

Я, як студент ХНУРЕ, розумію і підтримую політику закладу із академічної доброчесності. Я не надавав та не одержував недозволену допомогу під час підготовки кваліфікаційної роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

14.06.2024



Шахов П.В.

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Факультет _____ АКТ
Кафедра _____ КІТАР
Рівень вищої освіти _____ перший(бакалаврський)
Спеціальність _____ 151 Автоматизація та комп'ютерно-інтегровані технології
Тип програми _____ Освітньо-професійна
Освітня програма _____ Автоматизація та комп'ютерно-інтегровані технології
(шифр і назва)

ЗАТВЕРДЖУЮ:

Зав.кафедри _____
(підпис)

«__» _____ 2024р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

студентові _____ Шахову Павлу Володимировичу
(шифр і назва)

1. Тема роботи: _____ Розроблення системи автоматизації для відстеження
об'єктів у робочій зоні мобільного робота

Затверджена наказом університету від _____ 03.06.2024 №544Ст

2. Термін подання студентом роботи до екзаменаційної комісії _____ 14.06.2024р.

3. Вихідні дані до роботи: 3.1 Розробка НМІ інтерфейсу користувача

4. Перелік питань, що потрібно опрацювати в роботі: 4.1 Вступ; 4.2 Аналіз систем комп'ютерного зору; 4.3 Аналіз методів відстеження об'єктів; 4.4 Аналіз алгоритмів роботи систем комп'ютерного зору; 4.5 Аналіз аналогічних рішень; 4.6 Розробка структури програми; 4.7 Обґрунтування та вибір середовища розробки; 4.8 Математичне представлення методу; 4.9 Розробка загального алгоритму роботи програми; 4.10 Розробка алгоритму відстеження об'єктів; 4.11 Розрахунок швидкості обробки кадрів при відстеження об'єктів; 4.12 Розробка НМІ інтерфейсу користувача; 4.13 Реалізація функцій відстеження об'єктів; 4.14 Проведення експерименту; 4.15 Висновки

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій Графічний демонстраційний матеріал в форматі PowerPoint(*.ppt) формату А4 –15 сторінок.

6. Консультанти розділів роботи

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

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз систем комп'ютерного зору	18.04.2024-20.04.2024	виконано
2	Аналіз методів відстеження об'єктів	21.04.2024-24.04.2024	виконано
3	Аналіз алгоритмів роботи систем комп'ютерного зору	25.04.2024-29.04.2024	виконано
4	Аналіз аналогічних рішень	30.04.2024-01.05.2024	виконано
5	Розробка структури програми	02.05.2024-05.05.2024	виконано
6	Обґрунтування та вибір середовища розробки	06.05.2024-08.05.2024	виконано
7	Математичне представлення методу	09.05.2024-12.05.2024	виконано
8	Розробка загального алгоритму роботи програми	13.05.2024-14.05.2024	виконано
9	Розробка алгоритму відстеження об'єктів	15.05.2024-17.05.2024	виконано
10	Розрахунок швидкості обробки кадрів при відстеження об'єктів	18.05.2024-22.05.2024	виконано
11	Розробка НМІ інтерфейсу користувача	23.05.2024-28.05.2024	виконано
12	Реалізація функцій відстеження об'єктів	29.05.2024-01.06.2024	виконано
13	Проведення експерименту	02.06.2024-03.06.2024	виконано

Дата видачі завдання 18 квітня 2024р.

Студент  (підпис)

Шахов П.В.
(прізвище, ініціали)

Керівник роботи _____
(підпис)

Омаров Ш.А.
(прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 70 с., 5 табл., 9 рис., 2 дод., 14 джерел.

ВІДСТЕЖЕННЯ ОБ'ЄКТІВ, РОБОЧА ЗОНА МОБІЛЬНОГО РОБОТА,
ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, ТРАЄКТОРІЯ ПЕРЕМІЩЕННЯ,
INDUSTRY 5.0.

Мета роботи – зменшення вигоди пошкодження об'єктів в
робочій зоні мобільного робота за рахунок розробки програмного
забезпечення відстеження об'єктів.

Об'єкт роботи – процес розпізнавання об'єктів у робочій зоні
мобільного робота.

Предмет роботи – методи, алгоритмічне та програмне забезпечення
розпізнавання знаходження об'єктів у робочій зоні мобільних роботів у
рамках концепції Industry 5.0.

В даній кваліфікаційній роботі проведено аналіз систем комп'ютерного
зору, методів відстеження об'єктів у робочій зоні мобільного
робота, проаналізовано алгоритми роботи систем комп'ютерного зору та
аналогічні рішення.

Після проведеного аналізу було розроблено структуру програми.
Проведено обґрунтування та вибір середовища розробки. Проведено
математичне представлення методу відстеження об'єктів у робочій зоні
мобільного робота. Розроблено загальний алгоритм роботи програми та
алгоритм відстеження об'єктів. Проведено розрахунок швидкості обробки
кадрів при відстеження об'єктів.

Розроблено НМІ інтерфейс користувача та реалізовано функції
відстеження об'єктів. Проведено експериментальні дослідження.

ABSTRACT

Explanatory note: 70 pages, 5 tables, 9 figures, 2 app, 14 sources.

OBJECT TRACKING, MOBILE ROBOT WORK AREA, SOFTWARE, MOVEMENT TRAJECTORY, INDUSTRY 5.0.

The purpose of the work is reducing the probability of damage to objects in the working area of the mobile robot due to the development of object tracking software.

Work object – the process of recognizing objects in the work area of a mobile robot.

The subject of the work is methods, algorithmic and software for recognizing the location of objects in the working area of mobile robots within the framework of the Industry 5.0 concept.

In this qualification work, an analysis of computer vision systems, methods of tracking objects in the working area of mobile

work, algorithms of computer vision systems and similar solutions were analyzed.

After the analysis, the structure of the program was developed. The justification and selection of the development environment was carried out. A mathematical presentation of the method of tracking objects in the working area of a mobile robot is carried out. The general algorithm of the program and the object tracking algorithm have been developed. The calculation of frame processing speed during object tracking was carried out.

HMI user interface was developed and object tracking functions were implemented. Experimental studies have been conducted.

ЗМІСТ

Перелік умовних скорочень	9
Вступ.....	10
1 Аналіз сучасних систем відстеження об'єктів у робочій зоні мобільного робота	12
1.1 Аналіз систем комп'ютерного зору.....	12
1.2 Аналіз методів відстеження об'єктів у робочій зоні мобільного робота	17
1.3 Аналіз алгоритмів роботи систем комп'ютерного зору.....	22
1.4 Аналіз аналогічних рішень.....	23
2 Розробка алгоритму програми відстеження об'єктів у робочій зоні мобільного робота.....	27
2.1 Розробка структури програми.....	27
2.2 Обґрунтування та вибір середовища розробки.....	29
2.3 Математичне представлення методу відстеження об'єктів у робочій зоні мобільного робота	31
2.4 Розробка загального алгоритму роботи програми.....	34
2.5 Розробка алгоритму відстеження об'єктів.....	39
2.6 Розрахунок швидкості обробки кадрів при відстеження об'єктів.....	44
3 Розробка програми для відстеження об'єктів у робочій зоні мобільного робота	47
3.1 Розробка НМІ інтерфейсу користувача	47
3.2 Реалізація функцій відстеження об'єктів	50
3.3 Проведення експерименту	55
3.4 Охорона праці.....	59
Висновки	61
Перелік джерел посилань	62

Додаток А Код програми.....	64
Додаток Б Демонстраційний матеріал	69

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

CNN – згорткові нейронні мережі;

HOG – гістограми орієнтованих градієнтів;

LBP – локальні бінарні шаблони;

SLAM – системи локалізації та мапування;

SVM – класифікатори на основі векторів підтримки.

ВСТУП

Розробка програми відстеження об'єктів у робочій зоні мобільного робота має велику актуальність у сучасному технологічному світі. Зростання зацікавленості в автономних системах та роботах спричиняє необхідність розробки програмного забезпечення, яке дозволить роботам ефективно взаємодіяти з оточенням та виконувати різноманітні завдання.

Відстеження об'єктів у робочій зоні мобільного робота дозволяє виявляти, відслідковувати та реагувати на різні об'єкти, такі як люди, тварини, автомобілі чи перешкоди, що можуть виникати під час руху робота. Це може бути важливо для безпеки, навігації, виявлення потенційних загроз та виконання завдань у різних сферах, від промисловості та складської логістики до медичних та рятувальних операцій.

З використанням передових технологій комп'ютерного зору, таких як машинне навчання та глибоке навчання, розробка програм відстеження об'єктів стає ще більш ефективною та точною. Вона дозволяє роботам адаптуватися до різних умов та змін у оточенні, що є важливим для їхньої успішної роботи в реальному світі. Таким чином, розробка програм відстеження об'єктів у робочій зоні мобільного робота є актуальним та перспективним напрямом досліджень і розробок.

Мета роботи – зменшення вимог до пошкодження об'єктів в робочій зоні мобільного робота за рахунок розробки програмного забезпечення відстеження об'єктів.

Об'єкт роботи – процес розпізнавання об'єктів у робочій зоні мобільного робота.

Предмет роботи – методи, алгоритмічне та програмне забезпечення розпізнавання знаходження об'єктів у робочій зоні мобільних роботів у рамках концепції Industry 5.0.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- провести аналіз систем комп'ютерного зору;
- провести аналіз методів відстеження об'єктів у робочій зоні мобільного робота;
- провести аналіз алгоритмів роботи систем комп'ютерного зору;
- провести аналіз аналогічних рішень;
- розробити структуру програми;
- провести обґрунтування та вибір середовища розробки;
- провести математичне представлення методу відстеження об'єктів у робочій зоні мобільного робота;
- розробити загального алгоритму роботи програми;
- розробити алгоритм відстеження об'єктів;
- провести розрахунок швидкості обробки кадрів при відстеження об'єктів;
- розробити НМІ інтерфейсу користувача;
- реалізувати функції відстеження об'єктів;
- провести експеримент.

Дана кваліфікаційна робота була виконана згідно ДСТУ 3008 – 15 [1], а також, керуючись навчальним посібником з дипломного проекту [2] та методичними вказівками [3].

1 АНАЛІЗ СУЧАСНИХ СИСТЕМ ВІДСТЕЖЕННЯ ОБ'ЄКТІВ У РОБОЧІЙ ЗОНІ МОБІЛЬНОГО РОБОТА

1.1 Аналіз систем комп'ютерного зору

Системи комп'ютерного зору – це технології, що використовуються для аналізу та обробки візуальної інформації, отриманої зі зображень або відео. Вони базуються на використанні алгоритмів та методів обробки зображень, машинного та глибокого навчання для вирішення різноманітних завдань, таких як розпізнавання облич, відстеження об'єктів, виявлення перешкод, визначення руху та багато інших.

Комп'ютерний зір є важливою частиною сфери штучного інтелекту (ШІ), яка використовує просунуті алгоритми для аналізу та інтерпретації зображень і відео. За допомогою цих алгоритмів комп'ютери можуть розпізнавати об'єкти, розуміти сцени, відстежувати рухи та виконувати багато інших завдань, пов'язаних із візуальною інформацією. Це дозволяє створювати системи, які можуть, наприклад, автоматично сортувати фотографії, розпізнавати обличчя або навіть допомагати у водінні автономних транспортних засобів.

Обробка зображень в системах комп'ютерного зору включає в себе різноманітні техніки та алгоритми, що застосовуються для підготовки вхідних даних перед подальшим аналізом. Попередня обробка зображень має на меті покращення якості та видалення непотрібних елементів з вхідних даних. Це може включати в себе операції, такі як зменшення шуму, застосування фільтрів для видалення артефактів, а також підсилення контрасту для покращення різкості зображення. Попередня обробка зображень грає ключову роль у підвищенні ефективності подальшого аналізу та розпізнавання об'єктів.

Процес роботи комп'ютерного зору включає кілька ключових етапів, кожен з яких спрямований на перетворення візуальної інформації у зрозумілі дані для аналізу і прийняття рішень. На першому етапі збираються візуальні дані у вигляді зображень або відео за допомогою камер або інших сенсорів. Ці дані можуть бути отримані в реальному часі або зберігатися для подальшої обробки. На етапі попередньої обробки зібрані дані очищаються і готуються до аналізу. Це може включати фільтрацію шумів, корекцію кольору, нормалізацію яскравості і контрасту, а також зміну розміру зображень. Далі алгоритми виділяють ключові ознаки з зображень, такі як контури, текстури, кольори або певні форми. Це допомагає створити узагальнене представлення об'єктів на зображенні. Використовуючи виділені ознаки, система застосовує алгоритми машинного навчання або глибокого навчання для розпізнавання об'єктів, визначення їх положення та класифікації. На цьому етапі можуть бути використані нейронні мережі, такі як згорткові нейронні мережі (CNN). Після розпізнавання система виконує подальший аналіз, щоб інтерпретувати результати у зрозумілій формі. Наприклад, визначається, які об'єкти присутні на зображенні, їх розташування та взаємодія між ними. Нарешті, результати аналізу можуть бути використані для прийняття рішень або виконання дій. Це може включати автоматичне керування транспортними засобами, сортування об'єктів, видачу рекомендацій або навіть взаємодію з користувачем.

Сегментація – це важливий етап обробки зображень, де зображення розділяється на окремі області або сегменти. Це допомагає ізолювати об'єкти або регіони інтересу на зображенні для подальшого аналізу. Існують різні методи сегментації, включаючи порогову сегментацію, розділення за кольором, різні методи розділення пікселів та групування пікселів за їхніми характеристиками. Для складних зображень можуть використовуватися комбінації цих методів для досягнення більш точних результатів.

Вилучення ознак є ще одним важливим етапом обробки зображень, де визначаються та виділяються характеристики об'єктів, які можуть бути використані для їхньої ідентифікації або класифікації. Це може включати в себе визначення форми, текстур, розміру та інших атрибутів об'єктів на зображенні. Вилучення правильних ознак є ключовим для успішного розпізнавання об'єктів та подальшого аналізу зображень у системах комп'ютерного зору [4].

Методи візуального відстеження грають ключову роль у системах комп'ютерного зору, які вимагають відстеження рухомих об'єктів на відеозаписах або в реальному часі. Одним з найпоширеніших методів є одноразове відстеження, де система визначає положення та форму об'єкта на одному кадрі і намагається відстежувати його на послідуєчих кадрах. Цей процес може бути складним через зміни освітлення, перекриття об'єкта іншими об'єктами або зміни його форми. Тому важливо використовувати адаптивні алгоритми, які можуть адекватно реагувати на зміни у вихідних даних.

Багаторазове відстеження є іншим важливим методом, який дозволяє системі слідкувати за об'єктом на послідовності кадрів. Цей процес може включати в себе використання методів відстеження руху, де шлях об'єкта прогнозується на основі його попередніх рухів та динаміки. Крім того, можуть використовуватися методи, що враховують контекстну інформацію, такі як моделі переміщення об'єкта або моделі поведінки.

Незважаючи на те, що методи візуального відстеження відкривають широкі можливості для застосування у різних областях, вони також стикаються з викликами. Наприклад, вони можуть бути чутливими до змін освітлення, тіней або перекриття об'єктів. Також можуть виникати проблеми з відстеженням об'єктів зі схожими характеристиками або об'єктів, які рухаються швидко або непередбачувано. Тому постійний розвиток та

вдосконалення методів візуального відстеження є важливим завданням у дослідженнях у галузі комп'ютерного зору.

Розпізнавання об'єктів є ключовим етапом в системах комп'ютерного зору, який вимагає ідентифікації та класифікації об'єктів на зображеннях або відео. Один з основних аспектів цього процесу – це визначення характеристик об'єктів, таких як форма, розмір, текстура та колір. Для досягнення цієї мети використовуються різноманітні методи, включаючи аналіз форми та текстури пікселів у визначеному регіоні зображення, використання нейронних мереж для автоматичного визначення ознак та використання дескрипторів, таких як гістограми орієнтованих градієнтів (HOG) або локальні бінарні шаблони (LBP).

Класифікація об'єктів визначає їхню приналежність до певних категорій або класів. Для цього застосовуються різні методи машинного навчання, такі як класифікатори на основі векторів підтримки (SVM), нейронні мережі, дерева рішень тощо. Ці алгоритми навчаються на підготовлених наборах даних, де кожен об'єкт має відому категорію, і потім використовуються для класифікації нових об'єктів [5].

Одним з викликів у розпізнаванні об'єктів є здатність системи адаптуватися до різних умов зйомки, таких як зміни освітлення, перспективи, масштабу та затемнення. Для подолання цих викликів можуть застосовуватися методи нормалізації зображень, аугментації даних та використання додаткових алгоритмів для адаптації до змінних умов. Крім того, важливою є робота над збільшенням стійкості та надійності систем розпізнавання об'єктів шляхом постійного вдосконалення алгоритмів та використання новітніх технологій у галузі штучного інтелекту.

Застосування систем комп'ютерного зору охоплюють широкий спектр галузей і відкривають безліч можливостей для автоматизації та покращення різних процесів. У медицині, наприклад, системи комп'ютерного зору використовуються для аналізу медичних зображень, таких як рентгенівські

знімки або знімки з магнітно-резонансної томографії (МРТ), для виявлення патологій та діагностики захворювань. Це дозволяє швидше та точніше встановлювати діагнози та розробляти ефективніші методи лікування для пацієнтів. У сфері автомобільної промисловості системи комп'ютерного зору використовуються для реалізації різноманітних функцій, таких як системи помічника при паркуванні, детекція дорожніх знаків, системи автоматичного гальмування та виявлення пішоходів для підвищення безпеки на дорогах. Ці технології сприяють уникненню аварій та зменшенню кількості травмованих у дорожньому русі. У сфері безпеки системи комп'ютерного зору використовуються для виявлення підозрілих об'єктів або поведінки на відео зйомках у громадських місцях, таких як аеропорти, станції метро чи вулиці міста. Вони можуть автоматично реагувати на потенційні загрози та сприяти забезпеченню громадської безпеки шляхом оперативного сповіщення служб безпеки або автоматичного включення систем безпеки. У сфері виробництва та робототехніки системи комп'ютерного зору використовуються для визначення положення та розпізнавання об'єктів на конвеєрах, візуального навігації роботів, а також для якісного контролю виробництва, що дозволяє забезпечити високу якість продукції та підвищити ефективність виробничих процесів. У наукових дослідженнях системи комп'ютерного зору застосовуються для вивчення поведінки та міграцій тварин, відслідковування змін в природних середовищах, вивчення атмосферних явищ та інших об'єктів, які необхідно спостерігати та аналізувати на великій території або протягом тривалого часу. Це дозволяє дослідникам отримувати нові знання та розуміння про природні явища та біологічні системи.

Впровадження систем комп'ютерного зору стикається з рядом викликів та обмежень, які можуть вплинути на їхню ефективність та надійність. Один з головних викликів – це здатність системи адаптуватися до різних умов зйомки, таких як зміни освітлення, тіні, а також різні кути та відстані зйомки. Відсутність стандартизованих умов може вплинути на якість та точність

роботи системи, тому важливо розробляти алгоритми, які можуть пристосовуватися до змінних умов зйомки та навчатися на динамічних даних [6].

Ще одним важливим викликом є необхідність обробки великої кількості даних в реальному часі. Завдання розпізнавання об'єктів та візуального відстеження може бути дуже обчислювально витратним, особливо при використанні складних алгоритмів та моделей машинного навчання. Це може призводити до затримок та нестабільності системи, що ставить під сумнів її придатність для застосування у реальних умовах.

Крім того, точність та надійність роботи системи також можуть бути обмежені складністю об'єктів та сцен, з якими вона має справу. Наприклад, викликом може бути розпізнавання об'єктів зі схожими характеристиками або об'єктів, які знаходяться в перекритому стані. Також системи можуть виявляти помилки в роботі при обробці складних сцен або при виникненні артефактів на зображеннях, що важко інтерпретувати для алгоритмів комп'ютерного зору.

Для подолання цих викликів і обмежень необхідно постійно вдосконалювати алгоритми та методи обробки зображень, використовувати нові технології та адаптувати їх до конкретних умов застосування. Крім того, необхідно вдосконалювати апаратне забезпечення для забезпечення достатньої продуктивності системи та зменшення часу обробки даних. Тільки шляхом постійного вдосконалення та інновацій можна досягти успіху у впровадженні систем комп'ютерного зору в різні галузі [7].

1.2 Аналіз методів відстеження об'єктів у робочій зоні мобільного робота

Методи відстеження об'єктів у робочій зоні мобільного робота відіграють критичну роль у забезпеченні ефективного та безпечного

функціонування таких систем. При роботі в реальному часі та у різних умовах робочого середовища мобільні роботи повинні бути здатні ефективно відстежувати об'єкти, що пересуваються або інтерактивно взаємодіють з ними.

Основні методи відстеження об'єктів у робочій зоні мобільного робота можна порівняти за різними критеріями, такими як точність, швидкість, вартість і надійність у різних умовах.

Візуальне відстеження за обличчями та рухомими об'єктами – це метод використання систем комп'ютерного зору для виявлення та відстеження облич або рухомих об'єктів на відео або зображеннях. Цей метод зазвичай базується на аналізі піксельних даних, що представляють зображення, з використанням різноманітних алгоритмів та технік обробки зображень. Він здатний відслідковувати обличчя або об'єкти в реальному часі, що робить його корисним для застосування в ситуаціях, де потрібно виявляти або навігувати за об'єктами на відео або у реальному світі.

Один з основних підходів до візуального відстеження за обличчями та рухомими об'єктами – це використання методів розпізнавання образів та відстеження руху. Алгоритми розпізнавання образів можуть ідентифікувати обличчя або об'єкти на зображенні шляхом аналізу їхніх характеристик, таких як форма, текстура та колір. Після ідентифікації об'єктів відстеження руху використовується для слідкування за їхнім рухом та визначення їхньої нової позиції на наступних кадрах.

Одним з основних викликів візуального відстеження за обличчями та рухомими об'єктами є здатність адаптуватися до змін в освітленні, масштабі, позі та обличчях об'єктів. Деякі методи можуть бути чутливими до цих факторів і можуть демонструвати низьку ефективність в умовах, де вони змінюються динамічно. Тому важливо розробляти алгоритми візуального відстеження, які можуть адекватно реагувати на такі зміни та надійно відстежувати обличчя та об'єкти навіть у варіативних умовах зйомки [8].

Системи локалізації та мапування (SLAM) – це технології, що використовуються для одночасної локалізації мобільного робота у просторі та побудови мапи його оточення. Основна мета SLAM полягає в тому, щоб робот міг визначати своє місцеположення у просторі та одночасно побудовувати карту оточення з використанням власних сенсорів, таких як камери, лазерні сканери, гіроскопи та акселерометри. Ця інформація може бути використана для навігації робота, планування маршруту, уникнення перешкод та виконання різних завдань у реальному часі.

Одним з ключових елементів SLAM є алгоритми фільтрації, такі як фільтр Калмана або фільтр частинок, які використовуються для обробки вимірів сенсорів та оцінювання положення робота у просторі на кожному кроці часу. Ці алгоритми дозволяють роботу інтегрувати дані від різних сенсорів та коригувати свою оцінку положення з урахуванням помилок вимірювання та неоднорідностей у середовищі. Такий підхід дозволяє роботу ефективно визначати своє місцезнаходження навіть у складних та змінних умовах.

Однією з переваг SLAM є її універсальність та застосовність в різних галузях, від промислової автоматизації до автономних автомобілів та робототехніки. Вона може бути використана для навігації роботів в складних та змінних умовах, а також для побудови докладних мап оточення, що можуть бути використані для подальшої аналізу або планування дій. Однак, важливо враховувати, що SLAM може вимагати значних обчислювальних ресурсів та розуміння щодо налаштування параметрів для досягнення оптимальної ефективності [9].

Розпізнавання об'єктів та перешкод – це процес використання систем комп'ютерного зору для ідентифікації та класифікації об'єктів або перешкод у відео або зображеннях. Цей процес включає в себе використання різних алгоритмів та технік обробки зображень для аналізу піксельних даних та визначення характеристик об'єктів, таких як їхня форма, розмір, текстура та

колір. Результатом розпізнавання може бути ідентифікація об'єктів певних класів, таких як люди, автомобілі, дорожні знаки тощо, а також виявлення перешкод, які можуть перешкоджати нормальному функціонуванню системи.

Один з основних підходів до розпізнавання об'єктів та перешкод полягає у використанні методів глибокого навчання, таких як нейронні мережі. За допомогою навчальних даних, системи можуть навчитися впізнавати об'єкти за їхніми характеристиками та шаблонами. Наприклад, нейронні мережі можуть бути навчені розпізнавати обличчя людей або типи автомобілів на зображеннях. Цей підхід дозволяє системам автоматично виконувати розпізнавання об'єктів без необхідності вручну програмувати правила розпізнавання.

Одним із викликів у розпізнаванні об'єктів та перешкод є здатність системи адаптуватися до різних умов зйомки та змінних параметрів об'єктів. Наприклад, освітлення, орієнтація та розмір об'єктів можуть варіюватися в залежності від умов зйомки, що може ускладнити їхнє розпізнавання. Для подолання цього виклику необхідно розробляти алгоритми розпізнавання, які є стійкими до змін у середовищі та здатні адаптуватися до різних умов зйомки для досягнення надійності та ефективності в роботі. Порівняльний аналіз даних методів представлено в таблиці 1.1.

Таблиця 1.1 – Порівняльний аналіз основних методів відстеження об'єктів у робочій зоні мобільного робота

Характеристика	ВВ (камери)	ЛД (Лідар)	УВ (Ультразвук)	RFID
1	2	3	4	5
Дальність виявлення (м)	0,5 - 30	0,2 - 100	0,02 - 10	0,01 - 5
Точність визначення (см)	1 - 10	1 - 3	2 - 5	1 - 15

Продовження таблиці 1.1

1	2	3	4	5
Частота оновлення (Гц)	10 - 60	10 - 100	5 - 50	1 - 50
Вартість обладнання (USD)	100 - 5000	1000 - 20000	10 - 1000	50 - 2000
Вимоги до обробки даних	Високі	Середні	Низькі	Низькі
Працездатність у темряві	Низька	Висока	Висока	Висока
Чутливість до перешкод	Висока	Низька	Середня	Середня
Час реакції (мс)	50 - 200	10 - 100	20 - 100	10 - 200
Споживана потужність (Вт)	1 - 15	10 - 50	0,1 - 5	0,1 - 2

Проведений аналіз основних методів відстеження об'єктів у робочій зоні мобільного робота показав, що кожен метод має свої переваги і недоліки. Візуальне відстеження за допомогою камер (ВВ) забезпечує високу точність і деталізацію, але має значні вимоги до обробки даних та не працює ефективно в умовах поганого освітлення. Лідар (ЛД) забезпечує високу точність і дальність виявлення, але є найдорожчим варіантом і потребує середніх обчислювальних ресурсів.

Ультразвукове відстеження (УВ) є найдоступнішим і простим у використанні методом, але має обмежену дальність і середню точність. Радіочастотна ідентифікація (RFID) відзначається низькою вартістю та вимогами до обробки даних, але має обмежену дальність виявлення і точність. Кожен із методів має свої специфічні характеристики, які роблять

його придатним для різних застосувань в залежності від конкретних вимог і умов експлуатації.

Загалом, вибір методу відстеження об'єктів у робочій зоні мобільного робота залежить від конкретних задач і умов роботи. Для точного і далекого відстеження краще використовувати Лідар, для недорогих і простих систем – ультразвукові сенсори або RFID. Візуальне відстеження підходить для завдань, де потрібна висока деталізація, але з урахуванням високих вимог до обробки даних. Таким чином, кожен метод має свої переваги і недоліки, і вибір залежить від конкретної ситуації [10].

1.3 Аналіз алгоритмів роботи систем комп'ютерного зору

Алгоритми роботи систем комп'ютерного зору – це ключовий елемент багатьох сучасних технологій, включаючи відстеження об'єктів, розпізнавання образів, аналіз відео та багато іншого.

Методи відстеження об'єктів можуть бути розділені на кілька типів, таких як фонові модель (background subtraction), оптичний потік (optical flow), калманівські фільтри та методи з використанням нейронних мереж. Кожен з цих методів має свої переваги та недоліки і може бути використаний залежно від конкретного завдання та умов роботи.

Розпізнавання образів включає в себе використання алгоритмів глибокого навчання для ідентифікації об'єктів на зображеннях або відео. Популярними методами є згорткові нейронні мережі (CNN), які можуть використовуватися для класифікації об'єктів або виявлення об'єктів у відео. Також використовуються методи, засновані на особливостях (features), такі як метод гістограми орієнтованих градієнтів (HOG), які можуть бути ефективними для розпізнавання облич, автомобілів та інших об'єктів.

Обробка відео включає в себе різноманітні алгоритми для виявлення, відстеження та аналізу об'єктів у відеопотоці. Це може включати в себе

відстеження руху, виявлення та відслідковування об'єктів, аналіз динамічних змін у відеопотоці тощо. Популярними методами є алгоритми, які базуються на виявленні змін у пікселях між кадрами (background subtraction), а також алгоритми, які використовують нейронні мережі для виявлення та класифікації об'єктів у відеопотоці [11].

Основні алгоритми роботи систем комп'ютерного зору можна порівняти за такими критеріями, як точність, швидкість, складність та застосовність. У таблиці 1.2 представлено порівняння основних поширених алгоритмів роботи систем комп'ютерного зору.

Таблиця 1.2 – Порівняльний аналіз основних поширених алгоритмів роботи систем комп'ютерного зору

Алгоритм	Точність	Швидкість	Складність	Застосовність
Відстеження об'єктів	Варіює	Висока	Середня	Висока
Розпізнавання образів	Висока	Середня	Висока	Висока
Обробка відео	Варіює	Висока	Висока	Висока

Обираючи конкретний алгоритм для використання в конкретному випадку, важливо враховувати вимоги до точності, швидкості, складності та застосовності в конкретних умовах роботи.

1.4 Аналіз аналогічних рішень

Аналіз програм відстеження об'єктів у робочій зоні мобільного робота показує різноманітність рішень, які використовуються для оптимізації і автоматизації процесів у промисловості та інших сферах. Ці програми використовують різні технології, такі як комп'ютерний зір, Лідар, ультразвукові сенсори і RFID. Вибір відповідного рішення залежить від

конкретних вимог задачі, таких як точність, швидкість обробки, витрати та умови експлуатації [12].

Програми, що базуються на комп'ютерному зорі, є одними з найпоширеніших. Вони використовують камери та алгоритми машинного навчання для аналізу зображень і відео. Ці системи здатні розпізнавати об'єкти, відстежувати їх рух та навіть класифікувати за типами. Програми комп'ютерного зору забезпечують високу точність і деталізацію, але потребують значних обчислювальних ресурсів і працюють неефективно в умовах слабого освітлення. Наведемо приклади таких рішень:

- OpenCV – це відкрита бібліотека для комп'ютерного зору, яка надає широкий набір інструментів для обробки зображень і відео, розпізнавання об'єктів та аналізу руху. OpenCV використовується в багатьох робототехнічних проектах завдяки своїй гнучкості і потужності;

- TensorFlow є платформою машинного навчання від Google, яка включає можливості для обробки зображень. За допомогою моделей глибокого навчання на базі TensorFlow можна створювати складні системи розпізнавання і відстеження об'єктів.

Програми на основі лідара використовують лазерні промені для сканування простору і створення тривимірної карти. Вони забезпечують високу точність і дальність виявлення об'єктів, працюють незалежно від освітлення і можуть функціонувати в складних умовах. Лідарні системи часто використовуються в автономних транспортних засобах та робототехніці, але їхня вартість залишається високою. Наведемо приклади таких рішень:

- Velodyne виробляє лідарні сенсори, які широко використовуються в автономних транспортних засобах і робототехніці. Ці сенсори забезпечують високоточне тривимірне сканування простору, дозволяючи роботам точно визначати положення об'єктів;

– ROS (Robot Operating System) – це набір програмних бібліотек і інструментів для розробки робототехнічних додатків. ROS підтримує інтеграцію з лідарами і надає готові пакети для обробки даних від лідарів, таких як SLAM (Simultaneous Localization and Mapping).

Ультразвукові сенсори є популярними завдяки своїй доступності та простоті використання. Вони використовують звукові хвилі для виявлення об'єктів і визначення їх відстані. Ультразвукові програми підходять для завдань, де не потрібна висока точність і дальність, наприклад, для уникнення перешкод у роботах. Вони працюють добре в різних умовах, але їх точність обмежена. Наведемо приклади таких рішень:

– HC-SR04 є одним з найпоширеніших ультразвукових сенсорів, який використовується в багатьох проектах DIY і навчальних роботах. HC-SR04 дозволяє вимірювати відстань до об'єктів і уникати перешкод;

– ArduPilot – це платформа автопілотування для дронів і наземних роботів, яка підтримує використання ультразвукових сенсорів для уникнення перешкод і точного позиціонування.

RFID-системи використовують радіочастотні мітки для ідентифікації та відстеження об'єктів. Ці програми є дешевими і ефективними для відстеження об'єктів на коротких відстанях. RFID-технології широко використовуються в логістиці, складуванні та роздрібній торгівлі. Однак їх застосування обмежується дальністю дії та точністю, яка залежить від навколишніх умов:

– Impinj Speedway є високопродуктивним RFID-рідером, який використовується в логістичних системах для відстеження товарів. Він дозволяє швидко і точно зчитувати інформацію з RFID-міток;

– компанія Zebra пропонує широкий асортимент RFID-рішень для відстеження об'єктів у різних галузях, включаючи виробництво, роздрібну торгівлю і охорону здоров'я.

Комбінація кількох методів у програмах відстеження об'єктів може забезпечити більш надійні та ефективні рішення. Наприклад, поєднання комп'ютерного зору і Лідара дозволяє отримати високу точність і деталізацію в поєднанні з можливістю працювати в темряві. Використання таких комбінованих систем допомагає компенсувати недоліки окремих технологій.

Загалом, вибір програми відстеження об'єктів у робочій зоні мобільного робота залежить від конкретних потреб і обмежень. Для високоточних завдань у складних умовах найкраще підходять Лідар і комп'ютерний зір, тоді як для простих і недорогих рішень можуть використовуватися ультразвукові сенсори і RFID. Оптимальне рішення повинне враховувати всі аспекти завдання для досягнення максимальної ефективності та надійності [13].

2 РОЗРОБКА АЛГОРИТМУ ПРОГРАМИ ВІДСТЕЖЕННЯ ОБ'ЄКТІВ У РОБОЧІЙ ЗОНІ МОБІЛЬНОГО РОБОТА

2.1 Розробка структури програми

Розробка структури програми для відстеження об'єктів у робочій зоні мобільного робота є необхідною з декількох причин. По-перше, це дозволить ефективно використовувати функціонал мобільного робота, дозволяючи йому автоматично виявляти та відстежувати об'єкти у своєму оточенні. По-друге, програма відстеження об'єктів є важливою складовою для розвитку автономних систем, оскільки вона дозволяє роботу реагувати на зміни у своєму середовищі та взаємодіяти з ними.

Також, розробка структури програми дозволить забезпечити високу швидкодію та ефективність відстеження, що є критичними для успішної роботи мобільного робота. Крім того, вона дозволить розробникам легко розширювати та модифікувати функціонал програми для відстеження об'єктів у відповідності з конкретними потребами проекту. Таким чином, розробка структури програми відстеження об'єктів у робочій зоні мобільного робота є ключовим етапом у створенні ефективних та функціональних роботів.

Виходячи з проведеного аналізу розроблено наступну структуру програми відстеження об'єктів у робочій зоні мобільного робота, яка представлена на рисунку 2.1.

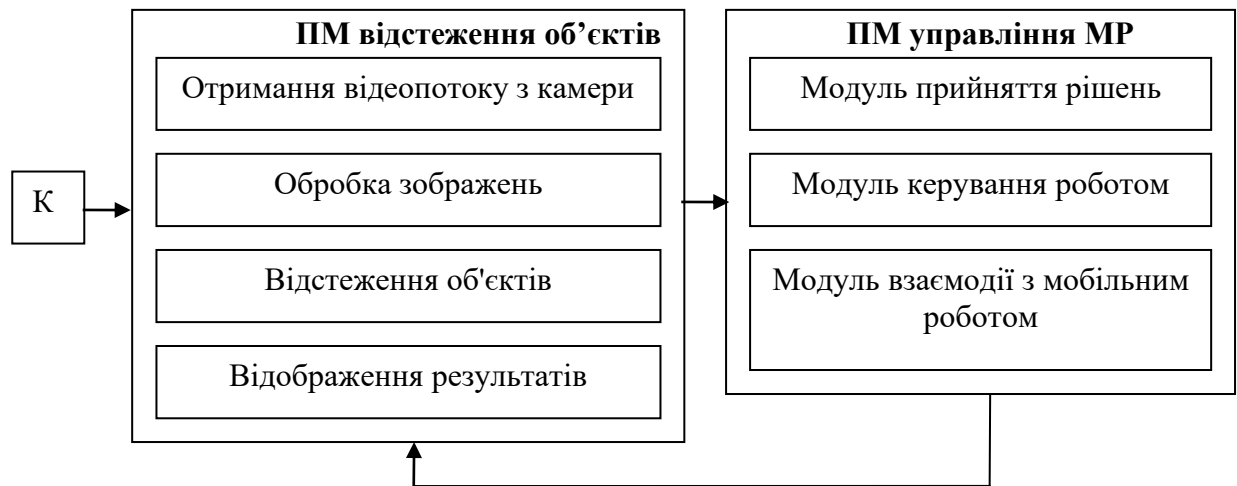


Рисунок 2.1 – Структура програми відстеження об'єктів у робочій зоні мобільного робота

Опишемо призначення кожного блока розробленої структури програми відстеження об'єктів у робочій зоні мобільного робота (рис. 2.1):

- К – камера;
- отримання відеопотоку з камери – відкриття потоку зображення з камери мобільного робота;
- обробка зображень – зчитування та обробка кадрів з відеопотоку (наприклад, виявлення об'єктів за допомогою алгоритмів комп'ютерного зору);
- відстеження об'єктів – визначення та відстеження об'єктів на кадрах (наприклад, за допомогою алгоритмів відстеження об'єктів);
- відображення результатів – відображення оброблених кадрів з відстеженими об'єктами (наприклад, у вікні відображення зображення);
- модуль прийняття рішень – необхідний для того, щоб робот міг приймати рішення щодо свого подальшого руху та дій на основі інформації, яку він отримує від системи відстеження об'єктів. Цей модуль аналізує дані про розташування та рух об'єктів у робочій зоні і виробляє стратегію керування рухом робота для досягнення поставленої мети, наприклад, досягнення об'єкту або уникнення перешкод;

– модуль керування роботом – визначення необхідних команд для відправлення на мобільний робот для керування його рухом;

– модуль взаємодії з мобільним роботом – відправлення команд керування на мобільний робот через відповідний інтерфейс.

Розробка структури програми відстеження об'єктів у робочій зоні мобільного робота має декілька переваг. По-перше, вона дозволяє ефективно використовувати ресурси мобільного робота, такі як обчислювальну потужність та пам'ять, що є критичним для оптимальної роботи в реальному часі. По-друге, гнучка структура дозволяє легко вносити зміни в програму, щоб адаптувати її до нових вимог або умов. Крім того, правильно спроектована структура забезпечує надійну роботу програми в різних умовах та уникнення помилок. Також вона полегшує супровід та розвиток програми з часом, що може значно зекономити час і ресурси. Нарешті, готова структура може значно прискорити процес розробки програми, особливо при використанні шаблонів та готових рішень.

2.2 Обґрунтування та вибір середовища розробки

Для аналізу мов високого рівня програмування для розроблення програми відстеження об'єктів у робочій зоні мобільного робота розглянемо Python, Java та JavaScript. Порівняння переваг та недоліків приведені в таблиці 2.1.

Таблиця 2.1 – Порівняння переваг та недоліків мов програмування

Характеристика	Python	Java	JavaScript
1	2	3	4
Простота вивчення та використання	+	-	+
Швидкодія	-	+	-

Продовження таблиці 2.1

1	2	3	4
Підтримка бібліотек та фреймворків для обробки зображень	+	+	+
Можливості для оптимізації та роботи з пам'яттю	-	+	-
Кросплатформенність	+	+	+
Спільнота користувачів та наявність документації	+	+	+
Підтримка паралельного програмування	+	+	+

Python виграє у простоті вивчення та використання, а також має сильну спільноту користувачів та широку підтримку бібліотек для обробки зображень. Однак, він може бути менш ефективним у плані швидкодії та оптимізації. Java має високу швидкодію та добре підходить для оптимізації та роботи з пам'яттю, але він складніший у вивченні та може вимагати більше часу для розробки. JavaScript, як і Python, має простоту вивчення та використання, а також має багато бібліотек для обробки зображень. Проте, він також може бути менш ефективним у швидкодії.

З урахуванням цих характеристик, Python може бути хорошим вибором для розроблення програми відстеження об'єктів у робочій зоні мобільного робота, особливо якщо важливою є простота розробки та наявність багатьох готових рішень для обробки зображень.

Для розроблення програми відстеження об'єктів у робочій зоні мобільного робота, вибір мови програмування та середовища розробки є важливим етапом. Мова програмування Python була обрана через її простоту та легкість вивчення, що є важливими аспектами для швидкого старту роботи над проектом. Python також має багато корисних бібліотек для обробки

зображень та роботи з даними, що дозволить зручно виконувати завдання з відстеження об'єктів.

Середовище розробки PyCharm 2022.2.3 було обране через його потужність та функціональність, що дозволить зручно писати, тестувати та налагоджувати код. PyCharm має вбудовану підтримку для мови Python, що спрощує роботу з нею. Також, в середовищі PyCharm є багато інструментів для роботи з відлагодженням коду та візуалізацією результатів, що є важливим для ефективної розробки програми відстеження об'єктів.

Отже, вибір мови Python та середовища розробки PyCharm 2022.2.3 є обґрунтованим для реалізації програми відстеження об'єктів у робочій зоні мобільного робота через їхню простоту використання та потужність функціоналу.

2.3 Математичне представлення методу відстеження об'єктів у робочій зоні мобільного робота

Найбільш поширений метод опису зображення складається з трьох окремих каналів кольору, що являють собою червоний колір (R), зелений (G) і синій (B). WIC забезпечує підтримку цих трьох каналів у порядку червоного зеленого синього (RGB) або синьо-зеленого червоного кольору (BGR). Нехай у нас є вхідне зображення потокового відео з камери мобільного робота в кольоровому форматі BGR.

Позначимо через $B(x, y)$, $G(x, y)$ та $R(x, y)$ значення пікселя на зображенні. Для реалізації розпізнавання та відстеження об'єкта в режимі реального часу на потоковому відео, на першому кроці проведемо перетворення на колірний простір HSV (Hue, Saturation, Value). Математична функція для перетворення з BGR HSV може бути записана наступним чином:

$$(H(x, y), S(x, y), V(x, y)) = f((B(x, y), G(x, y), R(x, y)), color_BGR2HSV), \quad (2.1)$$

де $B(x, y)$, $G(x, y)$, $R(x, y)$ – значення пікселя зображення у форматі BGR;

$H(x, y)$, $S(x, y)$, $V(x, y)$ – значення пікселя зображення у форматі HSV;

f – функція перетворює значення зображення BGR в HSV, у цьому дослідженні буде використовуватися функція `cv2.cvtColor` для бібліотеки `cv2` мови Python.

Після цього введемо опис порогових значень діапазонів відтінків, насиченості та яскравості виділення необхідного об'єкта в кадрі:

$$\begin{cases} H_{low} \leq x_i \leq H_{high} \\ S_{low} \leq x_k \leq S_{high} \\ V_{low} \leq x_q \leq V_{high} \end{cases}, \quad (2.2)$$

де x_i, x_k, x_q – відповідні значення відтінків, насиченості та яскравості для кольорового простору HSV (Hue, Saturation, Value), за умови, що значення $H_{low}, S_{low}, V_{low} \geq 0$ та $H_{high}, S_{high}, V_{high} \leq 255$.

На отримане зображення в колірний простір HSV проведемо бінаризацію, з умовою, що бінаризація виконується з використанням порогових значень для кожного каналу HSV, як описано у виразі (2.2) і створимо маску зображення відповідно до наступного виразу:

$$M(x, y) = \begin{cases} 1, & \text{if } H_{low} \leq x_i \leq H_{high}, S_{low} \leq x_k \leq S_{high}, V_{low} \leq x_q \leq V_{high} \\ 0 & \end{cases}, \quad (2.3)$$

де $M(x, y)$ – маска з координатами (x, y) ;

x_i, x_k, x_q – відповідні значення відтінків, насиченості та яскравості для колірною простору HSV (Hue, Saturation, Value), за умови, що значення $H_{low}, S_{low}, V_{low} \geq 0$ та $H_{high}, S_{high}, V_{high} \leq 255$.

За маскою необхідно визначити послідовність знаходження контурів. Нехай M – бінарна маска, що є зображенням після бінаризації. Пікселі в цій

масці приймають значення 0 або 1 де 1 позначає об'єкт, а 0 – фон. Спочатку визначимо операцію виділення контуру об'єкта. Нехай C – контур об'єкта на масці M , тоді операцію виділення можна представити у такому вигляді:

$$C = \{(x,y)/M(x, y) = 1\}, \quad (2.4)$$

де (x, y) – координати пікселя на масці;

$M(x, y)$ – значення пікселя.

Наступним кроком, використовуючи виділені контури, отримуємо список контурів, де кожен контур представлений безліччю точок:

$$C_{list} = \{C_1, C_2, \dots, C_m\}, \quad (2.5)$$

де C_{list} – список контурів, де кожен контур представлений безліччю точок;

C_m – m -й контур представлений як безліч точок.

Таким чином, послідовність математичних визначень знаходження контурів по масці виглядає так:

$$M \rightarrow C \rightarrow C_{list} \rightarrow \{C_1, C_2, \dots, C_m\} \rightarrow n_j, \quad (2.6)$$

де M – бінарна маска;

C – контур об'єкта на масці;

C_{list} – список контурів;

C_m – m -й контур зі списку контурів;

n_j – кількість точок в j -м контурі.

Ефективний метод для відстеження об'єктів на основі їх кольору. Метод простий у реалізації і може працювати в реальному часі, що є важливим для багатьох застосувань мобільних роботів. Використання цього

методу дозволяє швидко та ефективно виділяти об'єкти певного кольору на відео, без складних обчислень. Крім того, обраний метод може легко адаптуватися для використання з об'єктами різних кольорів шляхом зміни параметрів кольорової маски та має низьку обчислювальну навантаження і може працювати на простих обчислювальних пристроях, що робить її доступною для використання на бюджетних мобільних роботах. Розроблена програма на базі обраного методу має досить легко налаштовується за допомогою зміни параметрів, таких як розмір маркера або товщина лінії, в кодї, що робить її гнучкою і придатною для використання у різних сценаріях.

2.4 Розробка загального алгоритму роботи програми

Розробка загального алгоритму роботи програми для відстеження об'єктів у робочій зонї мобільного робота обґрунтовується кількома ключовими перевагами. По-перше, такий алгоритм стандартизує процес відстеження, забезпечуючи його однаковість для різних об'єктів та умов. Він також є універсальним, оскільки може застосовуватися до різних об'єктів з різними параметрами. Ефективність алгоритму полягає в його здатності використовувати оптимальні методи відстеження, що покращує продуктивність програми. Крім того, його простота використання дозволяє використовувати один алгоритм для різних завдань без додаткової розробки. Використання загального алгоритму також допомагає зменшити складність програми та зробити її більш зрозумілою. Масштабованість – ще одна перевага, оскільки алгоритм може легко адаптуватися до різних типів об'єктів та умов. Нарешті, розробка одного загального алгоритму забезпечить економію часу та ресурсів на розробку та супровід програми. Виходячи з мети роботи та розробленої структурної схеми (рис. 2.1), пропонується наступний загальний алгоритм роботи програми відстеження об'єктів у робочій зонї мобільного робота, яка представлена на рисунку 2.2.

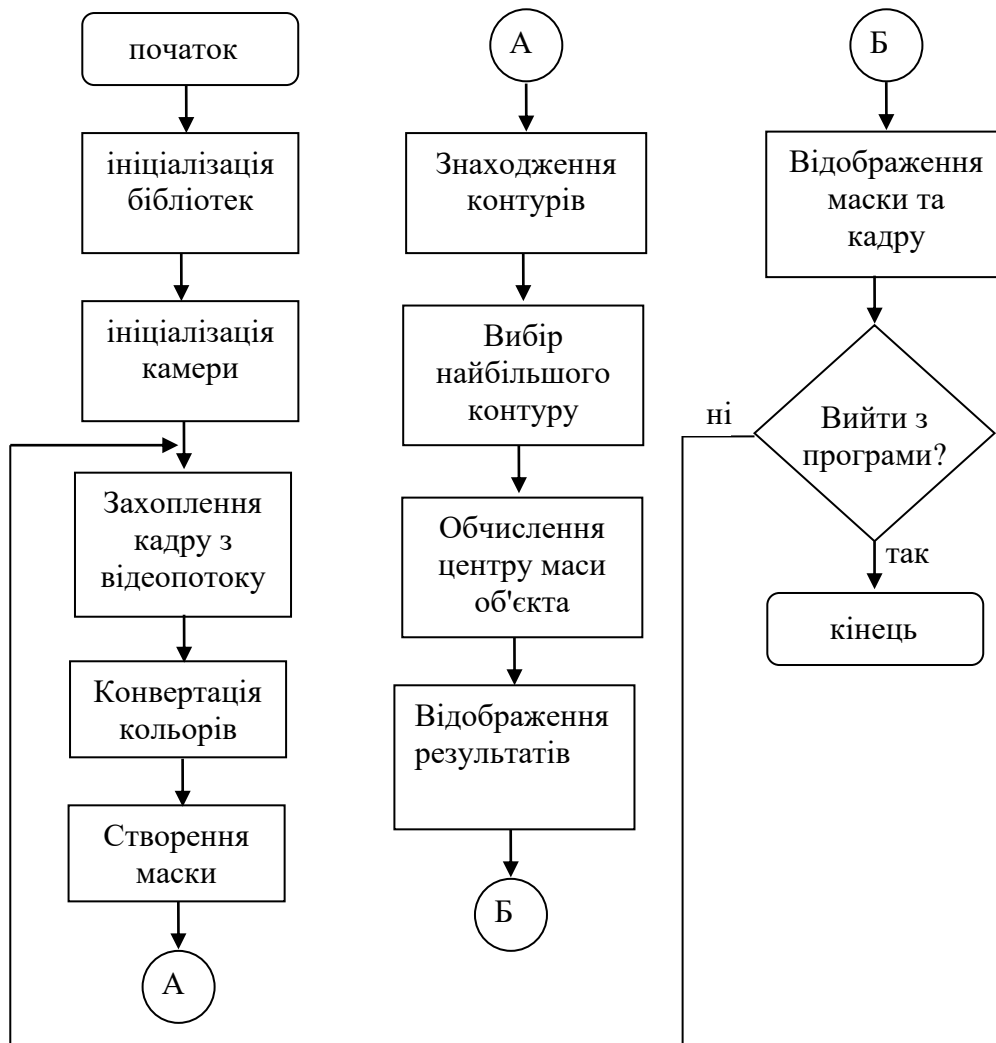


Рисунок 2.2 – Загальний алгоритм роботи програми відстеження об'єктів у робочій зоні мобільного робота

Опишемо призначення кожного блока загального алгоритму роботи програми відстеження об'єктів у робочій зоні мобільного робота:

– блок ініціалізація бібліотек відіграє ключову роль у підготовці середовища для правильної роботи програми. Цей блок включає в себе імпорт необхідних бібліотек, таких як OpenCV та NumPy, що забезпечують функціонал для обробки відео та математичних операцій. Також в цьому блоці може відбуватися налаштування параметрів алгоритму, наприклад, встановлення діапазону кольорів для відстеження. Ініціалізація бібліотек дозволяє забезпечити правильну роботу всіх функцій та методів, що будуть

використовуватися в програмі, та готує середовище для подальшої роботи з відеопотоком та обробкою зображень;

– блок ініціалізації камери в алгоритмі роботи програми відстеження об'єктів у робочій зоні мобільного робота є важливим етапом для забезпечення коректної роботи програми. Цей блок дозволяє програмі отримувати відеопоток з камери, який буде використовуватися для виявлення та відстеження об'єктів. Ініціалізація камери включає в себе встановлення параметрів камери, таких як роздільна здатність, частота кадрів, режими кольорової моделі тощо. Цей блок також може включати в себе перевірку доступності та правильності підключення камери, щоб уникнути можливих проблем у подальшій роботі програми. Після успішної ініціалізації камери програма готова до отримання відеопотоку та подальшої обробки зображень для відстеження об'єктів;

– блок захоплення кадру з відеопотоку в алгоритмі роботи програми відстеження об'єктів у робочій зоні мобільного робота необхідний для отримання зображень з камери та їх подальшої обробки. Цей блок дозволяє програмі отримувати поточний кадр відеопотоку з камери, що дозволяє виявляти та відстежувати об'єкти у реальному часі. Захоплення кадру включає в себе читання поточного кадру з відеопотоку та зберігання його у вигляді зображення, що може бути подане на відображення. Цей блок також може включати в себе операції підготовки кадру до подальшої обробки, такі як конвертація кольорових просторів або видалення шуму. Після захоплення кадру програма готова до подальшої обробки та виявлення об'єктів на зображенні;

– блок конвертації кольорів з відеопотоку в алгоритмі роботи програми відстеження об'єктів у робочій зоні мобільного робота необхідний для перетворення кольорів кадру з одного кольорового простору в інший. Це може бути потрібно, наприклад, для забезпечення коректної обробки кадрів у потрібному кольоровому форматі, який підходить для подальших обчислень.

Конвертація кольорів також може використовуватися для підготовки зображень до використання в різних алгоритмах обробки зображень, де потрібний певний кольоровий простір. Наприклад, у випадку відстеження об'єктів за кольором, необхідно перетворити кадр у кольоровий простір HSV, щоб зручно визначати діапазон кольорів для відстеження. Після конвертації кольорів програма може продовжити обробку та відстеження об'єктів на зображенні у потрібному форматі;

– блок створення маски з відеопотоку в алгоритмі роботи програми відстеження об'єктів у робочій зоні мобільного робота необхідний для виділення областей на зображенні, які відповідають заданим параметрам. У відстеженні об'єктів за кольором, наприклад, маска створюється для виділення пікселів, які потрапляють у визначений діапазон кольорів. Це дозволяє відокремити об'єкти цікавлення від фону або інших об'єктів на зображенні. Створення маски є важливим етапом, оскільки вона створює основу для подальшого визначення контурів об'єктів та їх аналізу. Після створення маски програма може використовувати її для виявлення та відстеження об'єктів у реальному часі;

– блок знаходження контурів з відеопотоку в алгоритмі роботи програми відстеження об'єктів у робочій зоні мобільного робота необхідний для визначення зовнішніх меж областей на зображенні, що мають схожу інтенсивність або колір. Контур – це набір точок на зображенні, які утворюють замкнуту криву і відображають зовнішній контур об'єкту. Після знаходження контурів програма може використовувати їх для визначення положення та форми об'єктів на зображенні. Це дозволяє відстежувати об'єкти у реальному часі та взаємодіяти з ними. Знаходження контурів є важливим етапом у відстеженні об'єктів, оскільки воно надає інформацію про геометрію об'єктів, яка може бути використана для подальшого аналізу та управління мобільним роботом;

– блок вибору найбільшого контуру з відеопотоку в алгоритмі роботи програми відстеження об'єктів у робочій зоні мобільного робота потрібний для визначення головного об'єкту чи області інтересу на зображенні. Він дозволяє вибрати найбільший контур серед всіх знайдених, що часто відповідає головному об'єкту на зображенні. Цей блок є важливим для точного визначення положення та розмірів об'єкта, що відстежується. Вибір найбільшого контуру допомагає уникнути помилок, пов'язаних з іншими областями на зображенні, які можуть ввести у нерозуміння систему відстеження. Цей етап є важливим для точного та надійного відстеження об'єктів у реальному часі;

– блок обчислення центру маси об'єкта з відеопотоку в алгоритмі роботи програми відстеження об'єктів у робочій зоні мобільного робота використовується для точного визначення положення центру маси об'єкта на зображенні. Цей центр маси є важливим показником для визначення точного положення об'єкта в просторі. Обчислення центру маси базується на обробці контурів об'єктів, знайдених на зображенні. Для кожного контуру обчислюється центр маси, що дозволяє визначити точку, яка найкраще відображає реальне положення об'єкта на зображенні. Цей показник може використовуватися для подальшого аналізу та управління рухом мобільного робота. Обчислення центру маси допомагає забезпечити точність та стабільність відстеження об'єктів у реальному часі;

– блок відображення результатів маси об'єкта з відеопотоку в алгоритмі роботи програми відстеження об'єктів у робочій зоні мобільного робота необхідний для візуалізації результатів обробки кадру. Цей блок відображає на екрані користувача область, де знаходиться об'єкт, його центр маси та інші важливі параметри. Візуалізація результатів дозволяє оператору чітко спостерігати за процесом відстеження об'єктів та вчасно реагувати на можливі зміни в середовищі. Крім того, відображення результатів дозволяє здійснювати візуальний контроль за правильністю роботи алгоритму та

вирішувати можливі проблеми, які можуть виникнути під час відстеження об'єктів. Такий підхід допомагає підвищити ефективність та надійність роботи програми відстеження об'єктів у реальному часі;

– блок відображення маски та кадру об'єкта з відеопотоку в алгоритмі роботи програми відстеження об'єктів у робочій зоні мобільного робота необхідний для візуалізації процесу обробки кадру та роботи алгоритму в реальному часі. Відображення маски дозволяє побачити області зображення, які відповідають критеріям відстеження об'єктів, що допомагає оператору визначити правильність вибору кольорів та параметрів фільтрації. Також, відображення маски може служити для налагодження параметрів алгоритму та виправлення можливих помилок.

2.5 Розробка алгоритму відстеження об'єктів

Розробка алгоритму відстеження об'єктів є ключовою для програми, яка керує роботом у робочій зоні. Цей алгоритм дозволяє роботу виявляти та відстежувати об'єкти навколишнього середовища, що дозволяє роботу адаптуватися до змін у середовищі. Відстеження об'єктів є важливим елементом для безпечного та ефективного руху робота в просторі. Алгоритм відстеження допомагає роботу уникати зіткнень з перешкодами та навчає його працювати в умовах обмеженого простору. Крім того, відстеження об'єктів може бути використане для взаємодії з оточуючим середовищем, наприклад, для збору даних або доставки предметів. Розробка алгоритму відстеження дозволяє оптимізувати роботу та підвищувати її продуктивність. Розроблений алгоритм відстеження об'єктів у робочій зоні мобільного робота представлено на рисунку 2.3.

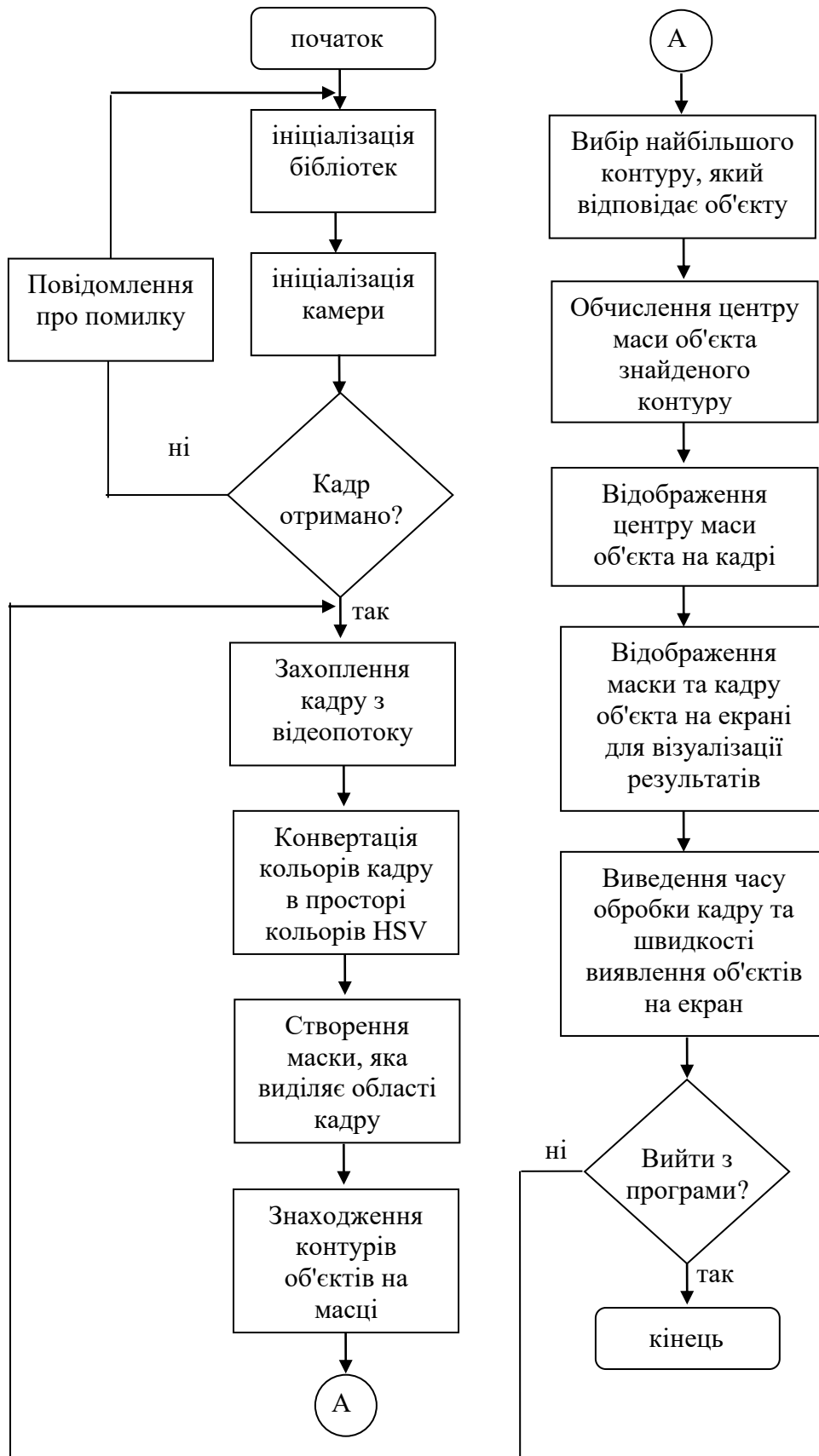


Рисунок 2.3 – Алгоритм відстеження об'єктів у робочій зоні мобільного
робота

Опишемо призначення кожного блоку розробленого алгоритму відстеження об'єктів у робочій зоні мобільного робота:

– блок "Захоплення кадру з відеопотоку" в алгоритмі відстеження об'єктів у робочій зоні мобільного робота є важливою частиною, оскільки він визначає основу для подальшого аналізу зображення. Цей блок відповідає за отримання зображення з відеопотоку, яке буде використовуватися для визначення положення та руху об'єктів у робочій зоні. Математично блок "Захоплення кадру" можна описати як функцію, яка приймає на вхід відеопотік і повертає матрицю зображення. Від цього зображення вже будуть починатися подальші обчислення, включаючи конвертацію кольорів, створення маски, знаходження контурів та інші операції, необхідні для відстеження об'єктів;

– блок "Конвертація кольорів кадру в просторі кольорів HSV" в алгоритмі відстеження об'єктів у робочій зоні мобільного робота відповідає за перетворення кольорів кожного пікселя зображення з простору кольорів BGR (Blue-Green-Red), що використовується в OpenCV, в простір кольорів HSV (Hue-Saturation-Value). Це перетворення дозволяє зручно працювати з кольорами, оскільки він розбиває кожен колір на компоненти: відтінок (Hue), насиченість (Saturation) і значення (Value);

– блок "Створення маски" в алгоритмі відстеження об'єктів у робочій зоні мобільного робота відповідає за виділення областей кадру, які відповідають певним критеріям. У випадку відстеження об'єктів за кольором, як у даному випадку, маска створюється за допомогою порівняння кольорів пікселів з певними властивостями (наприклад, відтінком, насиченістю і значенням у просторі HSV);

– блок "Знаходження контурів об'єктів на масці" в алгоритмі відстеження об'єктів у робочій зоні мобільного робота відповідає за виявлення замкнених кривих, що утворюють контури об'єктів на масці. Для цього можна використовувати алгоритми знайдення контурів, наприклад,

алгоритм знаходження зовнішніх контурів `cv2.RETR_EXTERNAL` з бібліотеки `OpenCV`;

– блок "Вибір найбільшого контуру" в алгоритмі відстеження об'єктів у робочій зоні мобільного робота відповідає за вибір контуру, який найбільше відповідає об'єкту, який слід відстежувати. Для цього можна використовувати функцію, яка знаходить контур з найбільшою площею або з найбільшим дуговим довжиною;

– блок «Обчислення центру маси об'єкта знайденого контуру» у алгоритмі відстеження об'єктів у робочій зоні мобільного робота буде використовувати метод моментів. Метод моментів – це техніка в аналізі зображень, яка використовується для обчислення різних властивостей об'єктів на зображенні, таких як центр маси, орієнтація, площа тощо. Цей метод базується на обчисленні математичних моментів області, яку потрібно аналізувати;

– блок «Відображення центру маси об'єкта на кадрі» використовується такий підхід: спочатку обчислюється центр маси об'єкта за допомогою методу моментів, який визначається як координати. Після обчислення центру маси об'єкта можна відобразити його на кадрі, наприклад, за допомогою функції відображення точки чи іншого маркеру на зображенні в місці цього центру. Це дозволяє візуалізувати результат обробки та слідкувати за рухом об'єкта на відео потоці;

– блок «Відображення маски та кадру об'єкта на екрані для візуалізації результатів» використовувати функції відображення зображень бібліотеки `OpenCV`, такі як `cv2.imshow()`. Такий підхід дозволяє в реальному часі відслідковувати рух об'єкта та побачити результати роботи алгоритму відстеження на відео потоці;

– блок «Виведення часу обробки кадру та швидкості виявлення об'єктів на екран» спочатку необхідно засікти час початку обробки кадру, наприклад, за допомогою функції `time.time()`. Після обробки кадру, засікти час

завершення обробки. Потім можна розрахувати час обробки кадру, віднявши час початку від часу завершення. Швидкість виявлення об'єктів можна розрахувати як обернене значення часу обробки, якщо час обробки більше нуля, або як нуль у протилежному випадку. Отримані значення можна вивести на екран для відображення користувачу. Такий підхід дозволяє контролювати час обробки кадру та швидкість виявлення об'єктів у реальному часі.

Розроблений алгоритм відстеження об'єктів у робочій зоні мобільного робота має кілька переваг:

- ефективність, завдяки використанню кольорової маски та методу моментів для знаходження контурів, алгоритм може швидко та точно відстежувати об'єкти на відео потоці;
- простота реалізації алгоритма складається лише з кількох кроків, що робить його легким у реалізації та розумінні;
- гнучкість, за допомогою налаштувань кольорової маски можна легко адаптувати алгоритм для відстеження різних об'єктів та умов освітлення;
- ресурсозбереження, алгоритм використовує лише базові операції обробки зображень, що дозволяє йому працювати на простій апаратурі;
- реалізація у реальному часі завдяки оптимізації та ефективності, алгоритм може працювати у реальному часі, що важливо для застосувань у робототехніці;
- візуалізація результатів алгоритм відображає на екрані не лише оброблений кадр, а й інформацію про об'єкт, що дозволяє користувачу легше відстежувати процес;
- надійність, завдяки використанню методу моментів для знаходження центру маси об'єкта, алгоритм стійкий до різних умов освітлення та шумів на зображенні;

– можливість розширення алгоритму для виявлення та відстеження більшої кількості об'єктів одночасно або для виконання додаткових завдань, таких як класифікація об'єктів.

2.6 Розрахунок швидкості обробки кадрів при відстеженні об'єктів

Розрахунок швидкості обробки кадрів є важливим етапом у відстеженні об'єктів, оскільки він дозволяє оцінити продуктивність програми і обладнання. Основна мета цього розрахунку – визначити, наскільки швидко програма може обробляти кадри і виявляти об'єкти. Знання швидкості обробки допомагає вибрати оптимальні параметри алгоритму, такі як розмір області відстеження або рівень деталізації маски. Це дозволяє підвищити продуктивність програми та забезпечити її ефективну роботу на конкретному обладнанні. Також, швидкість обробки кадрів може бути використана для прогнозування продуктивності програми на інших пристроях або в інших умовах. Виявлення недоліків у програмі, таких як неефективні алгоритми або обмеження обладнання, можливо завдяки аналізу швидкості обробки кадрів. Таким чином, розрахунок швидкості обробки кадрів є важливим для планування ресурсів, виявлення проблем та підвищення ефективності програм відстеження об'єктів у робочій зоні мобільного робота.

Приведемо приблизний розрахунок швидкості обробки кадрів для розробленої програми відстеження об'єктів у робочій зоні мобільного робота для наступних апаратних модулів: Raspberry Pi 3 Model B+ на базі 64-бітний 4-ядерний ARMv8 Cortex-A53 з тактовою частотою 1,4 ГГц, та ноутбука Acer Nitro 5 AN517-55 на базі Core i7-12650H 3,6 ГГц. Тоді час обробки одного кадру ($t_{processing}$) розраховується за формулою:

$$t_{processing} = t_{end} - t_{start}, \quad (2.7)$$

де t_{end} – час закінчення обробки кадру;

t_{start} – час початку обробки кадру.

Розрахунок FPS (кадрів в секунду) важливий для визначення швидкості обробки кадрів у відео або зображеннях. Розрахунок FPS допомагає забезпечити оптимальну якість відтворення або обробки зображень та уникнути затримок або низької продуктивності. Формула розрахунку приведена по наступній формулі:

$$FPS = \frac{1}{t_{processing}}, \quad (2.8)$$

де $t_{processing}$ – час обробки одного кадру.

В формули (2.7)-(2.8) підставимо аналітичні значення та проведемо розрахунки при роботі програми на різних апаратних модулях, таких як Raspberry Pi 3 Model B+ та Acer Nitro 5 AN517-55. Результати розрахунку представлені в таблиці 2.2.

Таблиця 2.2 – Результати розрахунку значення FPS для різних апаратних модулів Raspberry Pi 3 Model B+ та Acer Nitro 5 AN517-55

Апаратні модулі	t_{start}	t_{end}	$t_{processing}$	FPS
Raspberry Pi 3 Model B+	10,5	11,2	0,7	1,43
Acer Nitro 5 AN517-55	5,8	6,4	0,6	1,67

Відповідно отриманим результатам розрахунку, представлених в таблиці 2.2, можливо зробити наступні висновки. Acer Nitro 5 AN517-55 має кращі показники FPS, порівняно з Raspberry Pi 3 Model B+ через декілька ключових факторів. По-перше, Acer Nitro 5 AN517-55 в основному є ноутбуком, спеціально призначеним для геймінгу, тому він має потужніший

процесор, графічну карту та систему пам'яті, що забезпечує кращу продуктивність у відтворенні графіки та обробці відео. Додатково, Acer Nitro 5 AN517-55, як ноутбук, має більше ресурсів, таких як більший обсяг оперативної пам'яті та швидший накопичувач, що дозволяє йому більш ефективно обробляти графічні дані. З іншого боку, Raspberry Pi 3 Model B+ є одноплатним комп'ютером, який має обмежену потужність обчислювальних ресурсів порівняно з ноутбуком. Цей комп'ютер призначений для різних завдань, а не лише для графічних операцій, тому його продуктивність у відтворенні відео та обробці графіки може бути меншою. Крім того, Raspberry Pi 3 Model B+ має обмежену кількість оперативної пам'яті та менш швидкий процесор, що також може вплинути на його здатність обробляти графічні дані. Отже, через більш потужні обчислювальні ресурси та більшу кількість оперативної пам'яті Acer Nitro 5 AN517-55 може показувати кращі показники FPS у графічно вимогливих застосунках, порівняно з Raspberry Pi 3 Model B+.

3 РОЗРОБКА ПРОГРАМИ ДЛЯ ВІДСТЕЖЕННЯ ОБ'ЄКТІВ У РОБОЧІЙ ЗОНІ МОБІЛЬНОГО РОБОТА

3.1 Розробка НМІ інтерфейсу користувача

Розробка інтерфейсу користувача (НМІ) для програми відстеження об'єктів у робочій зоні мобільного робота є важливою складовою, яка дозволяє зручно керувати та візуалізувати процес відстеження. У НМІ можуть бути різноманітні елементи, такі як відеопотік з камери, відображення траєкторії руху об'єкта, контроль вибору кольору об'єкта, відображення інформації про швидкість обробки кадрів та інші. Важливою частиною розробки НМІ є зручний та інтуїтивно зрозумілий інтерфейс, що дозволяє оператору легко керувати програмою та візуалізувати важливі дані. В процесі розробки НМІ слід враховувати потреби та можливості користувача, а також забезпечити можливість налаштування параметрів відстеження для оптимальних результатів роботи. Внаслідок цього, пропонується наступна структура НМІ інтерфейсу програми для відстеження об'єктів у робочій зоні мобільного робота, яка представлена на рисунку 3.1.

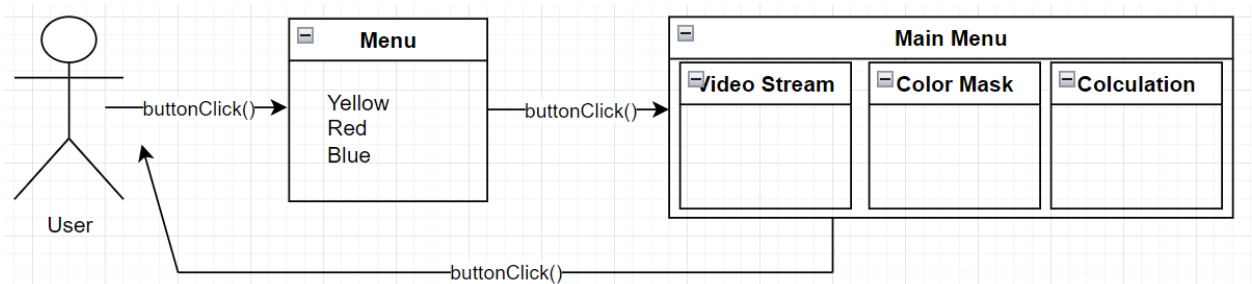


Рисунок 3.1 – Структура взаємодій користувача з НМІ інтерфейсу програми для відстеження об'єктів у робочій зоні мобільного робота

Елемент "User" у діаграмі UML може відображати користувача системи або певну групу користувачів, які взаємодіють з програмою або системою.

Елемент “buttonClick()” зазвичай використовується для опису дії, яка відбувається при натисканні кнопки на інтерфейсі користувача та використовуватися для навігації користувача в програмі, наприклад, перехід на іншу сторінку, відкриття нового вікна або виконання іншої дії, пов'язаної з інтерфейсом. В “Menu” користувач обирає колір маски для відстеження об'єктів у робочій зоні мобільного робота. Після цього "User" натискає елемент “buttonClick()”, та відкривається елемент “Main Menu”, який вміщує вікна “Video Stream”, “Color Mask” та “Colculation”. Елемент вікна “Video Stream” показує відео потік в реальному часі, елемент “Color Mask” показує маску обранного кольору та елемент “Colculation” виводить результати розрахунку FPS.

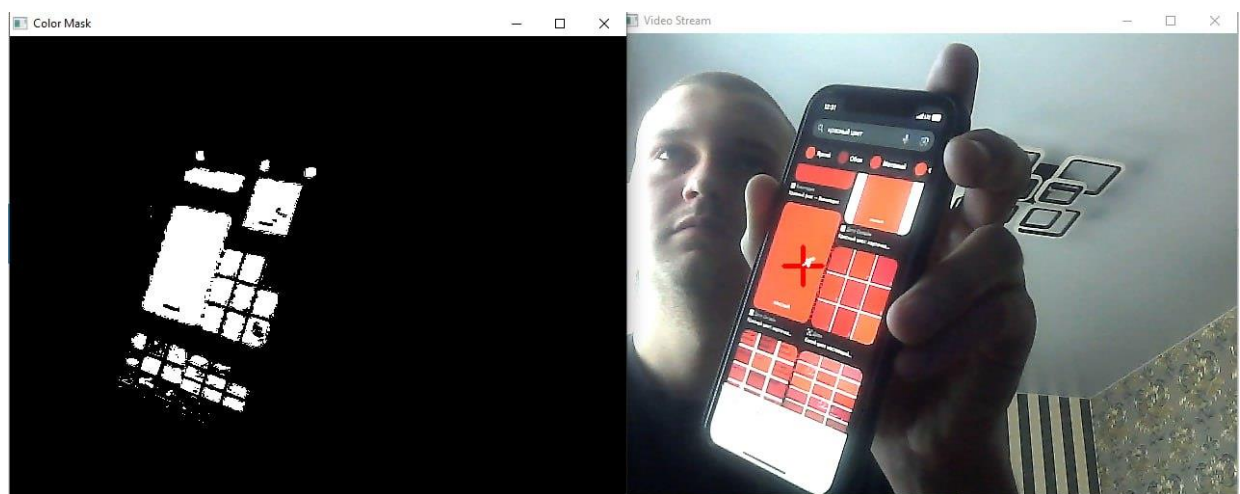
Відповідно до розробленої структури взаємодій користувача з НМІ інтерфейсу програми для відстеження об'єктів у робочій зоні мобільного робота (рис. 3.1) були розроблені наступні візуальні елементи інтерфейсу “Menu”, які реалізовані на базі мови Python, та дозволяють користувачу обрати колір який буде відстежуватися, представлено на рисунку 3.2.



Рисунок 3.2 – Реалізація візуальних елементів інтерфейсу “Menu” вибору кольору маски

Реалізація НМІ інтерфейсу користувача у вигляді трьох вікон ("Video Stream", "Color Mask", "Calculation") має кілька переваг. Вона дозволяє користувачу одночасно спостерігати за поточним відеопотоком, візуалізувати оброблені дані у вигляді кольорової маски та отримувати результати

обчислень. Такий підхід полегшує користувачеві сприйняття та навігацію у програмі, дозволяє швидко змінювати фокус уваги між різними аспектами роботи програми та забезпечує зручний доступ до необхідної інформації. Крім того, така структура інтерфейсу дозволяє оптимально використовувати простір екрану, зберігати стан вікон для подальшого використання та підвищує загальний комфорт та ефективність користувача під час використання програми. Приклад реалізації вікон "Video Stream", "Color Mask" приведено на рисунку 3.3.



а)

б)

а) вікно Video Stream;

б) вікно Color Mask

Рисунок 3.3 – Реалізація НМІ інтерфейсу користувача програми для відстеження об'єктів у робочій зоні мобільного робота

Вікно "Calculation" в розробленій програмі для відстеження об'єктів у робочій зоні мобільного робота було реалізовано для виведення результатів обчислення швидкості обробки кадрів. Це дозволяє користувачу в реальному часі відслідковувати продуктивність програми та виявляти можливі проблеми з швидкістю обробки кадрів. Виведення швидкості обробки кадрів у окреме вікно надає користувачеві можливість зосередитися на цій важливій метриці, не відволікаючись на інші аспекти програми. Крім того, такий підхід

дозволяє вчасно виявляти проблеми з продуктивністю та швидко реагувати на них, що важливо для забезпечення ефективної роботи програми у реальних умовах. Разом з тим, виведення результатів розрахунків у окреме вікно сприяє зручній інтеракції з користувачем та полегшує процес моніторингу та аналізу продуктивності програми. Приклад реалізації вікна виведення результатів розрахунку швидкості приведено на рисунку 3.4.

```
Processing Time: 0.0050 seconds
Detection Speed: 200.65 FPS
Processing Time: 0.0040 seconds
Detection Speed: 247.50 FPS
Processing Time: 0.0050 seconds
Detection Speed: 200.17 FPS
Processing Time: 0.0050 seconds
Detection Speed: 200.17 FPS
Processing Time: 0.0030 seconds
Detection Speed: 334.37 FPS
Processing Time: 0.0050 seconds
Detection Speed: 200.51 FPS
Processing Time: 0.0050 seconds
Detection Speed: 200.17 FPS
Processing Time: 0.0040 seconds
Detection Speed: 250.75 FPS
Processing Time: 0.0050 seconds
Detection Speed: 200.46 FPS
Processing Time: 0.0010 seconds
Detection Speed: 1004.38 FPS
Processing Time: 0.0010 seconds
Detection Speed: 1048.31 FPS
Processing Time: 0.0050 seconds
Detection Speed: 200.54 FPS
Processing Time: 0.0040 seconds
Detection Speed: 250.75 FPS
Processing Time: 0.0050 seconds
Detection Speed: 200.56 FPS
Processing Time: 0.0050 seconds
```

Рисунок 3.4 – Реалізація вікна виведення результатів розрахунку швидкості обробки кадрів

3.2 Реалізація функцій відстеження об'єктів

У розробленій програмі для відстеження об'єктів у робочій зоні мобільного робота використовується комплексний підхід до реалізації

функцій відстеження об'єктів. Перш за все, програма використовує відеопотік з веб-камери для отримання зображень. Після цього кадри конвертуються у кольоровий простір HSV для кращої обробки кольорів. Далі використовується кольоровий фільтр, що дозволяє виділити об'єкти певного кольору на зображенні. Пошук контурів об'єктів здійснюється за допомогою алгоритмів комп'ютерного зору, що дозволяє визначити форму та положення об'єктів на кадрі. Для відстеження руху об'єктів використовується аналіз зміни їхнього положення на кадрах. Також реалізовано можливість відображення траєкторії руху об'єктів на зображенні. Програма також вміє відображати результати аналізу у вигляді масок кольорів на зображенні. Для виведення результатів та контролю над програмою використовується графічний інтерфейс користувача. Усі ці етапи реалізації дозволяють програмі ефективно відстежувати об'єкти та аналізувати їхній рух у робочій зоні мобільного робота. Програмна реалізація функцій відстеження об'єктів приведено нижче.

```
# Функція для оновлення траєкторії
def update_trajectory(x, y):
    trajectory.append((x, y))
    if len(trajectory) > TRAJECTORY_LENGTH:
        del trajectory[0]
```

Даний фрагмент коду представляє собою функцію, яка використовується для оновлення траєкторії руху об'єкта на зображенні. Кожен раз, коли функція викликається з новими координатами (x, y) об'єкта, вона додає ці координати до списку траєкторії. Після цього функція перевіряє довжину списку траєкторії: якщо він перевищує задану максимальну довжину (TRAJECTORY_LENGTH), то видаляє перший елемент списку. Це реалізує логіку зберігання та оновлення історії руху об'єкта, дозволяючи відображати його траєкторію на зображенні.

```
# Функція для відображення траєкторії
```

```
def draw_trajectory(frame):
    for i in range(1, len(trajectory)):
        if i % 5 == 0: # Відображає тільки кожну п'яту точку
            cv2.line(frame, trajectory[i - 1], trajectory[i], (255, 255, 255), 2)
```

Даний фрагмент коду представляє собою функцію, яка використовується для відображення траєкторії руху об'єкта на кадрі зображення. Функція проходить по списку координат траєкторії і кожній п'ятій точці в списку (з початку) викликає функцію `cv2.line`, яка малює лінію між поточною точкою та попередньою точкою траєкторії на кадрі. Це дозволяє візуалізувати траєкторію руху об'єкта на зображенні, що є корисним для аналізу його руху та поведінки.

```
# Функція для обробки кадру
def process_frame(frame, color):
    start_time = time.time()
    hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
    lower_color, upper_color = get_color_bounds(color)
    mask = cv2.inRange(hsv, lower_color, upper_color)
    contours, _ = cv2.findContours(mask, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)
    if len(contours) > 0:
        largest_contour = max(contours, key=cv2.contourArea)
        M = cv2.moments(largest_contour)
        if M["m00"] != 0:
            cx = int(M["m10"] / M["m00"])
            cy = int(M["m01"] / M["m00"])
            cv2.drawMarker(frame, (cx, cy), (0, 0, 255),
cv2.MARKER_CROSS, markerSize=40, thickness=3)
            update_trajectory(cx, cy)
            draw_trajectory(frame)
```

```

end_time = time.time()
processing_time = end_time - start_time
detection_speed = 1 / processing_time if processing_time > 0 else 0
print(f"Processing Time: {processing_time:.4f} seconds")
print(f"Detection Speed: {detection_speed:.2f} FPS")
cv2.imshow('Video Stream', frame)
cv2.imshow('Color Mask', mask)

```

Цей фрагмент коду відповідає за обробку кожного кадру з відеопотоку. Основна задача цієї функції – відстеження об'єкта певного кольору на кадрі та відображення результатів в реальному часі. Кожен кадр спочатку перетворюється в колірний простір HSV, що спрощує виявлення кольорів. Далі виконується порівняння кольорів на кадрі з визначеним користувачем кольором, щоб створити маску, яка відокремлює області, які відповідають цьому кольору. Пошук контурів на масці дозволяє визначити форму та положення об'єкта.

Якщо знайдено контури, вибирається найбільший за площею, і на його основі визначається центр маси. Далі центр маси використовується для відображення позначки (хрестик) на кадрі у місці центру маси. Після цього оновлюється траєкторія руху об'єкта та відображається на кадрі. Також обчислюються час обробки кадру та швидкість виявлення (кадрів на секунду). Результати обробки виводяться у консоль, а також відображаються на екрані у вікнах "Video Stream" і "Color Mask" за допомогою OpenCV функції cv2.imshow().

Функція для запуску стриму та маски після вибору кольору

```

def start_stream():
    color = color_var.get()
    root.withdraw() # Скриваємо головне вікно
    cap = cv2.VideoCapture(0)
    while True:

```

```

ret, frame = cap.read()
process_frame(frame, color)
key = cv2.waitKey(1)
if key == ord('q') or key == 27: # 27 – код клавіши 'Esc'
    break
cap.release()
cv2.destroyAllWindows()
root.quit() # Завершуємо роботу програми

```

Цей фрагмент коду відповідає за запуск відеостріму та відображення маски певного кольору після вибору користувачем цього кольору у графічному інтерфейсі.

`color = color_var.get()`: Отримання вибраного користувачем кольору для відстеження.

`root.withdraw()`: Сховати головне вікно програми (Tkinter вікно).

`cap = cv2.VideoCapture(0)`: Відкриття відеопотоку з камери за допомогою OpenCV.

`while True::` Безкінечний цикл для обробки кадрів в реальному часі.

`ret, frame = cap.read()`: Зчитування кадру з відеопотоку.

`process_frame(frame, color)`: Виклик функції обробки кадру для відстеження об'єкту обраного кольору.

`key = cv2.waitKey(1)`: Очікування натискання клавіші для виходу з циклу. Код клавіші 'q' або 'Esc' (27).

`cap.release()`: Звільнення ресурсів відеопотоку.

`cv2.destroyAllWindows()`: Закриття всіх вікон OpenCV.

`root.quit()`: Завершення роботи програми Tkinter.

Ця функція відповідає за відображення в реальному часі відеопотоку з камери та результатів відстеження об'єкта обраним кольором. Коли користувач натискає 'q' або 'Esc', відеопотік завершується, і програма закривається.

Результат роботи розробленої програми для відстеження об'єктів у робочій зоні мобільного робота з візуалізації траєкторій переміщення об'єкта представлено на рисунку 3.4.

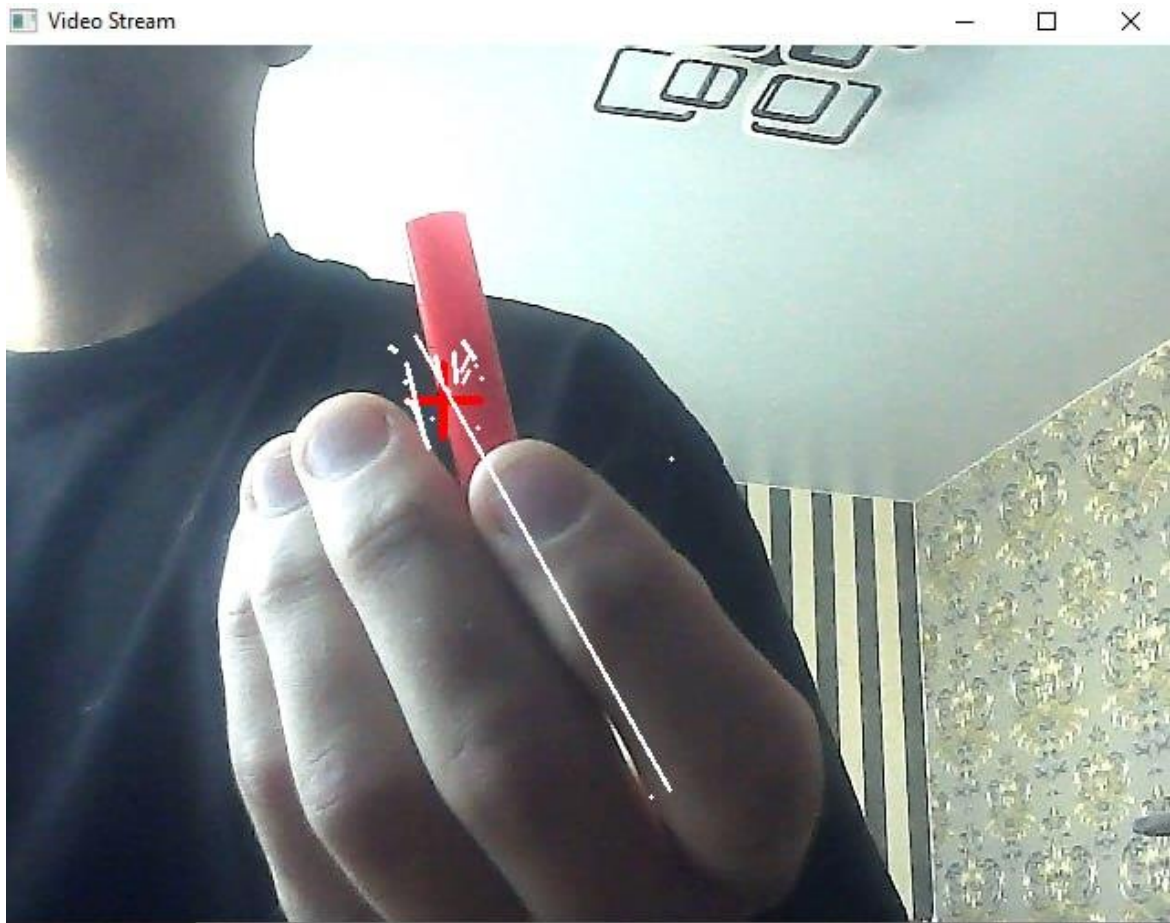


Рисунок 3.4 – Приклад роботи програми для відстеження об'єктів у робочій зоні мобільного робота з візуалізації траєкторій переміщення об'єкта

3.3 Проведення експерименту

Для проведення експерименту щодо впливу рівня освітленості на швидкість розрахунку обробки кадрів в програмі для відстеження об'єктів у робочій зоні мобільного робота з візуалізацією траєкторій переміщення об'єкта, спочатку необхідно створити контрольовані умови освітленості, що можна зробити за допомогою штучного освітлення або зміною природної освітленості за допомогою штор або жалюзі. Під час експерименту важливо

систематично змінювати рівень освітленості та фіксувати час обробки кадру та візуалізацію траєкторії переміщення об'єкта для кожного рівня освітленості. Рекомендується провести кілька повторень для кожного рівня освітленості для отримання більш точних результатів. Зібрані дані можна аналізувати, порівнюючи час обробки кадру та якість візуалізації траєкторії переміщення об'єкта при різних рівнях освітленості. Такий аналіз допоможе визначити оптимальний рівень освітленості для ефективної роботи програми в різних умовах освітлення.

Для проведення експерименту буде використовуватися ПК с наступними характеристиками: Intel(R) Core(TM) i5-9300H CPU 2.40 ГГц, RAM 16.0 ГБ, OS Windows 10 Pro 64-разрядная система. Для вимірювання рівня освітленості буде використана програма «Світломір» за посиланням <https://play.google.com/store/apps/details?id=com.lux.light.meter&hl=uk&gl=PT>. Результати, отримані при проведенні експерименту приведено в таблиці 3.1.

Таблиця 3.1 – Результати проведеного експерименту впливу рівня освітленості об'єкта на швидкість обробки (FPS) та час обробки кадрів (сек/ кадр) при відстеженні об'єкта колір жовтий

Освітленість (lux)	Швидкість обробки (FPS)	Швидкість обробки кадрів (сек/кадр)
230	201	0,0050
500	267	0,0050
800	310	0,0050
1100	399	0,0044
1400	465	0,0040
1700	500	0,0038
2000	575	0,0032
2300	632	0,0030
2600	743	0,0027
2900	894	0,0022
3100	1040	0,0010

Для зручності аналізу та візуалізації отриманих даних проведеного експерименту впливу рівня освітленості об'єкта на швидкість обробки (FPS)

та час обробки кадрів (сек/ кадр) представимо їх у вигляді наступних графіків: графік залежності швидкості обробки кадрів (FPS) від рівня освітленості (Lux) при відстеженні об'єкта колір жовтий на рисунку 3.5 та графік залежності швидкості обробки кадрів (сек/кадр) від рівня освітленості (Lux) при відстеженні об'єкта колір жовтий на рисунку 3.6, та для кожного графіку зробимо висновки відповідно до отриманих результатів проведеного експерименту.

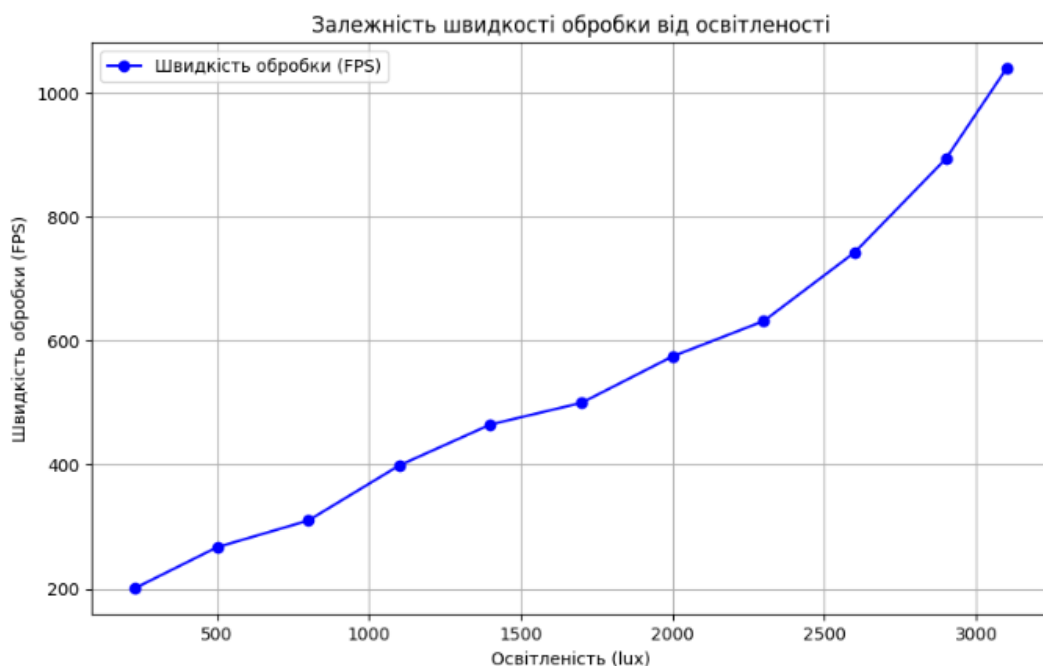


Рисунок 3.5 – Графік залежності швидкості обробки кадрів (FPS) від рівня освітленості (Lux) при відстеженні об'єкта колір жовтий

За отриманими результатами експерименту (рис. 3.5) можна зробити наступні висновки:

- існує чітка залежність між освітленістю та швидкістю обробки кадрів: зі збільшенням освітленості зростає і швидкість обробки кадрів;
- починаючи з освітленості близько 230 lux, швидкість обробки кадрів починає зростати, але цей зріст стає менш виразним при подальшому збільшенні освітленості;

– швидкість обробки кадрів значно зростає при освітленості від 230 lux до 500 lux, після чого зростання стає менш інтенсивним;

– найвищі значення швидкості обробки кадрів спостерігаються при освітленості від 2900 lux до 3100 lux, що вказує на високу швидкість обробки в умовах високої освітленості.

В цілому, залежність швидкості обробки кадрів від освітленості можна описати як неоднорідну, з більш вираженим зростанням швидкості при низькій освітленості та менш виразним при високій освітленості.

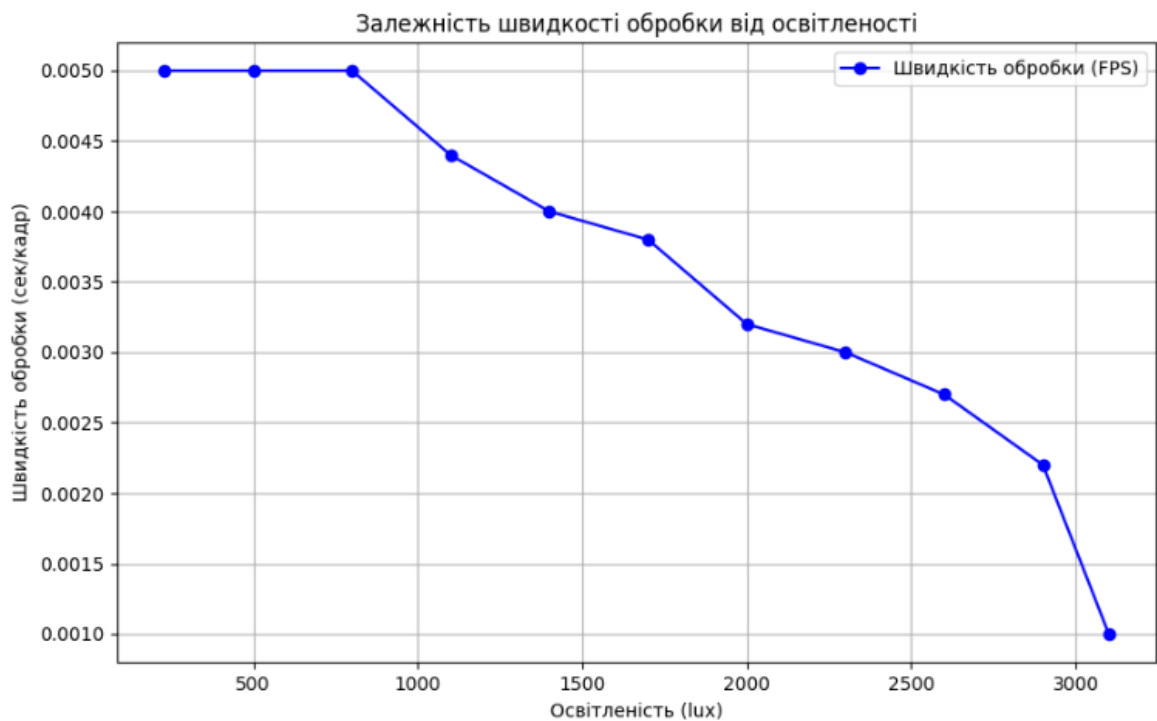


Рисунок 3.6 – Графік залежності швидкості обробки кадрів (сек/кадр) від рівня освітленості (Lux) при відстеженні об'єкта колір жовтий

За отриманими результатами експерименту (рис.3.6) можна зробити наступні висновки:

– існує зворотно пропорційна залежність між освітленістю та швидкістю обробки кадрів: зі збільшенням освітленості зменшується час обробки одного кадру;

– починаючи з освітленості близько 230 lux, швидкість обробки кадрів починає зменшуватись, але це зменшення стає менш виразним при подальшому збільшенні освітленості;

– найвищі значення швидкості обробки кадрів спостерігаються при освітленості від 2900 до 3100 lux, що вказує на найшвидшу обробку кадрів в умовах високої освітленості;

– залежність швидкості обробки кадрів від освітленості можна описати як неоднорідну, з більш вираженим зменшенням швидкості при низькій освітленості та менш виразним при високій освітленості.

Для забезпечення найвищої швидкості обробки кадрів рекомендується забезпечувати достатньо високий рівень освітленості в робочій зоні.

3.4 Охорона праці

Розробка програми відстеження об'єктів у робочій зоні мобільного робота вимагає врахування вимог охорони праці для забезпечення безпеки персоналу, який працює з програмним забезпеченням та апаратурою. Основними аспектами охорони праці в даному проекті є зниження ризику травматизму та небезпеки для операторів під час експлуатації та обслуговування системи відстеження. Проведемо розрахунок електричної безпеки. Струм короткого замикання (I_{sc}) можна визначити за формулою:

$$I_{sc} = V / Z, \quad (3.1)$$

де V – номінальна напруга електричного обладнання (наприклад, 230 В для однофазного обладнання в домашніх умовах);

Z – еквівалентний імпеданс системи (сума імпедансів джерела живлення та обладнання).

Нехай номінальна напруга $V = 230$ В, а еквівалентний імпеданс $Z = 0,2$ Ом. Проведемо розрахунок:

$$I_{sc} = 230 / 0,2 = 1150 \text{ А.}$$

Цей розрахунок допоможе визначити необхідність захисних пристроїв і перевірити, чи вони відповідають вимогам електробезпеки. Наступним етапом проведемо розрахунок механічної безпеки. При розробці програми важливо враховувати механічні навантаження на обладнання, зокрема при кріпленні камер або сенсорів до рухомих платформ. Проведемо дані розрахунки за формулою:

$$F = M \cdot g / r, \quad (3.2)$$

де F – максимальне віджимне навантаження;

M – маса обладнання;

g – прискорення вільного падіння (приблизно $9,81$ м/с²);

r – радіус підшипника.

Нехай маса обладнання $M = 5$ кг, радіус підшипника $r = 0,1$ м, тоді:

$$F = 5 \cdot 9,81 / 0,1 = 490,5 \text{ Н.}$$

Цей розрахунок допомагає визначити необхідну міцність кріпильних елементів та забезпечити безпеку операторів під час роботи з обладнанням. Проведення таких розрахунків є критично важливим для забезпечення високого рівня охорони праці під час розробки та експлуатації програми відстеження об'єктів у робочій зоні мобільного робота. Вони допомагають ідентифікувати потенційні небезпеки та застосовувати необхідні заходи для їх запобігання [14].

ВИСНОВКИ

В даній кваліфікаційній роботі проведено аналіз систем комп'ютерного зору, методів відстеження об'єктів у робочій зоні мобільного робота, проаналізовано алгоритми роботи систем комп'ютерного зору та аналогічні рішення.

Після проведеного аналізу було розроблено структуру програми. Проведено обґрунтування та вибір середовища розробки. Проведено математичне представлення методу відстеження об'єктів у робочій зоні мобільного робота. Розроблено загальний алгоритм роботи програми та алгоритм відстеження об'єктів. Проведено розрахунок швидкості обробки кадрів при відстеження об'єктів.

Розроблено НМІ інтерфейс користувача та реалізовано функції відстеження об'єктів. Проведено експериментальні дослідження.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. ДСТУ 3008-15. Документація. Звіти у сфері науки та техніки. структура та правила оформлення. – Введ. 2015-06-22. – К. Держстандарт України, 2017 – 29 с.
2. Невлюдов, І.Ш. Дипломне проектування для студентів усіх форм навчання спеціальностей 151 «Автоматизація та комп'ютерно-інтегровані технології» [Текст]: навч. посіб. / І.Ш. Невлюдов, А.О. Андрусевич, О.В. Токарева, Г.В. Пономарьова. – Київ-58, пр. Космонавта Комарова, 1, 2016. – 320 с.
3. Методичні вказівки з підготовки кваліфікаційної роботи для здобувачів першого (бакалаврського) рівня вищої освіти денної і заочної форми навчання спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» освітньої програми «Автоматизація та комп'ютерно-інтегровані технології» / Упоряд.: І.Ш. Невлюдов, О.І. Филипченко, О.В. Токарева, С.П. Новоселов, О.В. Сичова. Харків: ХНУРЕ, 2023. 64 с.
4. Мозговенко, А. А., & Зінов'єва, О. Г. (2022). Аналіз методів комп'ютерного зору в задачах ідентифікації осіб у відеопотоці. Таврійський науковий вісник. Серія: Технічні науки, (3), 56-62.
5. Гороховатський, В. О., & Гадецька, С. В. (2020). Статистичне оброблення та аналіз даних у структурних методах класифікації зображень.
6. Лесюк, А. М., & Яцишин, С. П. (2020). Комп'ютерний зір та його застосування. Editorial board, 530.
7. Кондратьєв, С. Б., Костенко, В. Л., & Ядрова, М. В. (2021). Метод контурів для позиціонування об'єктів в мобільних системах комп'ютерного зору.
8. Запорожець В. А. Розроблення автоматизованої системи для аналізу об'єктів робочого простору MP Festo Robotino: пояснювальна записка до

атестаційної роботи здобувача вищої освіти на другому (магістерському) рівні, спеціальність 151 - Автоматизація та комп'ютерно-інтегровані технології / В. А. Запорожець; М-во освіти і науки України, Харків. нац. ун-т радіоелектроніки – Харків, 2021 – 81 с.

9. Тімков, Ю. Ю., & Рувінська, В. М. (2021). Вдосконалений метод трекінгу об'єктів для систем відеоспостереження.

10. Шевченко К. О. Розробка безконтактного методу ідентифікацій виробів на промисловій лінії : пояснювальна записка до кваліфікаційної роботи здобувача вищої освіти на другому (магістерському) рівні, спеціальність 151 – Автоматизація та комп'ютерно-інтегровані технології / К. О. Шевченко ; М-во освіти і науки України, Харків. нац. ун-т радіоелектроніки. – Харків, 2022. – 81 с.

11. Невлюдов, І., & Гурін, Д. (2023). Аналіз існуючих алгоритмів розпізнавання безлічі об'єктів на зображенні та відео потоці.

12. Чуть, М. О. Дослідження методів адаптивного керування інтелектуальною робототехнічною платформою / М. О. Чуть // Автоматизація та приладобудування = Automation and Development of Electronic Devices (ADED'2017) : зб. студ. наук. ст. – Харків : ХНУРЕ, 2017. – С. 11–15.

13. Мамонько Д. В. Удосконалення методу прокладення шляху мобільної платформи в невизначеному просторі : пояснювальна записка до атестаційної роботи здобувача вищої освіти на другому (магістерському) рівні, спеціальність 151 Автоматизація та комп'ютерно-інтегровані технології / Д. В. Мамонько ; М-во освіти і науки України, Харків. нац. ун-т радіоелектроніки. – Харків, 2020. – 78 с.

14. Охорона праці. // Сайт GCC. URL: <https://gc.ua/uk/oxorona-praci-v-ofisi-vimogi-do-robochogo-miscya-ofisnogo-pracivnika/> (дата звернення: 04.06.2024).