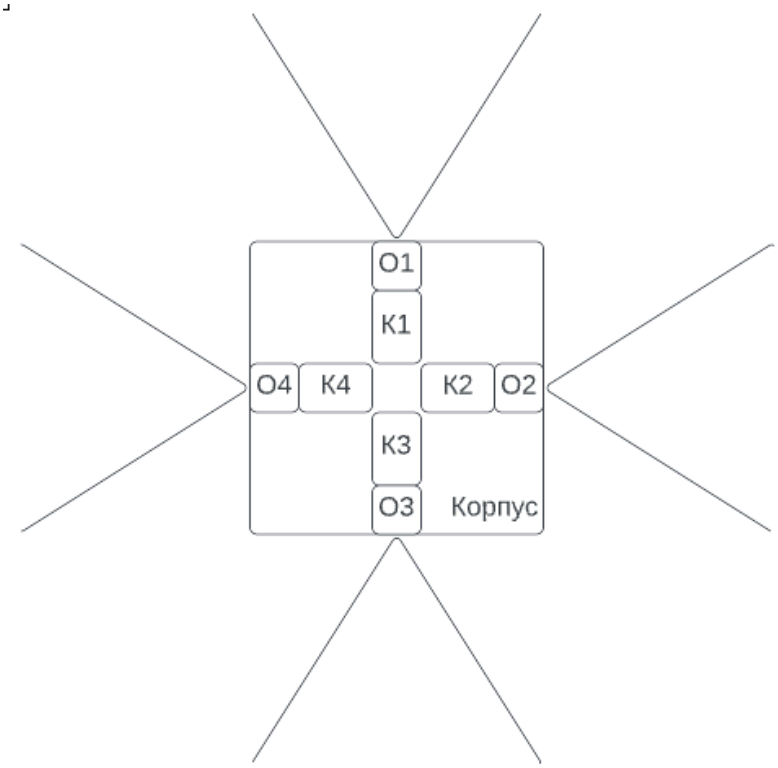

Рисунок 2.14 – Структурна схема системи комп`ютерного зору

2.4 Висновки

У другому розділі кваліфікаційної роботи магістранта було проведено аналіз технологій, які можна використовувати для реалізації бортової багатозонової системи комп’ютерного зору. Було проаналізовано популярну бібліотеку комп’ютерного зору OpenCV. Крім цього було проаналізовано сімейство нейронних мереж YOLO. У системі, що розроблюється, будуть використовуватися обидві технології. Також систему комп’ютерного зору було описано за допомогою теорії множин. Було отримано множину системи комп’ютерного зору. Запропоновано структуру системи комп’ютерного зору, яка складається з чотирьох камер.

3 РОЗРОБКА БОРТОВОЇ БАГАТОЗОНОВОЇ СИСТЕМИ КОМП`ЮТЕРНОГО ЗОРУ

3.1 Опис використаних технологій

Для розробки програмного забезпечення бортової багатозонової системи комп`ютерного зору було вирішено використовувати мову програмування Python 3 та OpenCV 4.6.0.

Python – це доволі популярна мова програмування загального призначення високого рівня. Вона була створена у 1991 році та вдосконалена Python Software Foundation. Її розроблено з акцентом на зручність читання коду, а синтаксис дозволяє програмістам висловлювати свої концепції в меншій кількості рядків коду ніж в інших мовах програмування [30].

Python має відносно простий синтаксис в порівнянні з іншими мовами програмування, такими як Java, C++, C#, тощо. Ця перевага дає можливість за доволі невеликий час розробити програмний застосунок. Також швидкість розробки застосунку за допомогою цієї мови програмування обумовлена її популярністю і як наслідок великою кількістю корисної інформації. Ще однією дуже великою перевагою Python є те, що цю мову доволі часто використовують для вирішення задач глибокого навчання. Бортова багатозонова система комп`ютерного зору буде використовувати методи глибокого навчання для вирішення задач розпізнавання та ідентифікації.

До основних переваг цієї мови програмування можна віднести [31]:

– легко вивчити та читати код;

– інтерпретована;

– динамічна;

– безкоштовна;

– високорівнева;

– може працювати на різних операційних системах;

– є можливість розширення за допомогою інших мов програмування.

OpenCV – це доволі популярна безкоштовна бібліотека, яка має дуже велику кількість інструментів, які використовуються для вирішення проблем комп`ютерного зору. Цю бібліотеку було обрано через її популярність і доступність. Існує багато бібліотек, які використовуються для роботи із зображеннями але у них є недоліки до яких можна віднести: високу вартість, меншу кількість інструментів. Через свою популярність ця бібліотека має велику кількість інформації щодо вирішення різних проблем комп`ютерного зору.

Отже підводячи висновки щодо обраної бібліотеки для роботи із зображеннями:

– безкоштовна;

– велика кількість інформації;

– надає велику кількість інструментів вирішення проблем комп`ютерного зору.

Для вирішення задачі розпізнавання та ідентифікації об`єктів на зображенні було вирішено використовувати нейрону мережу YOLO 5. Існує 5 різновидів цієї мережі:

– YOLO 5n;

– YOLO 5s;

– YOLO 5m;

– YOLO 5l;

– YOLO 5x.

Для програмної реалізації було обрано YOLO 5s. Це рішення обумовлено через її відносно інших різновидів YOLO невеликі системні вимоги до апаратної частини та точність. Щодо інших варіантів, то YOLO 5n має дуже погану точність розпізнавання та ідентифікації та її не рекомендується використовувати. YOLO 5m

має трішки кращу точність ніж YOLO 5s але вимагає вже більш потужну апаратну частину. YOLO 5l та YOLO 5x доволі точні але, як і у випадку з YOLO 5m, потребують доволі потужну апаратну частину. Під потужною апаратною частиною мається на увазі швидкий центральний процесор та наявність потужного графічного процесора. Так як передбачається можливість використання цієї системі на Rasberi PI 4, то є обмеження щодо потужності апаратної частини. Саме тому було вирішено використовувати YOLO 5s для реалізації розпізнавання та ідентифікації.

3.2 Апаратна реалізація

Оскільки система комп’ютерного зору повинна бути багатозоновою, то необхідно використовувати декілька камер, щоб покрити велику ділянку простору. Для реалізації апаратної частини бортової багатозонової системи комп`ютерного зору було вирішено використовувати три камери, кожна з яких покриває певну частину простору. На рисунку 3.1 наведено структурну схему бортової багатозонової системи комп`ютерного зору.

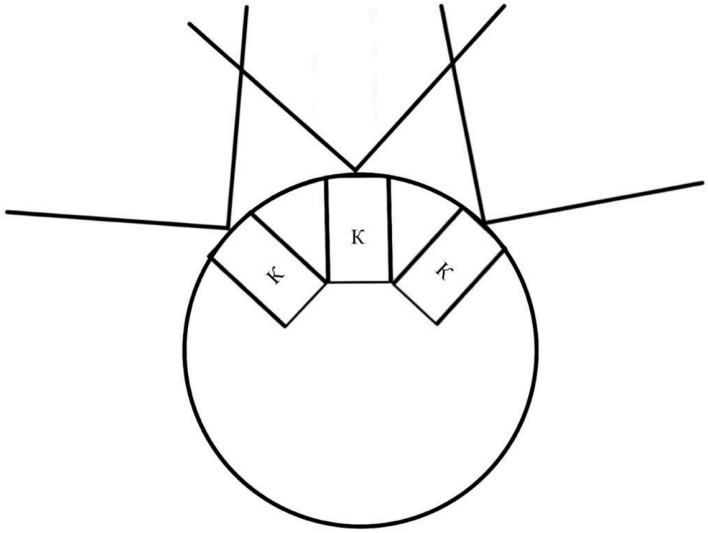

Рисунок 3.1 – Структурна схема бортової багатозонової системи комп`ютерного зору

Як можна зауважити на рисунку, фокусний простір камер перетинається по краях. Це зроблено для того, щоб в подальшому за допомогою програмного забезпечення можна було склеїти ці зображення в одне і створити панорамне зображення.

3.3 Програмна реалізація

В ході програмної реалізації необхідно було виконати наступні кроки:

– навчити нейронну мережу розпізнавати та ідентифікувати об`єкти;

– отримати зображення з камер;

– склеїти зображення з камер у панорамне та виконати пошук об`єктів на ньому.

Як вже було сказано раніше, для програмної реалізації бортової багатозонової системи комп`ютерного зору буд використовуватися нейронна мережа YOLO V5s.

Для того, щоб навчити нейронну мережу необхідно:

– знайти фотографій за якими буде відбуватися навчання нейронної мережі;

– виконати розмітку даних;

– використовуючи результати з попередніх етапів навчити нейронну мережу розпізнавати та ідентифікувати.

Система, що розроблюється буде мати змогу розпізнавати та ідентифікувати три типи об`єктів:

– протипіхотну міну ПФМ-1;

– протитанкову міну ТМ-62М;

– артилерійський снаряд.

Для створення набору даних було знайдено 38 фотографій протитанкової міни ТМ-62М, 36 фотографій артилерійських снарядів, 35 фотографій протипіхотних мін ПФМ-1.

Після пошуку зображень із об`єктами, які нас цікавлять, необхідно виконати розмітку даних. Розмітка даних являє собою процес при якому необхідно за допомогою спеціальної програми вказати ту частину зображення, на якій знаходиться об`єкт. Для виконання розмітки даних використовується програма під назвою «label image» . Розмітка даних відбувалася наступним чином. В програму було завантажено зображення і після чого для кожного зображення було вказано область із об`єктом, який нас цікавить. На рисунку 3.2 наведено процес розмітки зображення.

Рисунок 3.2 – Процес розмітки зображення

В результаті проведення розмітки зображення отримуються два txt файли. В першому зберігаються назви класів об`єктів. Цей файл створюється один раз в самому початку і далі лише оновлюється у випадку додавання нових класів об`єктів. На рисунку 3.3 наведено вміст txt файлу в якому зберігаються назви класів об`єктів.

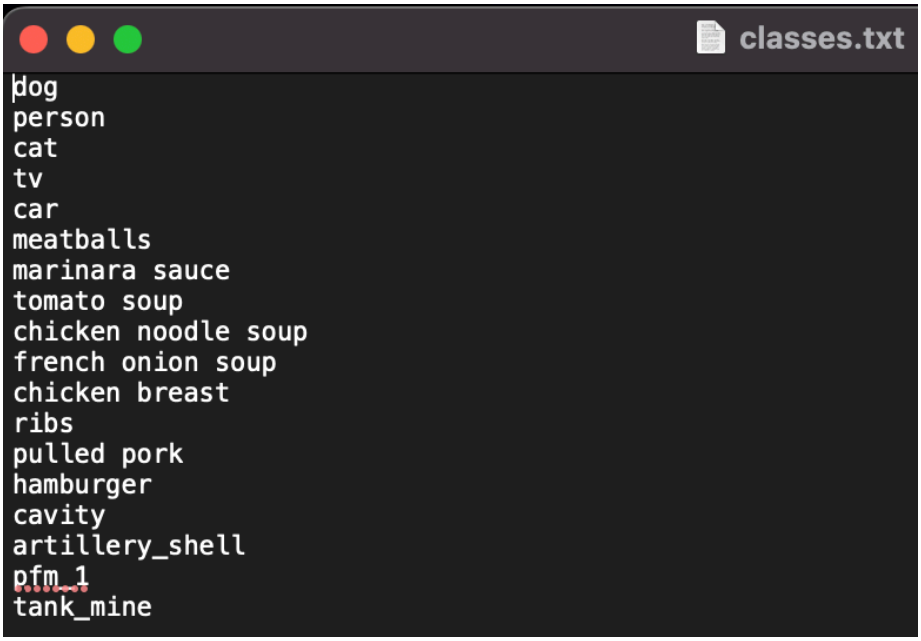

Рисунок 3.3 – Вміст txt файлу із назвами класів об`єктів

В іншому файлі зберігається результат розмітки зображення. Цей файл строюється для кожного зображення свій. На рисунку 3.4 наведено txt файл із результатами розмітки.

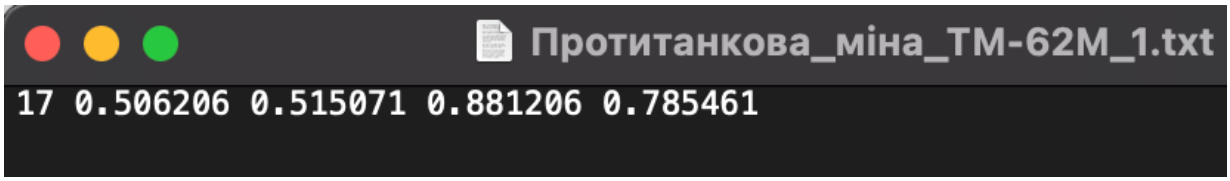

Рисунок 3.4 – Txt файл із результатами розмітки

Щоб навчити нейронну мережу було проведено 327 епох або ітерацій навчання. Як можна побачити на рисунку 3.5, модель швидко покращувалася з точки зору точності, запам’ятовування та середньої точності приблизно до 130-ї ітерації. Такі показники, як box_loss, obj_loss, cls_loss також показали швидке зниження приблизно до 130-ї ітерації. Це доволі непоганий результат, якщо брати до уваги невеликий розмір набору даних.

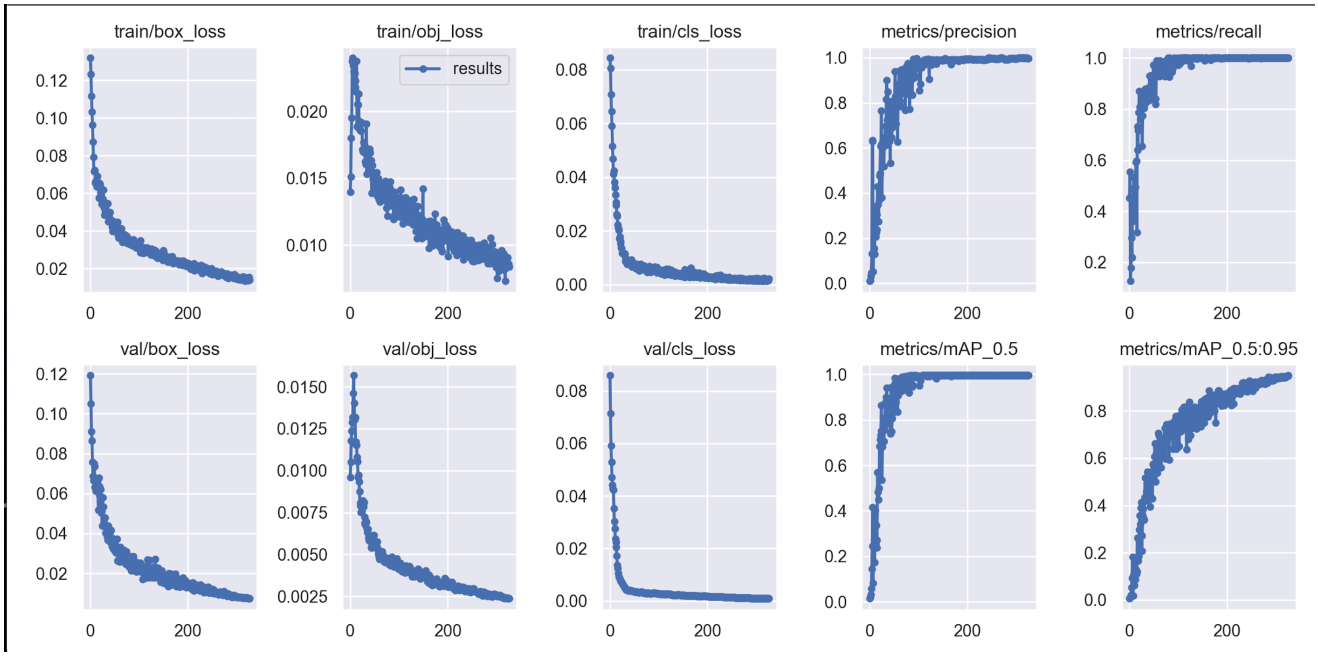

Рисунок 3.5 - Результати навчання нейронної мережі

Далі необхідно отримати зображення з камер. Для того, щоб отримати зображення із камери в програмному додатку, що розроблюється, використовується бібліотека OpenCV, а саме клас VideoCapture. Для того, щоб отримати зображення із камери необхідно у конструктор цього класу передати числове значення, а саме індекс камери з якої береться зображення, і потім викликати метода read на екземплярі класу VideoCapture.

В ході написання програми виникла проблема через обмеження зчитування зображень із декількох камер. Максимальна кількість камер з яких можна зчитувати одночасно зображення у програмному додатку дорівнює двом.

Для того, щоб вирішити цю проблему було вирішено створити багатопотоковий програмний додаток. Кожній камері відповідає свій потік, який отримає зображення з камери. Такий додаток може підтримувати безліч камер одночасно.

Для зручності створення багатопотокового додатку було вирішено створити клас VideoCapture. Цей клас використовується для отримання зображення з камери. На рисунку 3.6 приведений цей клас.

```
class VideoCapture:

    def __init__(self, src=0):
        self.stream = cv2.VideoCapture(src)
        (self.grabbed, self.frame) = self.stream.read()
        self.stopped = False

    def start(self):
        Thread(target=self.get, args=()).start()
        return self

    def get(self):
        while not self.stopped:
            if not self.grabbed:
                self.stop()
            else:
                (self.grabbed, self.frame) = self.stream.read()

    def stop(self):
        self.stopped = True
        self.stream.release()
```

Рисунок 3.6 – Клас VideoCapture

Для того, щоб додати підтримку будь-якої кількості зображень було реалізовано метод, який створює необхідну кількість об`єктів VideoCapture для подальшого отримання зображень з камер. На рисунку 3.7 приведений метод, який створює та налаштовує екземпляри класу VideoCapture.

```
45  def initVideoCaptures(cameraIDs=[]):
46      print('[INFO] Initializing VideoCapture objects ...')
47      for index, cameraID in enumerate(cameraIDs):
48          videoCaptures.append(VideoCapture(cameraID).start())
```

Рисунок 3.7 – Метод для створення об`єктів класу VideoCapture

На наступному етапі виникає проблема обробки зображення з камер для подальшого пошуку об`єктів на ньому. Було розглянуто два варіанти обробки зображення.

У першому варіанті зображення з кожної камери оброблялося окремою моделлю для пошуку об`єктів. Такий варіант реалізації не є доцільним, бо зі збільшенням кількості камер збільшиться і кількість моделей, а швидкодія буде погіршуватися через збільшення навантаження.

У другому варіанті зображення з усіх камер склеюється у панорамне і після цього отримане зображення обробляється моделлю для пошуку об`єктів. В цьому випадку алгоритм, який присутній в моделі, обробляє лише одне зображення і тим самим швидкодія не страждає. Отже, було вирішено реалізовувати цей варіант.

Для реалізації функціоналу склеювання було реалізовано клас Stitcher в якому є метод stitch. Саме цей метод і виконує склеювання двох зображень. На рисунку 3.8 приведений клас Stitcher.

```
6   class Stitcher:
7
8       def __init__(self):
9           self.cachedH = None
10
11      def stitch(self, imageA, imageB, ratio=0.75, reprojThresh=4.0):
12          if self.cachedH is None:
13              (kpsA, featuresA) = self.detectAndDescribe(imageA)
14              (kpsB, featuresB) = self.detectAndDescribe(imageB)
15              M = self.matchKeypoints(kpsA, kpsB,
16                  featuresA, featuresB, ratio, reprojThresh)
17              if M is None:
18                  return None
19              self.cachedH = M[1]
20          result = cv2.warpPerspective(imageA, self.cachedH,
21              (imageA.shape[1] + imageB.shape[1], imageA.shape[0]))
22          result[0:imageB.shape[0], 0:imageB.shape[1]] = imageB
23          return result
```

Рисунок 3.8 – Клас Stitcher

Так як система багатозонова, то необхідно склеїти зображення з двох чи більше камер. Для цієї цілі було створено метод, який склеює зображення з декількох камер. На рисунку 3.9 приведений метод, для створення панорамного зображення.

```
58  def stitchVideoFrames(videoCaptures=[], stitchers=[], camerasOrder=[], videoFramePoints=[]):
59      rightVideoFrameIndex = 0
60      leftVideoFrameIndex = 1
61      result = None
62      for stitcher in stitchers:
63          if result is None:
64              rightVideoFrame = videoCaptures[camerasOrder[rightVideoFrameIndex]].frame
65              rightVideoFrame = cv2.resize(rightVideoFrame, videoFramePoints, interpolation= cv2.INTER_LINEAR)
66          else:
67              rightVideoFrame = result
68          leftVideoFrame = videoCaptures[camerasOrder[leftVideoFrameIndex]].frame
69          leftVideoFrame = cv2.resize(leftVideoFrame, videoFramePoints, interpolation= cv2.INTER_LINEAR)
70          result = stitcher.stitch(rightVideoFrame, leftVideoFrame)
71          leftVideoFrameIndex = leftVideoFrameIndex + 1
72      return result
```

Рисунок 3.9 – Метод stitchVideoFrames

Після створення панорамного зображення необхідно виконати пошук об`єктів. За для цього використовується модель, яка представляє собою алгоритм пошуку об’єктів але для її роботи необхідно не тільки зображення, а і ваги, які описують об`єкти. Ваги було отримано на попередньому етапі після навчання нейронної мережі. На рисунку 3.10 приведений метод startSystem в якому відбувається пошук об`єктів на зображенні за допомогою моделі (строчка коду 90).

```
74  def startSystem(cameraIDs=[],camerasOrder=[],debug=True):
75      assert len(cameraIDs) >= 2, 'Not enough cameras (less than 2)'
76      videoFramewidth = 1280
77      videoFrameheight = 960
78      videoFramePoints = (videoFramewidth, videoFrameheight)
79      initVideoCaptures(cameraIDs)
80      initStitchers(cameraIDs)
81      yoloModel = torch.hub.load('ultralytics/yolov5', 'custom', path='experiments/exp22/weights/last.pt', force_reload=True)
82      while True:
83          if (cv2.waitKey(1) == ord("q")):
84              stopVideoCaptures()
85              break
86          stitchedResult = stitchVideoFrames(videoCaptures, stitchers, camerasOrder, videoFramePoints)
87          if stitchedResult is None:
88              print("[INFO] Homography could not be computed")
89              break
90          result = yoloModel(stitchedResult)
91          if debug:
92              showRowVideos(videoCaptures, videoFramePoints)
93          else:
94              cv2.imshow("Result", np.squeeze(result.render()))
```

Рисунок 3.10 – Метод startSystem

Алгоритм роботи розробленого програмного додатку зображено на рисунку 3.11.

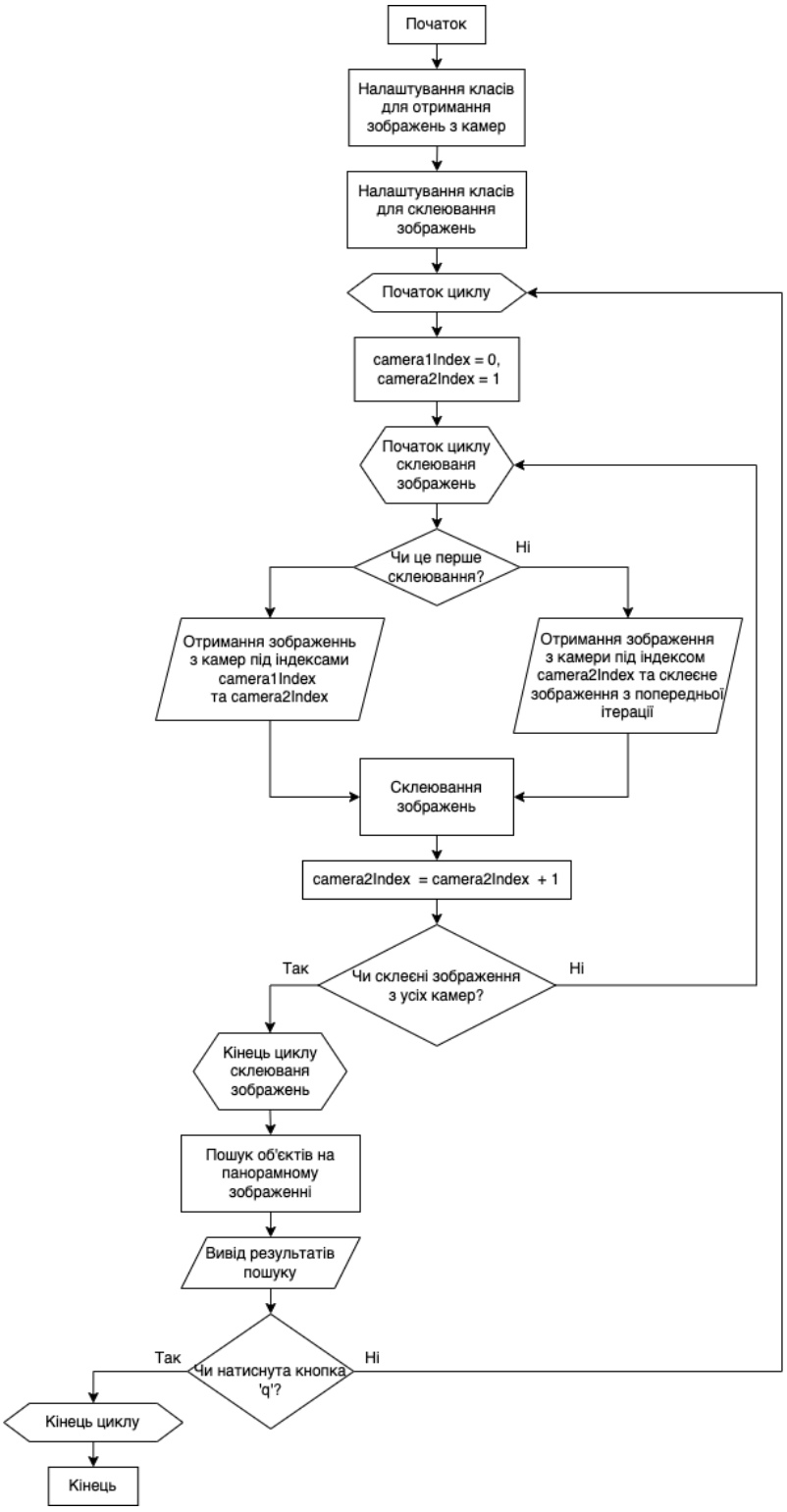

Рисунок 3.11 – Алгоритм роботи розробленого програмного додатку

3.4 Експерименти з перевірки роботи розробленої системи

Для того, щоб провести експерименти, щодо можливостей розпізнавання та ідентифікації розробленої системи, було обрано три фотографії. Кожна фотографія містить один клас об`єктів. Розмір області на зображенні із артилерійським снарядом становить 400 на 600 пікселів (ширина та висота відповідно). На рисунку 3.12 приведений експеримент із розпізнавання та ідентифікації артилерійського снаряду на відстані 15 см.

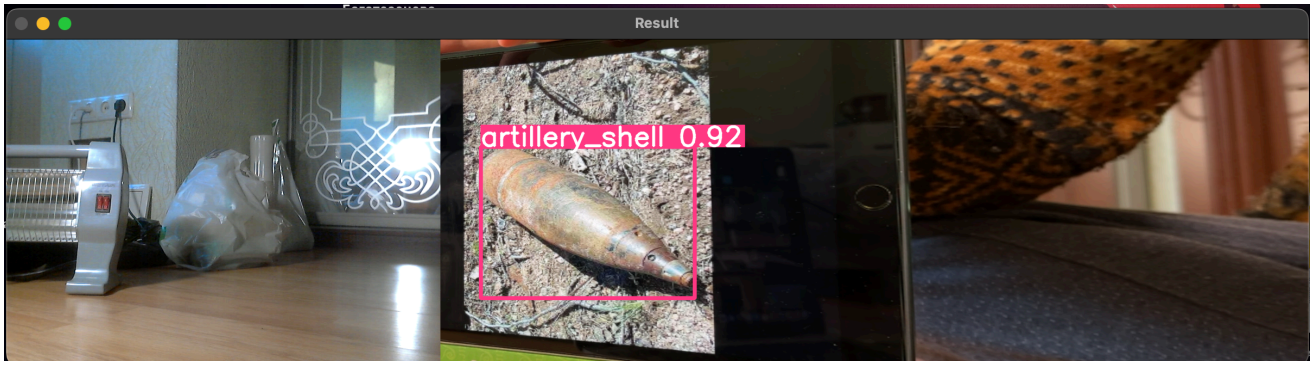

Рисунок 3.12 – Експеримент із розпізнавання та ідентифікації артилерійського снаряду

У таблиці 3.1 наведено результати проведення експериментів із розпізнавання та ідентифікації артилерійських снарядів.

Таблиця 3.1 – Дослідження максимальної відстані розпізнавання та ідентифікації артилерійських снарядів

Номер експерименту	Відстань від камери до об`єкту (см)	Відсоток розпізнавання
1	2	3
1	15	0,92

Продовження таблиці 3.1

1	2	3
2	30	0,90
3	35	0,85
4	40	0,82
5	45	0,74
6	50	0,67
7	55	0,55
8	60	0

Отже, після проведення серії експериментів із розпізнавання та ідентифікації артилерійських снарядів було визначено, що мінімальна відстань від камери до об`єкту становить 55 см. Якщо відстань більше, то система майже не розпізнає об`єкт.

Далі було проведено експерименти із зображенням протитанкової міни. Розмір області на зображенні із протитанковою міною становить 420 на 470 пікселів (ширина та висота відповідно). На рисунку 3.13 приведений експеримент із розпізнавання та ідентифікації протитанкової міни на відстані 15 см.

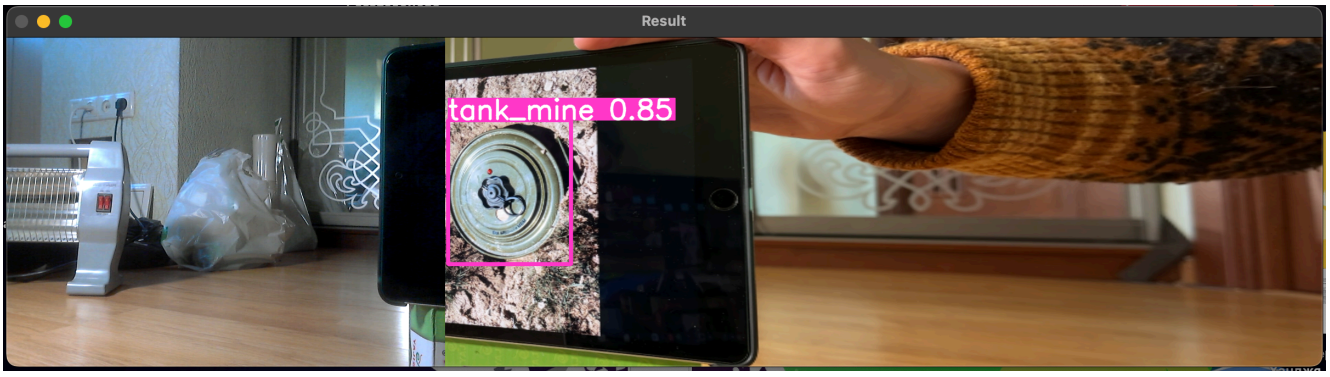

Рисунок 3.13 – Експеримент із розпізнавання та ідентифікації протитанкової міни

У таблиці 3.2 наведено результати проведення експериментів із розпізнавання та ідентифікації протитанкової міни.

Таблиця 3.2 – Дослідження максимальної відстані розпізнавання та ідентифікації протитанкової міни

Номер експерименту	Відстань від камери до об`єкту (см)	Відсоток розпізнавання
1	15	0,85
2	30	0,82
3	35	0,77
4	40	0,71
5	45	0,63
6	50	0

Отже, після проведення серії експериментів із розпізнавання та ідентифікації протипіхотної міни було визначено, що мінімальна відстань від камери до об`єкту становить 45 см. Якщо відстань більше, то система майже не розпізнає об`єкт.

Далі було проведено експерименти із зображенням протипіхотної міни. Розмір області на зображенні із протипіхотною міною становить 500 на 400 пікселів (ширина та висота відповідно). На рисунку 3.14 приведений експеримент із розпізнавання та ідентифікації протипіхотної міни на відстані 15 см.

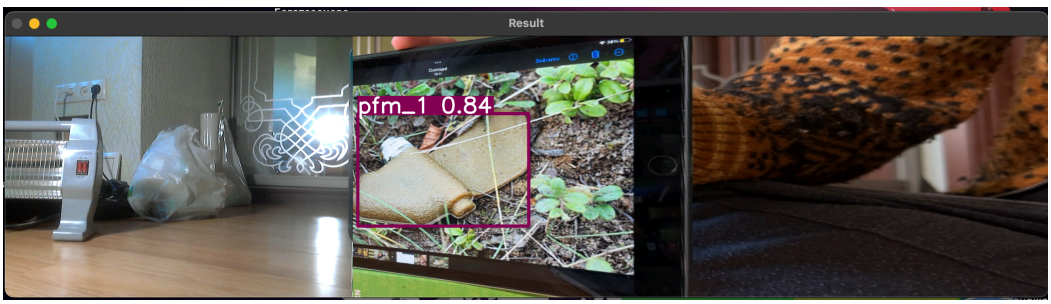

Рисунок 3.14 – Експеримент із розпізнавання та ідентифікації протипіхотної міни

У таблиці 3.3 наведено результати проведення експериментів із розпізнавання та ідентифікації протипіхотної міни.

Таблиця 3.3 – Дослідження максимальної відстані розпізнавання та ідентифікації протипіхотної міни

Номер експерименту	Відстань від камери до об`єкту (см)	Відсоток розпізнавання
1	15	0,84
2	30	0,81
3	35	0,75
4	40	0,69
5	45	0,57
6	50	0,52
7	55	0

Отже, після проведення серії експериментів із розпізнавання та ідентифікації протипіхотної міни було визначено, що мінімальна відстань від камери до об`єкту становить 50 см. Якщо відстань більше, то система майже не розпізнає об’єкт.

Далі були проведені експерименти з перевірки можливості системи розпізнавати одразу декілька об’єктів.

Для проведення першого експерименту було вирішено використовувати чотири об’єкти одно класу. На рисунку 3.15 приведений результат першого експерименту.

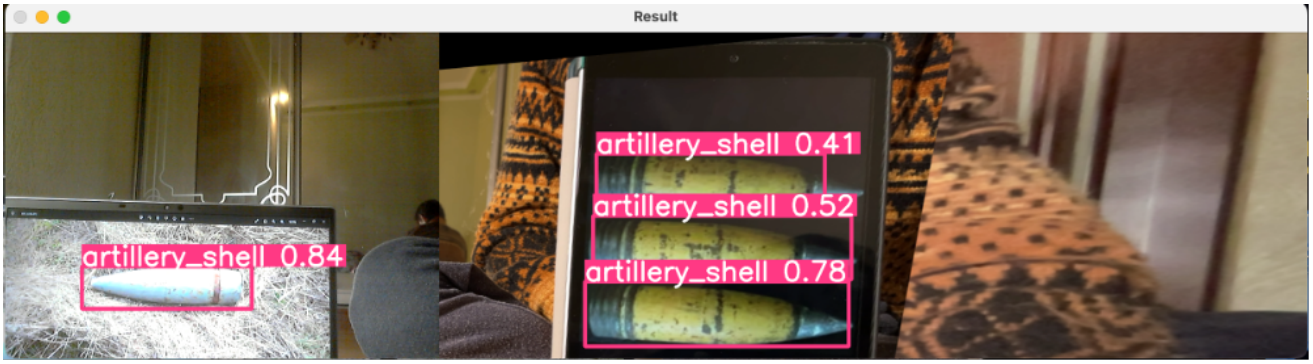

Рисунок 3.15 – Перший експеримент з перевірки можливості системи розпізнавати одразу декілька об’єктів

Як можна побачити з результату першого експерименту, система може розпізнавати одразу декілька об’єктів одного класу.

Для проведення другого експерименту використовуються два об’єкти різного класу. На рисунку 3.16 приведений результат другого експерименту.

Рисунок 3.16 – Другий експеримент з перевірки можливості системи розпізнавати одразу декілька об’єктів

Як можна побачити з результату другого експерименту, система може розпізнавати одразу декілька об’єктів різного класу.

3.5 Висновки

У третьому розділі кваліфікаційної роботи магістранта була розглянута мова програмування Python та біблиотека комп’ютерного зору OpenCV. Було описано алгоритм роботи програми. Також була реалізована структуру бортової багатомовної системи. Було написано програмне забезпечення для роботи бортової багатозонової системи комп’ютерного зору.

Було проведено низьку експерементів для визначення максимальної відстані об’єкту від камери бортової багатозонової системи комп’ютерного зору (результати експериментів наведено у таблицях 3.1-3.3). Отже максимальна відстань від об’єкту до камери становить 55, 45, 50 см для артилерійського снаряду, протитанкової міни, протипіхотної міни відповідно.

Було проведено низьку експерементів для перевірки можливості розробленої системи розпізнавати та ідентифікувати одразу декілька об’єктів різного класу. Як показали результати експериментів, розроблена система може одразу розпізнавати та ідентифікувати декілька об’єктів різного класу.

4 ОХОРОНА ПРАЦІ

4.1 Загальні положення

Загальні положення [32]:

– інструкція з охорони праці встановлює вимоги щодо безпеки діяльності інженера-програміста під час виконання його службових обов’язки;

– інструкція з охорони праці для інженера-програміста поширюється на всі кабінети та приміщення;

– інструкція встановлює порядок безпечного ведення робіт інженером програмістом у приміщеннях, на території технікуму та інших місцях, де він виконує свої посадові обов’язки;

– інструкція є обов'язковою для виконання інженером-програмістом вимог з питань охорони праці відповідно до Закону України «Про охорону праці» і Кодексу законів про працю України;

– інженер-програміст призначається з числа осіб, які мають вищу освіту, за станом здоров'я можуть виконувати відповідний вид роботи та пройшли навчання і відповідні інструктажі з питань охорони праці, безпеки життєдіяльності;

– перед призначенням на роботу періодично, один раз на рік, інженерпрограміст повинен проходити медичний огляд;

– інженер-програміст один раз на три роки проходить навчання з питань охорони праці, безпеки життєдіяльності з наступною перевіркою знань;

– інженер з охорони праці проводить із працівником,який приймається на роботу, вступний інструктаж з охорони праці,знайомить із правилами внутрішнього розпорядку, із заходами щодо забезпечення належного рівня пожежної безпеки, санітарними правилами улаштування і утримання приміщення;

– інженер-програміст повинен знати правила пожежної безпеки і вміти користуватися первинними засобами пожежогасіння (вогнегасниками);

– інженер-програміст повинен мати навички надання першої (долікарської) допомоги;

– про виявлені несправності обладнання, устаткування, пристроїв, інші небезпечні прояви та нещасні випадки, які трапилися в приміщенні 3 технікуму, інженер-програміст повинен повідомити директора або особу, яка його заміняє.

4.2 Вимоги безпеки перед початком роботи

Вимоги безпеки перед початком роботи [32]:

– оглянути своє робоче місце з метою виявлення небезпечних для життя та здоров'я факторів;

– у разі виявлення порушень або несправностей вжити заходів щодо їх усунення, а за необхідності – повідомити директора або інженера з охорони праці.

4.3 Вимоги безпеки під час виконання роботи

Вимоги безпеки під час виконання роботи [32]:

– виконувати роботу згідно зі своїми функціональними обов'язками;

– не залишати без нагляду своє робоче місце, коли обладнання підключено до електромережі (комп'ютер, електроприлади тощо);

– вимоги до безпеки після закінчення роботи;

– перевірити своє робоче місце;

– відключити від електромережі електрообладнання;

– закрити вікна, погасити світло;

– про виявлені недоліки повідомити директора або особу, що його заміняє.

4.4 Вимоги до безпеки в аварійних ситуаціях

Вимоги до безпеки в аварійних ситуаціях [32]:

– при виявлені небезпечної ситуації (пожежа, землетрус, радіаційна небезпека, неполадки в електрогосподарстві тощо) для його життя та життя співробітників заспокоїтись і заспокоїти оточуючих. Оцінити важкість аварійної ситуації;

– не усувати самому несправностей електромережі та електрообладнання, а вимкнути загальне електропостачання;

– при виявленні пожежі негайно викликати пожежну частину за телефоном 101, повідомити заступника директора з адміністративно-господарської роботи або директора;

– вжити заходів згідно з планом евакуації на випадок пожежі та вивести людей у безпечне місце. Організувати роботу щодо збереження державного майна та цінних паперів;

– при появі сторонньої особи, яка застосовує протиправні дії щодо його безпеки або оточуючих, викликати міліцію за телефоном 102;

– у випадку травмування себе, або інших працівників необхідно звернутися до медичного працівника, викликати швидку допомогу 103 або за необхідності надати першу долікарську допомогу, повідомити заступника директора з навчальної роботи;

– надання першої медичної допомоги треба починати з оцінки загального стану потерпілого і на підставі цього скласти думку про характер 4 пошкодження. У разі різкого порушення або відсутності дихання, зупинки серця негайно приступити до проведення штучного дихання та зовнішнього масажу серця, негайно викликати за телефоном 103 швидку медичну допомогу.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи магістранта було проведено аналіз існуючих систем комп'ютерного зору в області виявлення вибухонебезпечних об’єктів. Було розроблено архітектуру бортової багатозонової системи. Також було розроблено бортову багатозонову систему. Розроблена бортова багатозонова система має підртимку великої кількості камер і має можливість розпізнавати та ідентифікувати артилерійські снаряди, протитанкову міну ТМ-62М, протипіхотну міну ПФМ-1. Отриману бортову багатозонову систему було перевірено на достовірність роботи. Крім цього були розглянуті питання з охорони праці, яких повинен дотримуватися інженер-програміст під час підготовки, виконання роботи та в аварійних ситуаціях.

Усі цілі, що були поставлені на початку роботи було виконано.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Цимбал О. М., Каплін О. П. Розроблення бортової багатозонової системи комп’ютерного зору із функціями розпізнавання та ідентифікації // Виробництво & Мехатронні Системи 2022 : матеріали VI Міжнародної конференції. Харків, 2022. С. 61-63.

2. ДСТУ 3008: 2015. Інформація та документація. Звіти у сфері науки і техніки. Структура і правила оформлення. К.: ДП “УкрНДНЦ”. 2016. 30 с.

3. Методичні вказівки з підготовки та захисту кваліфікаційної роботи здобувачами другого (магістерського) рівня вищої освіти спеціальності 151 Автоматизація та комп’ютерно-інтегровані технології, освітньо-професійних програм: «Автоматизоване управління технологічними процесами», «Комп’ютерно-інтегровані технологічні процеси і виробництва», «Комп’ютеризовані та робототехнічні системи» / Упоряд. І. Ш. Невлюдов, Р. В. Артюх, В. В. Безкоровайний, Н. П. Демська, В. В. Євсєєв, О. І. Филипенко, О. М. Цимбал. Харків: ХНУРЕ, 2021. 55 с.

4. Комп’ютерний зір в технологіях доповненої реальності та його основні напрями використання. // Прикладні аспекти сучасних міждисциплінарних досліджень : матеріали наук.-практ. конф. (м. Вінниця, 26 листоп. 2021 р.). Вінниця, 2021. С. 200.

5. Цимбал О.М. Системи адаптації роботів та технологія OpenCV / О.М. Цимбал, А.І. Бронніков – Харків, ХНУРЕ, 2019. – 134 с.

6. Голубков П., Путников Д., Егоров В. Комп’ютерний зір у вирішенні проблеми розпізнавання форми кубічного пельменя // Автоматизація технологічних і бізнес-процесів: матеріали XI Міжнародної конференції. Одеса, 2019. С. 51-61.

7. Аналіз методів розпізнавання об’єктів на зображенні із застосуванням бібліотеки OPENCV. Комп’ютерні технології: інновації, проблеми, рішення : матеріали II Всеукр. науково-техн. конф., Житомир, 2019. С. 172.

8. Потьомкіна К. О. Згорткові нейронні мережі як основа машинного навчання / К. О. Потьомкіна // Радіоелектроніка та молодь в XXI столітті : матеріали 24 Міжнар. молодіж. форуму, 7-9 квіт. 2020 р. – Харків : ХНУРЕ, 2020. – Т. 5. – С. 149-150.

9. Глибокі нейронні мережі для вирішення завдань розпізнавання і класифікації зображення // Інформаційні технології та комп'ютерне моделювання : матеріали ст. Міжнар. науково-практ. конф. (м. Івано-Франківськ, 15 трав. 2019 р.). Івано-Франківськ, 2019. С. 464.

10. Нейромережі GAN у створенні нових моделей // Комп’ютерні засоби, мережі та системи : зб. наук. пр., (м. Київ, 18 листоп. 2019 р.). Київ, 2019. С. 119.

11. Deep Learning for Vision Systems 1st Edition: Comprehensive Guide. New York, 2020, 480 p.

12. Бондар І.Б. Нейромережне розпізнаваня об’єктів : текстова частина до курсової роботи. Київ, 2020. С. 25.

13. Detect industrial defects at low latency with computer vision at the edge with Amazon SageMaker Edge. URL: https://aws.amazon.com/ru/blogs/machine-learning/detect-industrial-defects-at-low-latency-with-computer-vision-at-the-edge-with-amazon-sagemaker-edge/ (дата звернення: 17.09.2022)

14. Project InnerEye – Democratizing Medical Imaging AI. URL: https://www.microsoft.com/en-us/research/project/medical-image-analysis/ (дата звернення: 18.09.2022)

15. X-Sight – New AI And AR Pilot Helmet. URL: https://i-hls.com/archives/115031 (дата звернення: 19.09.2022)

16. Intel RealSense Depth Camera D435. URL: https://evo.net.ua/intel-realsense-depth-camera-d435/?gclid=CjwKCAjws--ZBhAXEiwAv-

RNL119Pq2nfKswgwBgtUAe5flQjgXd4mTdmOX6DiQwMCbJycHHczLV3hoCPWEQAvD_BwE\ (дата звернення: 20.09.2022)

17. SOLOSHOT3 with Optic65 Camera [Електронний ресурс]. URL: https://soloshot.com/collections/soloshot3 (дата звернення: 24.09.2022)

18. Andros FX. URL: https://www.peraton.com/wp-content/uploads/2020/10/2021_DS_PERATONREMOTEC_ANDROS_FX.pdf (дата звернення: 26.09.2022)

19. Andros Titus. URL: https://www.militarysystems-tech.com/sites/militarysystems/files/supplier_docs//Andros%20Titus.pdf (дата звернення: 27.09.2022)

20. Andros F6B. URL: https://www.militarysystems-tech.com/sites/militarysystems/files/supplier_docs//Andros%20F6.pdf (дата звернення: 28.09.2022)

21. OpenCV – Overview. URL: https://www.geeksforgeeks.org/opencv-overview/ (дата звернення: 29.09.2022)

22. Афанасьєва К. О. Дослідження механізмів бібліотеки комп'ютерного зору OpenCV для розробки мобільних додатків для Android OS // Комп'ютерні ігри та мультимедіа як інноваційний підхід до комунікації : матеріали I Всеукр. наук.-техн. конф. молодих вчених, аспірантів та студентів (Одеса, 25–26 берез. 2021 р.). Одеса, 2021. С. 84–86.

23. Чуприна А. С., Магда М. А. Огляд використання нейронних мереж у задачі виявлення об’єктів // Public communication in science: philosophical, cultural, political, economic and it context : матеріали Міжнародної науково-практичної конференції. Х'юстон, 2020. С. 73-75.

24. YOLO V3 Explained. URL: https://towardsdatascience.com/yolo-v3-explained-ff5b850390f (дата звернення: 15.10.2022)

25. Object detection using YOLO: challenges, architectural successors, datasets and applications. URL: https://link.springer.com/content/pdf/10.1007/s11042-022-13644-y.pdf (дата звернення: 03.11.2022)

26. Suraj Beera, Manish Chembeti, Naren Hrithik, Ashish Chitturi «The Yolo V5 Based Smart Cellphone Detector» // Nat. Volatiles & Essent: 2021.

27. Мірчук Н. П. Система трекінгу відвідувачів : дипломний проект. Київ, 2020. С. 42.

28. Марценюк Є.О. Комп`ютерна дискретна математика / навчальний посібник. Тернопіль, 2020, 53 с.

29. Цимбал О. М., Каплін О. П. Розробка бортової багатозонової системи комп'ютерного зору із функціями розпізнавання та ідентифікації // ГО «Молодіжна наукова ліга»: матеріали II Міжнародної студентської наукової конференції (Т.3), Мукачево, 2021. С. 36-37.

30. Introduction To PYTHON. URL: https://www.geeksforgeeks.org/introduction-to-python/ (дата звернення: 17.11.2022)

31. Top 10 Features of Python You Must Know. URL: https://www.interviewbit.com/blog/features-of-python/ (дата звернення: 20.11.2022)

32. ІНСТРУКЦІЯ З ОХОРОНИ ПРАЦІ № 22-ОП-18 для інженера-програміста. URL: https://ddteit.dp.ua/document/P_1_25.pdf (дата звернення: 25.11.2022)