Faculty of Electrical and Electronic Engineering Technology

INTEGRATION OF COMPUTER VISION AND MOTION CONTROL FOR SORTING ROBOTS WITH IOT INTEGRATION

FARIQ ANUGRAWAN BIN NUR MUHAMAD

Bachelor of Electrical Engineering Technology (Industrial Automation & Robotics) with Honours

2023

INTEGRATION OF COMPUTER VISION AND MOTION CONTROL FOR SORTING ROBOTS WITH IOT INTEGRATION

FARIQ ANUGRAWAN BIN NUR MUHAMAD

A project report submitted
in partial fulfillment of the requirements for the degree of
Bachelor of Electrical Engineering Technology (Industrial Automation & Robotics) with Honours

Faculty of Electrical Technology and Engineering

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2023

UNIVERSITI TEKNIKAL MALAYSIA MELAKA
FAKULTI TEKNOLOGI KEJUTERAAN ELEKTRIK DAN ELEKTRONIK

BORANG PENGESAHAN STATUS LAPORAN
PROJEK SARJANA MUDA II

Tajuk Projek : Integration Of Computer Vision And Motion Control For Sorting Robots With IOT Integration.

Sesi Pengajian : 2023/2024

Saya Fariq Anugrawan Bin Nur Muhamad mengaku membenarkan laporan Projek Sarjana Muda ini disimpan di Perpustakaan dengan syarat-syarat kegunaan seperti berikut:

1. Laporan adalah hakmilik Universiti Teknikal Malaysia Melaka.
2. Perpustakaan dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan laporan ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. Sila tandakan (✓):

☐ **SULIT*** (Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)

☐ **TERHAD*** (Mengandungi maklumat terhad yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan)

☑ **TIDAK TERHAD**

Disahkan oleh:

(TANDATANGAN PENULIS)
Alamat Tetap: No 40, Jalan Mesra Indah 1, Taman Mesra Indah, 82000 Pontian, Johor

MOHD RAZALI BIN MOHAMAD SAPIEE
PENSYARAH KANAN
Jabatan Teknologi Kejuruteraan
Fakulti Teknologi dan Kejuruteraan Elektrik
Universiti Teknikal Malaysia Melaka

Tarikh: 12/1/2024

Tarikh: 14/1/2024

*CATATAN: Jika laporan ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali tempoh laporan ini perlu dikelaskan sebagai SULIT atau TERHAD.

DECLARATION

I declare that this project report entitled “Integration of Computer Vision and Motion Control for Sorting Robots with IOT Integration” is the result of my own research except as cited in the references. The project report has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature	:	
Student Name	:	Fariq Anugrawan Bin Nur Muhamad
Date	:	12/1/2024

APPROVAL

I hereby declare that I have checked this project report and in my opinion, this project report is adequate in terms of scope and quality for the award of the degree of Bachelor of Electrical Engineering Technology (Industrial Automation & Robotics) with Honours.

Signature :

Supervisor Name : Ts. Mohd Razali Bin Mohamad Sapiee

Date : 14/1/2024

DEDICATION

To my beloved mother, Susni Elifida, and father, Nur Muhamad Bin Bustamam

ABSTRACT

The development of robotic systems has increased significantly in recent years, and one of the areas that has shown great potential is the use of robots for sorting tasks. Sorting robots can significantly increase the efficiency and accuracy of industrial processes, particularly in warehouses and logistics centres. This abstract focuses on the design and implementation of a sorting robot that integrates computer vision, motion control, and IoT technology. The sorting robot is equipped with a camera that captures images of the objects to be sorted. Computer vision algorithms are used to process the images and identify the patterns. The robot's motion control system moves the arm and gripper to pick up the objects and place them in the appropriate location. The integration of IoT enables the robot to not only acquire data from the sorting operation but also facilitates manual control of the robot, fostering seamless communication with sensor and systems within the warehouse. This integration enhances the sorting process, contributing to overall operational efficiency. The system's hardware consists of a Raspberry Pi and a robotic arm with a gripper. The software includes Python for computer vision and motion control, and MQTT for IoT communication. The sorting robot has the potential to significantly improve sorting processes in industrial settings, reducing the need for manual labour and increasing efficiency. With further development and optimization, this technology could have wide-ranging applications across various industries, including manufacturing, logistics, and e-commerce.

ABSTRAK

Pembangunan sistem robotik telah meningkat dengan ketara dalam beberapa tahun kebelakangan ini, dan salah satu bidang yang telah menunjukkan potensi besar ialah penggunaan robot untuk pengasingan barang. Robot pengasingan barang boleh meningkatkan kecekapan dan ketepatan proses perindustrian dengan ketara, terutamanya di gudang dan pusat logistik. Abstrak ini memfokuskan pada reka bentuk dan pelaksanaan robot pengasingan yang mengintegrasikan penglihatan komputer, kawalan pergerakan dan teknologi IoT. Robot pengasingan dilengkapi dengan kamera yang menangkap imej objek yang hendak diasingkan. Algoritma penglihatan komputer digunakan untuk memproses imej dan mengenal pasti objek. Sistem kawalan pergerakan robot menggerakkan lengan dan pencengkam untuk mengambil objek dan meletakkannya di dalam bekas yang sesuai. Penyepaduan IoT membolehkan robot berkomunikasi dengan mesin dan sistem lain dalam gudang, mengoptimumkan proses pengasingan dan meningkatkan kecekapan keseluruhan. Perkakasan sistem terdiri daripada Raspberry Pi dan lengan robot dengan pencengkam. Perisian ini termasuk Python untuk penglihatan komputer dan kawalan gerakan, dan MQTT untuk komunikasi IoT. Robot pengasingan mempunyai potensi untuk meningkatkan proses pengasingan dengan ketara dalam tetapan industri, mengurangkan keperluan untuk buruh manual dan meningkatkan kecekapan. Dengan pembangunan dan pengoptimuman selanjutnya, teknologi ini boleh mempunyai aplikasi yang meluas merentasi pelbagai industri, termasuk pembuatan, logistik dan e-dagang.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my gratitude to my supervisor, Ts. Mohd Razali Bin Mohamad Sapiee for their precious guidance, words of wisdom and patience throughout this project.

I am also indebted to Universiti Teknikal Malaysia Melaka (UTeM) and my mom, dad, and sisters for their financial support through my study which enables me to accomplish the project. Not forgetting my fellow colleague, my housemate for the willingness of sharing his thoughts and ideas regarding the project.

My highest appreciation goes to my parents and family members for their love and prayer during the period of my study.

Finally, I would like to thank all the staffs at the FTKE, fellow colleagues and classmates, the faculty members, as well as other individuals who are not listed here for being co-operative and helpful.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

IoT	-	Internet of Things
MP	-	Megapixel
GB	-	Gigabyte
DL	-	Deep Learning
DOF	-	Degree of Freedom
DH	-	Denavit–Hartenberg parameter
GUI	-	Graphical User Interface

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Background

In recent years, the use of robots in industrial settings has increased rapidly due to their ability to improve efficiency and reduce costs. One area where robotics has shown promise is in the field of sorting tasks, particularly in logistics centre and warehouses. Sorting robots have the potential to significantly increase the accuracy and speed of the sorting process, resulting in increased productivity and reduced labour costs.

The integration of computer vision and motion control for sorting robots with IoT integration in the BDP aligns with Sustainable Development Goal 9 (SDG 9) which focuses on building resilient infrastructure, promoting inclusive and sustainable industrialization, and fostering innovation. By leveraging advanced technologies, such as computer vision and IoT, the project contributes to the development of smarter and more efficient industrial processes. This integration improves the productivity and effectiveness of sorting systems, enabling industries to optimize their operations and reduce waste. Additionally, the use of automation reduces the reliance on manual labour, promoting safer working conditions. By enhancing industrial processes and promoting innovation, the project supports the goal of sustainable industrialization and economic growth while ensuring environmental sustainability. SDG 9 emphasizes the importance of technological advancements in driving inclusive and sustainable industrial development, and this BDP exemplifies the application of such technologies towards achieving this goal.

1.2 Relevance to Sustainable Development Goal 9: Industry, Innovation, and Infrastructure.

This project "Integration of Computer Vision and Motion Control for Sorting Robots with IoT Integration" aligns with the United Nations' Sustainable Development Goal (SDG) 9: Industry, Innovation, and Infrastructure. SDG 9 focuses on building resilient infrastructure, promoting inclusive and sustainable industrialization, and fostering innovation. This project encompasses the integration of computer vision, motion control, and IoT technologies to enhance sorting robots' efficiency and effectiveness in industrial processes. By embracing SDG 9, this project contributes to sustainable infrastructure development, technological advancements, and the promotion of inclusive and sustainable industrial practices. The successful implementation of this project signifies a commitment to fostering innovation, improving industrial processes, and contributing to a sustainable future.

1.3 Problem Statement

A sorting robot with computer vision, motion control, and IoT integration aims to address the need for efficient and accurate sorting processes in industrial settings. Manual sorting processes are often time-consuming, labour-intensive, and prone to errors, which can lead to increased costs and decreased productivity. Sorting robots have the potential to address these issues by automating the sorting process and reducing the need for manual labour. However, the development of sorting robots that can accurately identify and sort objects in a variety of settings remains a challenge. This sorting robot system seeks to overcome these challenges by integrating computer vision, motion control, and IoT technology to create a highly efficient and accurate sorting system.

1.4 Project Objective

The objective of the sorting robot with computer vision, motion control, and IoT integration is to create a highly efficient and accurate sorting system for industrial settings. The specific objectives of the system are:

a) To integrate computer vision and IoT technology into the sorting robot.

b) To develop a motion control system that can move the robot's arm and gripper to pick up and sort objects.

c) To test the system's performance in sorting objects with high accuracy.

By achieving these objectives, the sorting robot system aims to significantly improve the efficiency and accuracy of sorting processes in various industries, including manufacturing, logistics, and e-commerce. The ultimate goal is to create reliable and cost-effective technology that can reduce labour costs and increase productivity in industrial settings.

1.5 Scope of Project

The scope of this project are as follows:

i) Design and development of a functional prototype system for sorting objects in an industrial setting.

ii) Software components include computer vision algorithms, motion control algorithms, and IoT communication protocols.

iii) Testing and optimization of the system's performance in identifying and sorting objects.

iv) Providing a functional prototype system that can be further developed and integrated into larger logistics systems in the future.

v) Not including development of a complete end-to-end logistics solution or integration with an existing warehouse management system.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

As industries continue to face challenges in the sorting process, there is a growing interest in the potential benefits of using a sorting robot system with computer vision, motion control, and IoT integration. In this context, the literature review will focus on identifying and analyzing existing literature on robotic sorting systems and their applications in industrial settings. The specific objectives of the literature review will be to review the current state-of-the-art technologies and methodologies for computer vision, motion control, and IoT integration in sorting robot systems. Additionally, the review will identify the limitations and challenges faced by current sorting robot systems and propose potential solutions and areas for future research. Overall, the literature review will provide a comprehensive overview of the current state of research and development in the field of sorting robot systems. By highlighting the potential benefits of integrating computer vision, motion control, and IoT technology in these systems and identifying the limitations and challenges that need to be addressed, the literature review will serve as a foundation for future research and development in this area.

2.2 Computer Vision

Computer vision is a field of artificial intelligence that focuses on enabling machines to recognize, analyze, and interpret visual information from the world around them [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12]. In the context of sorting robots with computer vision, motion control, and IoT integration, computer vision plays a crucial role in enabling the system to identify and classify different objects for sorting.

Computer vision algorithms are used to analyze images captured by a camera and extract important features that allow the system to identify objects. These features can include shape, color, texture, and other characteristics that distinguish one object from another [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12]. Once objects are identified and classified, the system can use this information to determine the optimal sorting destination for each object.

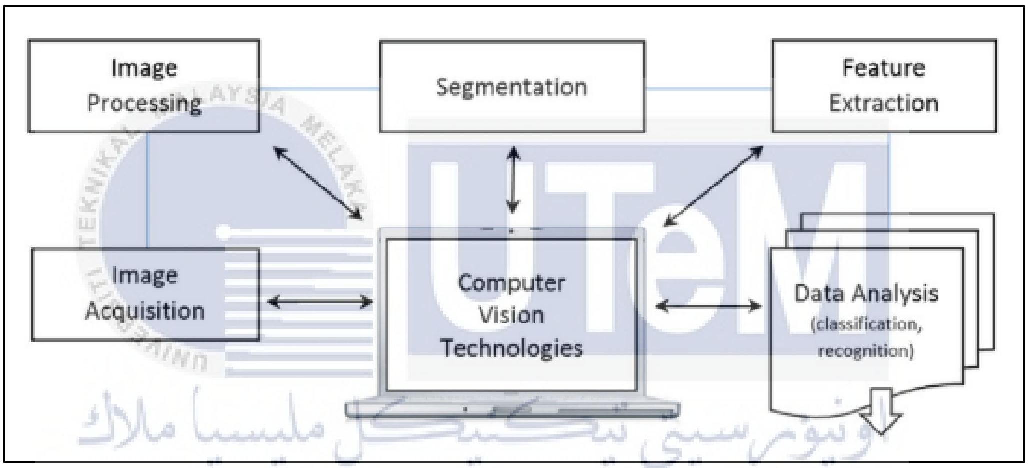

Figure 2.1 Computer vision system technologies. [13]

From Xuan Duong *et al.*,[5] the study focuses on developing a control system for a 6DOF robot arm utilizing deep learning-based computer vision techniques for object recognition and grasp tasks. By incorporating DL1, DL2, and DL3 models, the system recognizes voice commands, identifies objects through camera input, and controls the robot's motion accordingly. The experimental results demonstrate the effectiveness of computer vision techniques with a recognition accuracy of 92%.

The research by Abu Salman Shaikat *et al.*,[6] focuses on utilizing Arduino Nano and computer vision (CV) methods for efficient movement and placement of products in industrial operations. Various pick and place robot systems have been implemented, including an Android mobile phone-based CV technique for waste separation and an intelligent colour and height sorting pick and place robot using the Haar Cascade method. The system design incorporates servo motors for motion control, a camera for image processing using OpenCV, and a cascade classifier for object detection. The results demonstrate the effectiveness and efficiency of the system, with a success rate of nearly 99%.

The utilization of computer vision in the field of robotics and automation is gaining significant momentum, as highlighted in [3]. This research from Md Abdullah-Al-Noman *et al*.,[3] focuses on the development of a computer vision-based automated system for colour sorting, shape detection, and size identification of objects. The system incorporates a robotic arm controlled by four high-torque servo motors connected to an Arduino Mega. A PixyCMU camera sensor is employed to detect object colours, while OpenCV libraries in Python are utilized for shape and size identification. The system successfully detects and manipulates objects based on their colour, shape, and size, thereby showcasing its efficiency and effectiveness. The proposed system holds great potential for reducing labour costs and enhancing productivity in industrial applications, aligning with the objectives of integrating computer vision and motion control for sorting robots with IoT integration and SDG 9 agenda which is to efficiency and accuracy, considering energy-efficient and precise motion control mechanisms [14].

The presented research by Wenchang Zhang *et al*.,[2] focuses on the utilization of machine vision in industrial fields, specifically for high-speed pick-and-place tasks using a delta robot. The proposed system integrates a vision module and a motion control module to achieve intelligent sorting tasks. The vision module comprises a computer, image acquisition card, and CCD camera, while the motion control module utilizes servo motors and a synchronous conveyor. LabVIEW software is employed for system development. Image acquisition and processing techniques are applied to analyse and recognize target objects. Dynamic target tracking is achieved by synchronizing the servo motor with the conveyor. The control strategy of the delta robot involves a specific path and acceleration profile. The velocity and acceleration of the conveyor are adjusted based on the density of objects to optimize sorting efficiency. The position of the next-picking object is calculated in real-time. Experimental results demonstrate the efficiency and effectiveness of the proposed vision-based control strategy for high-speed pick-and-place tasks. The research provides valuable insights into automation in industrial manufacturing and showcases the potential of machine vision and motion control integration.

The journal from Institute of Electrical and Electronics Engineers [4] highlights the adaptation of AI in the field of computer vision and its importance in modern manufacturing, particularly in robotics. The objective of the project is to develop a robotic manipulator with the capability to lift, carry, and unload objects at specific locations. The results demonstrate the functionality of the vision system, which detects the colour of objects and sends signals to an Arduino controller. Based on the colour detection, the robotic arm executes preprogrammed commands to grip, lift, rotate, and place the object. The claw then returns to its original position for the next object. The paper concludes by emphasizing the successful implementation of a 6 DOF articulated robotic manipulator and a 3D vision system. The robotic arm performs multifunctional tasks with the assistance of computer vision, while the 3D vision system recognizes object deformation and spatial coordination. The vision system utilizes a camera and computer vision algorithms to detect objects and their distance from the end-effector. It is noted that the vision system requires separate computing hardware capable of processing complex visual data. The research showcases the integration of computer vision and robotics, highlighting their potential for various industrial applications.

The journal from Mohamed El-sayed *et al*.,[9] provides a study on smart solutions in industrial material flow handling systems. It discusses a prototype design for a hexagonal omnidirectional wheeled conveyor and proposes two camera tracking systems: one using conventional image processing techniques and the other based on deep learning (DL). Experimental results show that DL-based System II:YOLOv5 achieves the best performance in package detection and positioning. The camera tracking system is deemed sufficient for closed-loop control, eliminating the need for additional sensors. The study highlights advancements in camera-based tracking systems, offering improved efficiency and performance in industrial material handling.

The study from Cong *et al*.,[7] introduces a machine vision system with two robot arms for object classification and sorting based on shape and size. The system utilizes a hierarchical control system with a master and two slave controllers. Various algorithms are employed for object recognition and noise elimination. The experimental evaluation demonstrates the system's effectiveness and accuracy in sorting tasks, offering potential improvements in industrial production line performance. The mechanical models are designed using Autodesk Inventor and fabricated using a 3D printer. The vision system incorporates a Raspberry Pi Camera Module. Overall, the system provides efficient sorting capabilities with low energy consumption and cost-effective development.

This journal paper from Boysen *et al*.,[10] introduces a computer vision-based solution for order consolidation in robotized sorting systems used by retailers to streamline fulfilment processes. By incorporating computer vision technology, the system enhances the capabilities of autonomous mobile robots in identifying and manipulating stock keeping units (SKUs). The paper highlights the challenges of real-time decision-making in this context and proposes a multiple-scenario approach that leverages computer vision data for efficient scheduling. Additionally, the paper discusses the potential benefits of computer vision in improving sorting performance, optimizing order-to-collection point assignments, and enhancing overall system throughput. The findings underscore the significance of integrating computer vision into robotized sorting systems to enable faster, more accurate, and adaptive order fulfilment processes.

Journal from Lu *et al*.,[11] introduces waste sorting using computer vision (CV) technology. It highlights the advantages of CV-based sorting systems over traditional approaches and reviews the literature on the evolution, current status, and challenges of applying CV for waste sorting. The journal discusses various computer vision algorithms used for waste sorting, including traditional machine learning algorithms and deep learning algorithms.

The paper from Kjeldsen *et al*.,[15] explore the QR codes' impact on omnichannel customer experience and purchase intention offers insights applicable to the integration of computer vision, motion control, and IoT in sorting robots. The study underscores the nuanced relationship between digitalized, personalized QR-code information, and experiential value, suggesting a delicate balance. While less digitalized features enhance experiential value, the reverse holds for purchase intention. This dichotomy aligns with the complexities of customer experience (CX) in the IoT realm. In the context of sorting robots, understanding the type of information triggering experiential value versus purchase intention becomes pivotal. The findings emphasize the need for a strategic approach in implementing IoT features, mirroring the delicate equilibrium identified in QR code impacts. This resonates with the challenges faced in crafting IoT integration strategies for sorting robots, where optimizing user experience and operational outcomes requires a nuanced understanding of the impact of digitalized information on both experiential value and purchase intention.

Table 2.1 Literature review matrix focusing on computer vision.

Study	Key Finding	Aim	Results
[5]	The article focuses on building a control system for a 6DOF robot arm performing object pick-up tasks using computer vision techniques and deep learning models.	To develop a control system for a robot arm based on computer vision and deep learning.	The simulation and experiment results demonstrate the correctness and reliability of computer vision techniques, with an object recognition accuracy of 92%.
[6]	The researchers developed a pick and place robot using computer vision techniques, specifically a simple computer vision technique for robotics implemented in Android mobile phones.	To address the issue of separating degradable and non-degradable wastes in streets using a pick and place robot.	The robot successfully separated objects based on their degradability, reducing the risk of communicable diseases.
[3]	The study presents the design and	To design an intelligent system	The proposed system successfully detects

	development of a computer vision-based automated system for sorting colors, detecting shape and size, and picking and placing objects.	that can sort colors and detect object shape and size using computer vision techniques.	object colors, shapes, and sizes, picks up objects, and places them in specific locations.
[2]	The research proposes a vision-based control strategy for high-speed pick-and-place tasks on a delta robot, combining machine vision and motion control modules.	To develop a control system for a delta robot that can perform high-speed sorting tasks using vision-based guidance.	The system successfully tracks and sorts dynamic objects on a moving conveyor using machine vision techniques, enabling efficient pick-and-place operations.

2.3 Motion Control

Motion control refers to the technology and techniques used to control the movement of a robotic arm and gripper in a sorting robot system. In the context of sorting robots with computer vision, motion control, and IoT integration, motion control is responsible for enabling the robot to move and manipulate objects in the workspace based on information provided by the computer vision system.

Motion control algorithms use data from the computer vision system to determine the position and orientation of the objects that need to be sorted [4], [5], [16], [17]. Based on this information, the robot arm can be programmed to move and manipulate the gripper in a way that allows it to pick up and move each object to its designated sorting destination.

Advanced motion control techniques can optimize the movement of the robot arm, reducing the time required for sorting and increasing the system's overall efficiency [17]. These techniques can include predictive motion control, which uses machine learning algorithms to predict the motion required for a given task and optimize the robot's movement accordingly.

Overall, motion control is a critical component of any sorting robot system that aims to sort objects accurately and efficiently in an industrial setting. By enabling the robot to move and manipulate objects based on information from the computer vision system, motion control technology plays a key role in ensuring that the sorting process is both accurate and efficient.

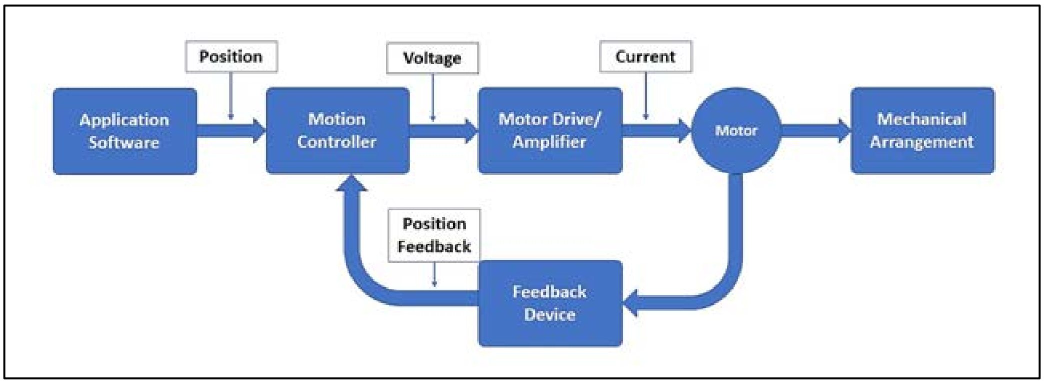

Figure 2.2 Components of motion control system. [18]

As in journal by Xuan Dong *et al*.,[5] The robot control problem is designed for different tasks such as controlling position, trajectory, or performing tasks of picking and dropping objects. It focuses on building the control system of the 6 DOF robot arm performing object pick-up task based on computer vision technique using several deep learning models. Object information from the ML output is the target for the DL2 model to identify through the camera and available learning data. The rest of the ML output combined with recognized object information gathered as input to the DL3. DL3 are the value of the joint variables that control the robot to perform the task as required.

The study from Aravinthkumar T *et al*.,[17] emphasizes the significance of articulated robots due to their advantages in terms of workspace and application versatility. The research proposes a methodology for trajectory generation using S-Curve profiles, which enable smooth and efficient motion. Results validate the effectiveness of the proposed approach through forward and inverse kinematics calculations, demonstrating its suitability for motion control in articulated industrial robot arms. This study contributes to advancing the field of motion control and its vital role in optimizing manufacturing processes.

From Asif *et al*.,[16] study introduces the concept of locomotion and the factors influencing the movement of a rigid body. It discusses forward kinematics, where the position of any point can be determined based on the length of each link and the angle of each joint. Inverse kinematics calculations are performed to find the joint values for a given position and orientation. The paper proposes a modified DH convention for forward kinematics and employs kinematics decoupling for inverse kinematics, simplifying the problem. The effectiveness of the proposed approach is validated through successful trajectory tracking in a test scenario.

The study from Taghizadeh *et al.*,[19] explore on PWM-driven servo-pneumatic systems offers insights applicable to the integration of computer vision and motion control in sorting robots with IoT. The study accentuates the impact of pneumatic circuit design on system behavior, enhancing control performances through a simplified circuit with a single fast switching valve. While their focus is on pneumatic systems, the principle of optimizing control mechanisms is transferable. In the context of sorting robots, the efficient motion control achieved by a well-designed pneumatic circuit resonates with the need for precise control in sorting operations. The study's emphasis on linear controllers and the quasi-linear behavior resulting from the proposed circuit aligns with the broader goal of achieving controlled and predictable motions, vital in sorting tasks. The findings suggest that simplified circuits and linear control strategies can improve the overall efficiency of motion control systems, a principle that could be relevant in the development of sorting robots with integrated IoT technologies.

The journal from Institute of Electrical and Electronics Engineers [4] highlights the significance of motion control in modern manufacturing and introduces a robotic arm as a programmable device capable of performing tasks similar to a human arm. The objective of the project is to enhance the manipulator's motion control capabilities for lifting, carrying, and unloading objects at specified locations. The paper describes the functionality of a vision system that enables the arm to detect object colors, execute precise movements, and perform actions such as gripping, lifting, rotating, and placing objects. The authors present their experience with a 6 DOF articulated robotic manipulator and a 3D vision system that not only recognizes objects but also calculates their spatial coordination, allowing for accurate manipulation. The contributions of the study include advancements in motion control techniques, facilitated by computer vision, to improve the object manipulation capabilities of robotic arms.

Table 2.2 Literature review matrix focusing on motion control.

Study	Key Finding	Aim	Results
[17]	Forward and inverse kinematics calculations are verified using Roboanalyser software.	To propose implementation methods for forward and inverse kinematics of a robotic arm.	The proposed methodology using DH parameters, Euler angle convention, and kinematic decoupling successfully relates the robot arm's base to its TCP. The calculations are verified through simulation and experimental results.
[16]	Inverse kinematics calculations involve finding joint values for a given position and orientation.	To explain the process of solving inverse kinematics for a robotic arm.	The inverse kinematics calculations for Comau NM45 robot arm involve finding joint angles for a given position and orientation. The calculations involve geometric and algebraic approaches.

[17]	Articulated robots have advantages in terms of workspace and versatility.	To explain the benefits of using articulated type configurations in industrial robots.	Articulated robots provide a spherical workspace, covering more volume, and are suitable for various applications. S-Curve profile is selected for smooth motion control, providing the shortest path time.
[16]	Forward kinematics determines the position of a rigid body based on link lengths and joint angles.	To describe the connection between locomotion and forward kinematics.	Forward kinematics uses transformation matrices to calculate the position of a point in a rigid body based on link lengths and joint angles.

2.4 Internet of Things (IoT)

IoT (Internet of Things) refers to the interconnectivity of physical objects, devices, and systems through the internet. In the context of sorting robots with computer vision, motion control, and IoT integration, IoT technology plays a crucial role in enabling the system to communicate and share data with other devices and systems in the industrial environment [8].

IoT technology allows the sorting robot system to collect and share data from various sources, including sensors, cameras, and other devices. This data can be used to optimize the sorting process by providing real-time information about the status of the sorting system and its environment. For example, sensors can provide information about the temperature and humidity in the sorting area, which can be used to optimize the performance of the sorting robot system.

IoT technology can also be used to enable remote monitoring and control of the sorting robot system. This allows operators to monitor the system from a remote location and adjust as needed to optimize the performance of the system.

Overall, IoT technology is a critical component of any sorting robot system that aims to optimize the sorting process in an industrial setting. By enabling the system to collect and share data with other devices and systems, IoT technology can help to improve the accuracy, efficiency, and overall performance of the sorting robot system.

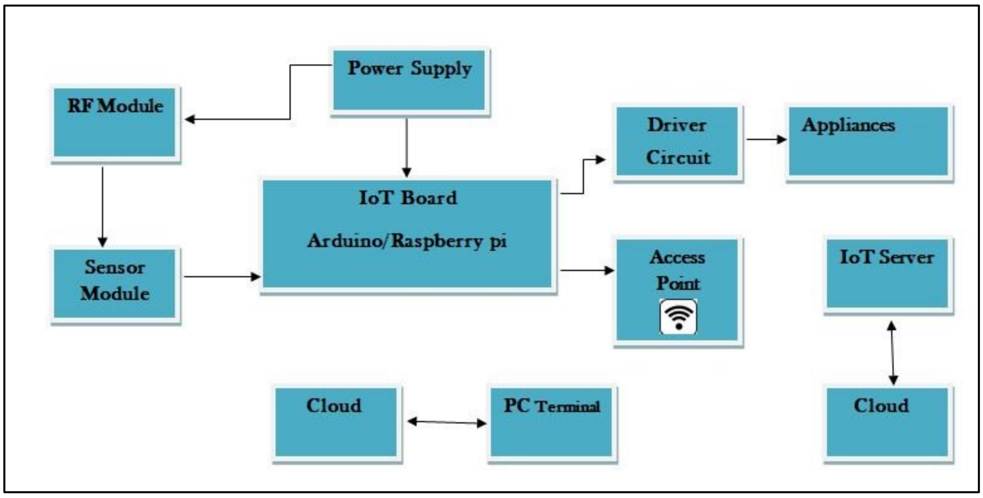

Figure 2.3 Block diagram of IoT. [20]

The study from Kareemullah *et al.*,[21] explores the development and implementation of a robotic arm controlled using an IoT application. The study focuses on designing a system that allows users to remotely control a robotic arm using a smartphone or web-based application. The integration of IoT technologies enables real-time communication and control, enhancing the usability and versatility of the robotic arm. This article presents several key contributions to the field. Firstly, it provides a comprehensive design and implementation process for developing a robotic arm controlled via an IoT application. The study outlines the hardware components, such as sensors and actuators, as well as the software architecture required for seamless communication and control. Secondly, the authors emphasize the development of an intuitive IoT application that enables users to remotely interact with the robotic arm. The application offers a user-friendly interface through which users can control the arm's movement, grasp objects, and perform various tasks. Thirdly, the integration of IoT technologies facilitates real-time communication between the robotic arm and the user. The article describes the protocols and mechanisms used to ensure low-latency and reliable data transmission, enhancing the system's responsiveness and accuracy. Lastly, the study addresses the usability and accessibility of the system by providing a platform-independent solution accessible through smartphones or web browsers. These contributions collectively contribute to advancing the field of robotic arm control using IoT applications, paving the way for further developments and applications in the field.

The journal from Rattanapoka *et al.*,[22] introduces an MQTT-based IoT cloud platform with Node-RED's flow design, streamlining the setup of IoT platforms for developers. Utilizing the MERN stack, the web application enables users to effortlessly create and manage MQTT broker instances, Node-RED instances, and database instances (MongoDB and InfluxDB). Employing Docker for efficient container management, the platform eliminates the need for users to configure servers and services independently. The demonstrated temperature IoT monitoring system showcases the platform's effectiveness in supporting IoT application development. This user-friendly solution minimizes the complexities associated with IoT platform setup, making it an accessible and efficient tool for developers working on IoT systems and applications.

The study from Sinan Cabuk *et al.*,[23] discuss about IoT-driven fault detection and prevention system in ship engine cooling pumps, as presented by Çabuk, aligns with the integration of computer vision, motion control, and IoT in sorting robots. The study emphasizes the pivotal role of IoT technology in reducing maintenance costs for ships, proposing a system involving smart sensors, a microcontroller, and the Node-RED platform. By leveraging IoT sensors, the system collects and analyzes real-time thermal, vibration, and current data from the ship's main engine cooling pump motor. The seamless integration of these components enables efficient data processing, transfer to a web interface, and storage on MySQL. This application mirrors the potential of IoT in enhancing the functionality of sorting robots, where real-time data monitoring and analysis contribute to operational efficiency. The study underscores the relevance of IoT-based preventive systems in industrial settings, resonating with the overarching theme of integrating IoT technologies for optimized performance in diverse domains, including sorting robots.

Table 2.3 Literature review matrix focusing on IoT.

Study	Key Finding	Aim	Results
[21]	Development of an IoT-controlled robotic arm	To design and implement a robotic arm controlled via an IoT application	Comprehensive design and implementation process, intuitive IoT application, real-time communication, usability, and accessibility achieved
[22]	MQTT-based IoT cloud platform with Node-RED for IoT setup	To streamline IoT platform setup for developers using MQTT and Node-RED	User-friendly web application, MERN stack utilization, Docker for container management, effective support for IoT application development
[23]	IoT-driven fault detection and prevention in ship engine cooling pumps	To reduce maintenance costs for ships using IoT technology	Integration of smart sensors, microcontroller, and Node-RED, real-time data collection, efficient data

			processing, transfer to a web interface, and storage on MySQL

CHAPTER 3

METHODOLOGY

3.1 Introduction

In the following sections, we present a detailed account of the methodologies employed in the successful completion of this project. This report serves as a comprehensive guide to the systematic approaches, techniques, and procedures that were meticulously applied. It unveils the intricate process of project planning, execution, and monitoring, shedding light on the methodologies that underpinned each phase. By delving into this methodology report, readers will gain insights into the structured framework that steered the project towards its fruition. This document aims to demystify the complexities, offering a transparent view of the strategies that contributed to the project's efficiency and success.

3.2 Selecting and Evaluating Tools for a Sustainable Development

Selection and evaluation of tools for sustainable development is crucial, no exception for this project on the integration of computer vision, motion control, and IoT technologies for the development of a sorting robot system [14]. This involves assessing computer vision frameworks for efficiency and accuracy, considering energy-efficient and precise motion control mechanisms, and evaluating IoT communication protocols and platforms for effective data exchange. The selection process prioritizes tools that align with sustainability principles, considering factors such as energy consumption, recyclability, and environmental impact. By employing this methodology, the sorting robot system can achieve optimal performance while minimizing resource consumption and contributing to sustainable practices in robotics.

3.3 Methodology

The project begins with the development of a high-level system architecture outlining the integration of computer vision, motion control, and IoT components. This involves defining interfaces and interactions between subsystems and determining the hardware requirements. The computer vision component includes selecting and implementing techniques for pattern recognition and sorting, choosing appropriate cameras or sensors, and developing algorithms for object detection and classification. The motion control component involves integrating suitable mechanisms, such as robotic arms, and developing control algorithms for precise object manipulation based on computer vision input. The IoT integration focuses on selecting an IoT platform, designing communication protocols, and implementing real-time data transmission.

System integration and testing involve integrating all components and thoroughly testing the system's functionality and performance, including pattern recognition accuracy and motion control precision. Performance evaluation is conducted by defining metrics, performing experiments, and analyzing results. Optimization and refinement are performed based on the outcomes of the evaluation, aiming to enhance system efficiency and effectiveness. The simple workflow for this project is shown in Figure 3.1 below.

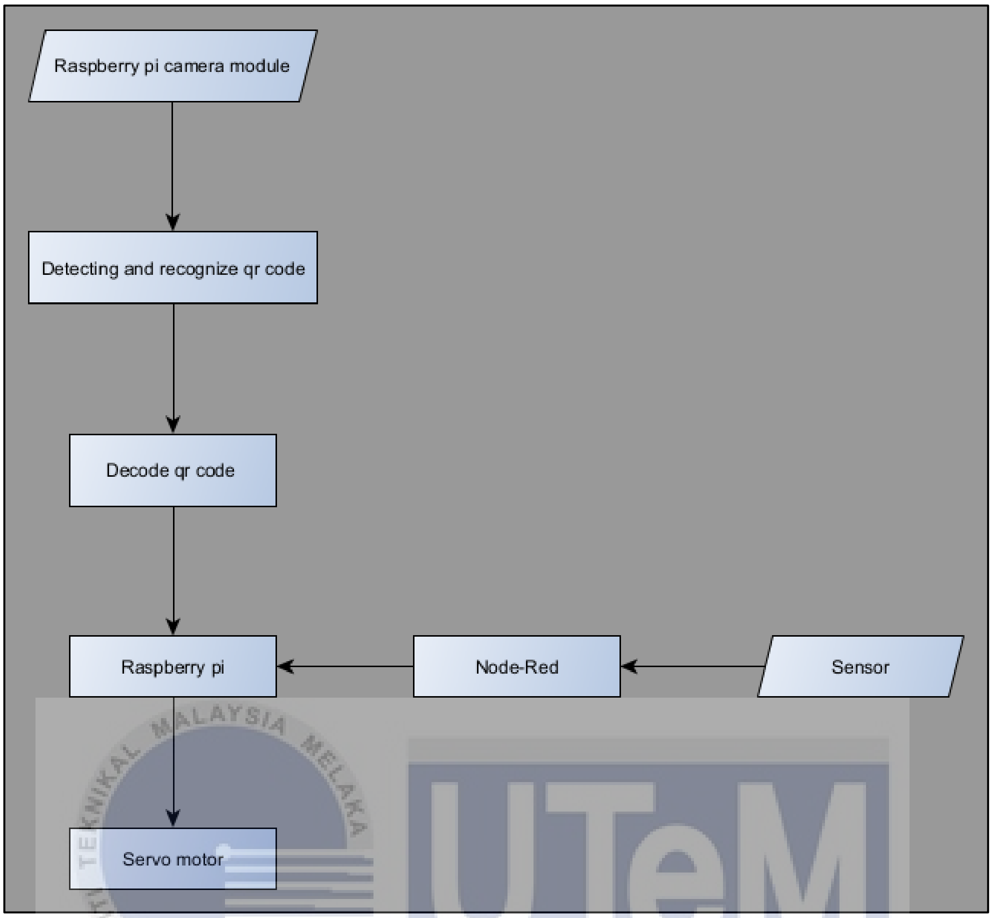

Figure 3.1 Project workflow.

3.4 Equipment

The equipment used and the operation is involved in the following steps. Firstly, a high-level system architecture is designed to integrate the Raspberry Pi, servo MG996R, 5MP Raspberry Pi Camera Module, and 6DOF robot arm. This architecture outlines the connections and interactions between these hardware components. Next, the computer vision component is developed using the Raspberry Pi and the 5MP Raspberry Pi Camera Module. Computer vision algorithms for pattern recognition and sorting are implemented on the Raspberry Pi, utilizing the captured visual data from the camera module. The motion control component incorporates the servo MG996R and the 6DOF robot arm, which are controlled by the Raspberry Pi. Control algorithms are developed to accurately manipulate, and position objects based on the information obtained from the computer vision component. The IoT integration aspect involves connecting the Raspberry Pi to an IoT platform, enabling remote monitoring of the sorting process. Communication protocols are implemented to exchange data between the Raspberry Pi and the IoT infrastructure. To determine the location of the sorted object, capacitive touch sensor is used. As items move through, the capacitive touch sensor swiftly identifies and contributes to the precise sorting of items.

3.4.1 Raspberry Pi

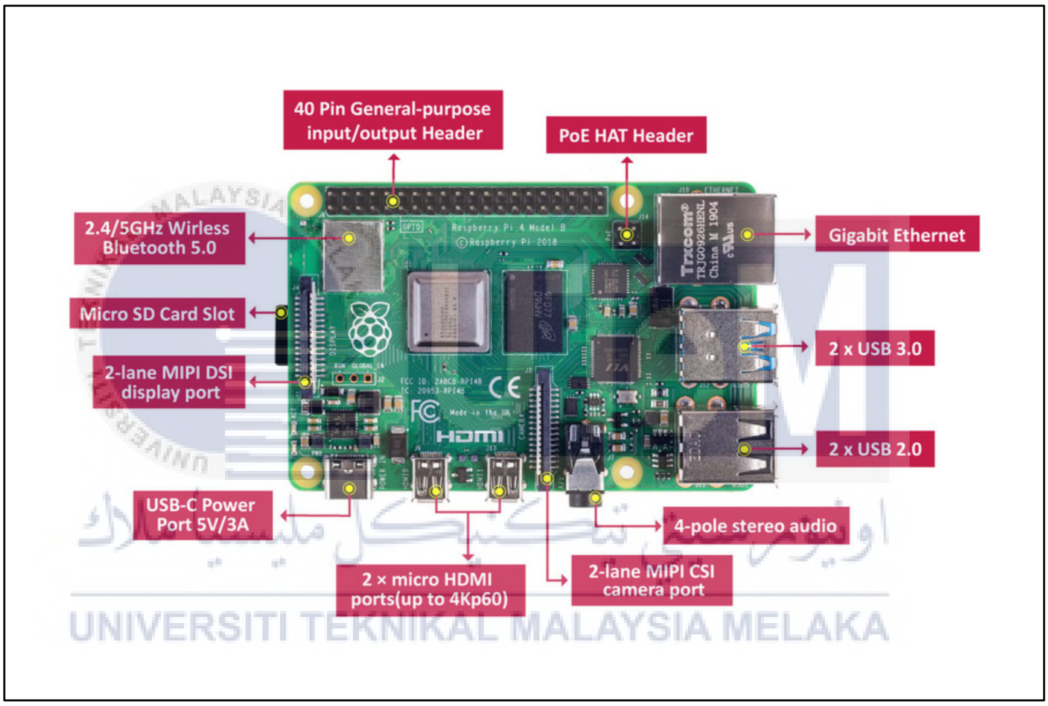

Figure 3.2 Raspberry Pi 4 model B.[24]

The project utilizes the Raspberry Pi 4 Model B with 4GB of memory. This compact and powerful single-board computer serves as the central processing unit, controlling the integration of computer vision, motion control, and IoT components for the sorting robots. Its ample memory capacity allows for efficient and smooth operation of the system.

3.4.2 Camera Module

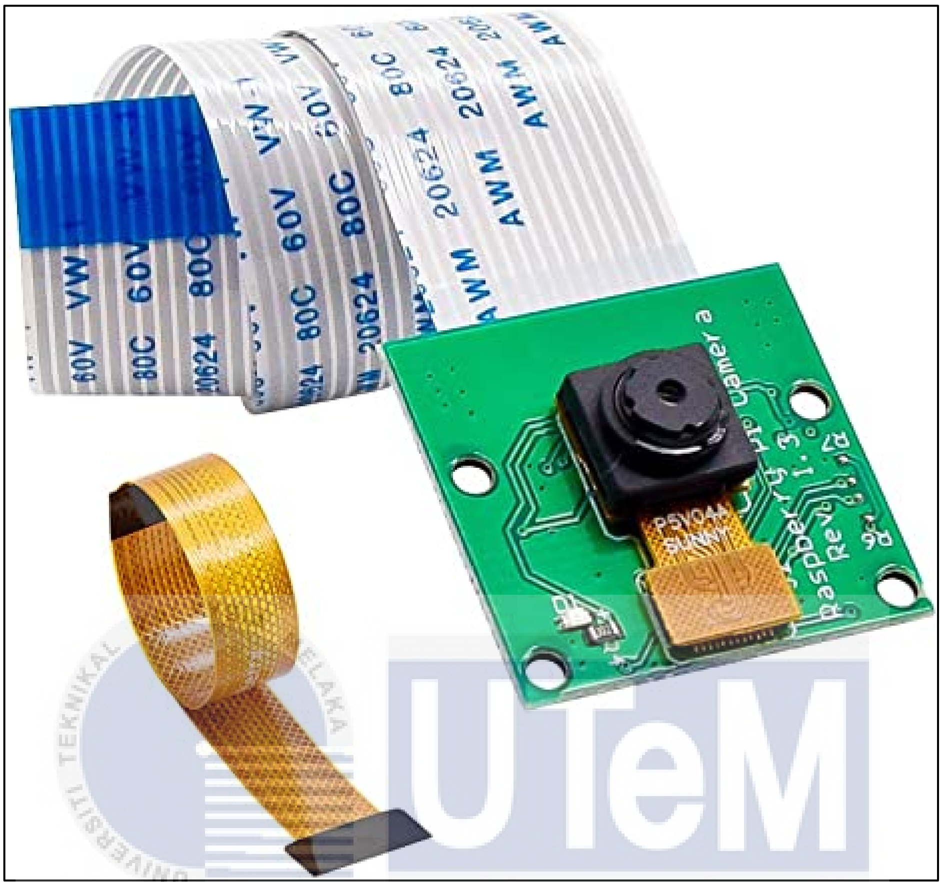

Figure 3.3 5MP 1080p Camera module with Pi Zero FFC Camera Cable.[25]

The 5MP 1080p Raspberry Pi Camera Module, provides high-resolution imaging capabilities for the computer vision component of the sorting robots. These components enable accurate pattern recognition and sorting based on visual data captured by the camera module.

3.4.3 Servo Motor

Figure 3.4 MG996R 180 Degree.[26]

The MG996R 180 Degree servo motor is utilized in the project for precise motion control of the sorting robots. With its high torque and 180-degree rotation capability, it enables accurate manipulation and positioning of objects during the sorting process.

3.4.4 Robot Arm

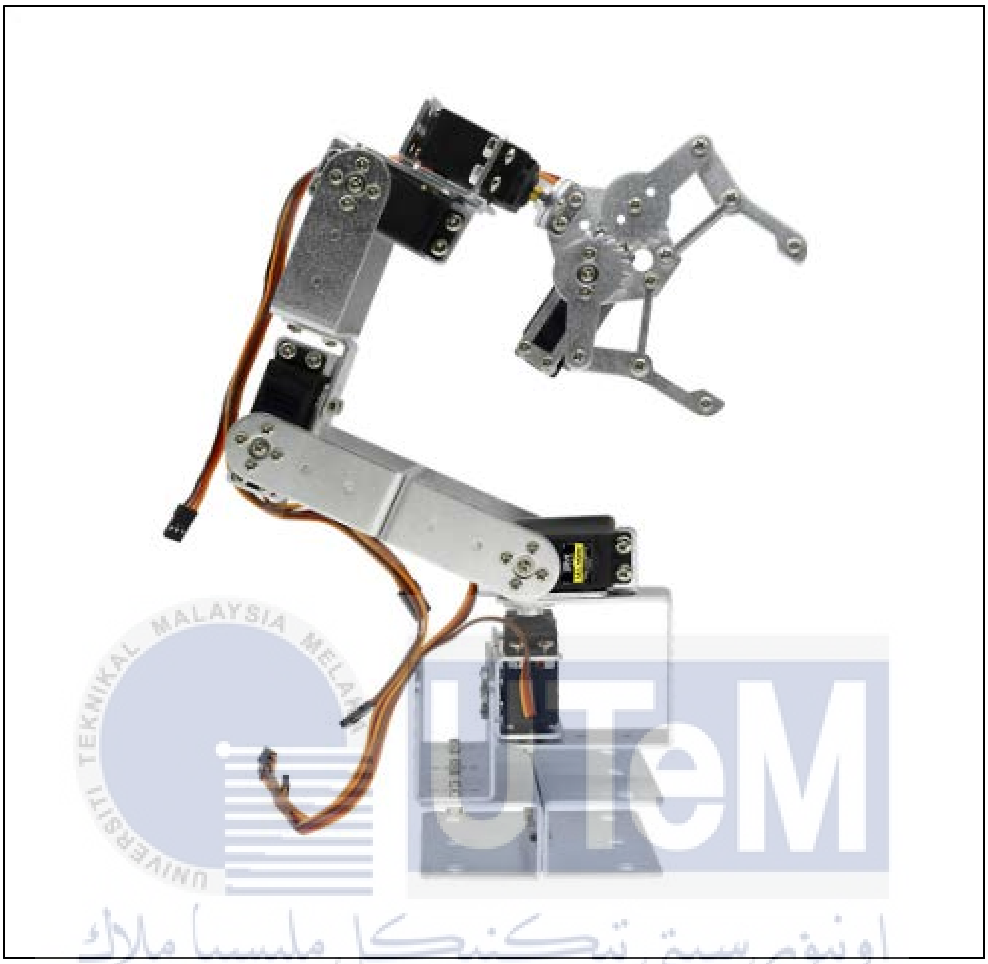

Figure 3.5 6DOF Robot arm.[27]

The 6DOF (Degree of Freedom) robot arm plays a crucial role in the project by providing versatile and precise manipulation capabilities. With its six independent axes of movement, it allows the sorting robots to handle objects with dexterity and perform complex sorting tasks effectively.

3.4.5 Capacitive Touch Sensor

Figure 3.6 Capacitive touch sensor.

A capacitive touch sensor detects touch or proximity by measuring changes in capacitance. When an item passes through a designated zone, disrupting the electric field, the sensor registers its presence.

3.5 Project Development

This section delves into the intricacies of the project's development, particularly focusing on the integration and operational nuances of key components.

3.5.1 Raspberry Pi Camera Module

The integration of the Raspberry Pi Camera Module plays a pivotal role in capturing visual data for our sorting robot project. This section serves as a comprehensive guide to enable and configure the Raspberry Pi Camera Module, ensuring seamless operation within the project's context.

3.5.1.1 Hardware Setup

To begin, ensure the Raspberry Pi board is powered off. Gently connect the ribbon cable from the Camera Module to the designated CSI port on the Raspberry Pi. Ensure a secure connection, taking care to align the connectors properly.

3.5.1.2 Enabling the Camera Interface

1. Power on the Raspberry Pi and log in to the terminal.
2. Run sudo raspi-config to access the configuration menu.
3. Navigate to "Interfacing Options" and select "Camera."
4. Choose "Yes" when prompted to enable the camera interface.
5. Reboot the Raspberry Pi for the changes to take effect.

3.5.1.3 Verifying Camera Functionality

To ensure the Camera Module is functioning correctly:

1. Open the terminal and execute raspistill -o image.jpg.
2. This command captures a still image and saves it as "image.jpg" in the home directory.
3. Confirm the successful capture of the image.

3.5.1.4 Python Integration

Integrating the Camera Module into Python scripts involves using the `*picamera*` library. Install it using the following commands on terminal:

sudo apt-get update

sudo apt-get install python3-picamera

3.5.2 QR Code Detection and Recognition using OpenCV

In this section, we delve into the pivotal aspect of QR code detection and recognition, a cornerstone in the functionality of our sorting robot. The integration of OpenCV serves as a powerful tool to accomplish this task efficiently.

3.5.2.1 OpenCV Installation

Begin by ensuring that OpenCV is installed on the Raspberry Pi. Execute the following commands in the terminal:

sudo apt-get update

sudo apt-get install python3-opencv

3.5.2.2 Configuring Camera module with OpenCV

Next, establish a seamless connection between the Raspberry Pi camera module and OpenCV. Utilize the following Python script as a template:


```python
import cv2
import numpy as np

# Initialize the camera
cap = cv2.VideoCapture(0)

while True:
    ret, frame = cap.read()

    # Apply necessary pre-processing steps (if any)

    # QR Code detection logic

    cv2.imshow('QR Code Detection', frame)

    # Break the loop on 'q' key press
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

# Release the camera and close all windows
```

```
cap.release()
cv2.destroyAllWindows()
```

This script initializes the camera, captures frames, and provides a foundation for implementing QR code detection.

3.5.3 QR Code Decoding using PyZbar

This section elucidates the crucial step of decoding QR codes, a fundamental operation for our sorting robot's precision. PyZbar, a Python library for reading barcodes and QR codes, is instrumental in this process.

3.5.3.1 PyZbar Installation

Install PyZbar on the system by execute the following command in the terminal:

pip install pyzbar

3.5.3.2 Implementing QR Code Decoding with PyZbar

Begin by importing the requisite modules and initializing the camera, similarly to the previous steps. Extend the Python script to include PyZbar-based QR code decoding:

```
import cv2
from pyzbar.pyzbar import decode

# Initialize the camera
cap = cv2.VideoCapture(0)

while True:
    ret, frame = cap.read()

    # QR Code decoding logic
    decoded_objects = decode(frame)

    for obj in decoded_objects:
        # Extract and process the decoded information
        print("Decoded Data:", obj.data.decode('utf-8'))
```

```
        # Optionally, perform additional actions based on the decoded data

    cv2.imshow('QR Code Decoding', frame)

    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()
```

This script leverages the PyZbar library to detect and decode QR codes from the captured frames. The decoded information is extracted, and subsequent actions can be performed based on the deciphered data.

3.5.4 Node-RED Integration

This section provides a comprehensive manual for integrating Node-RED into the project development, a critical aspect of enabling seamless communication and control within the sorting robot system.

3.5.4.1 Node-RED Installation

Ensure Node.js is installed on the system. If not, it can be download and install from https://nodejs.org/. Once Node.js is installed, follow these steps:

1. Open the terminal and execute the following command to install Node-RED globally:

npm install -g --unsafe-perm node-red

2. Start Node-RED by execute the following command:

node-red

3. Access Node-RED's visual editor by opening a web browser and navigating to `*http://localhost:1880*`.

3.6 Wiring Configuration for Servo and Sensor

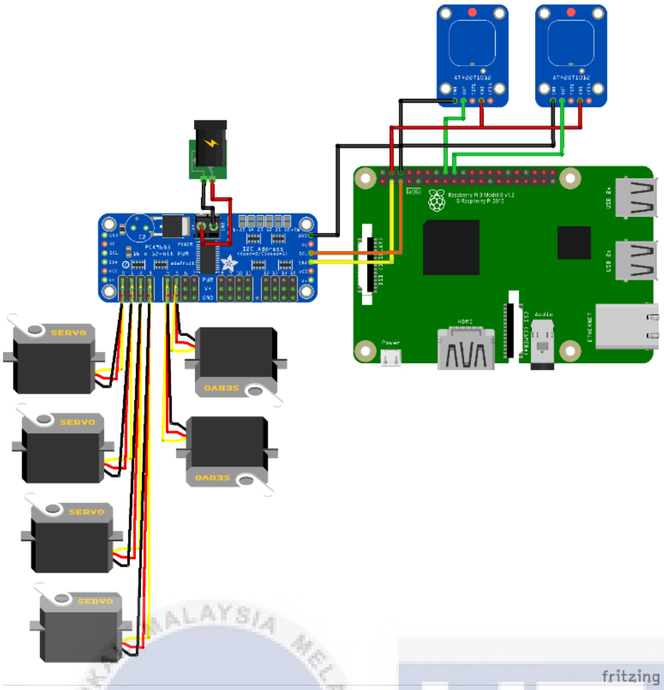

Figure 3.7 Connection for Servo and Sensor.

The servo is connected to a PCA9685 servo motor driver. To drive all six servos simultaneously, it requires an external power supply. Connect the servo motor driver to the 5V and ground on the Raspberry Pi GPIO pins for power. Communication between the motor driver and Raspberry Pi is achieved through I2C, where both the SDA and SCL pins must be connected. For the capacitive touch sensor, connect it to both 5V and ground. The trigger pin should be connected to pins 16 and 18.

3.7 Robot Arm Operation Workflow

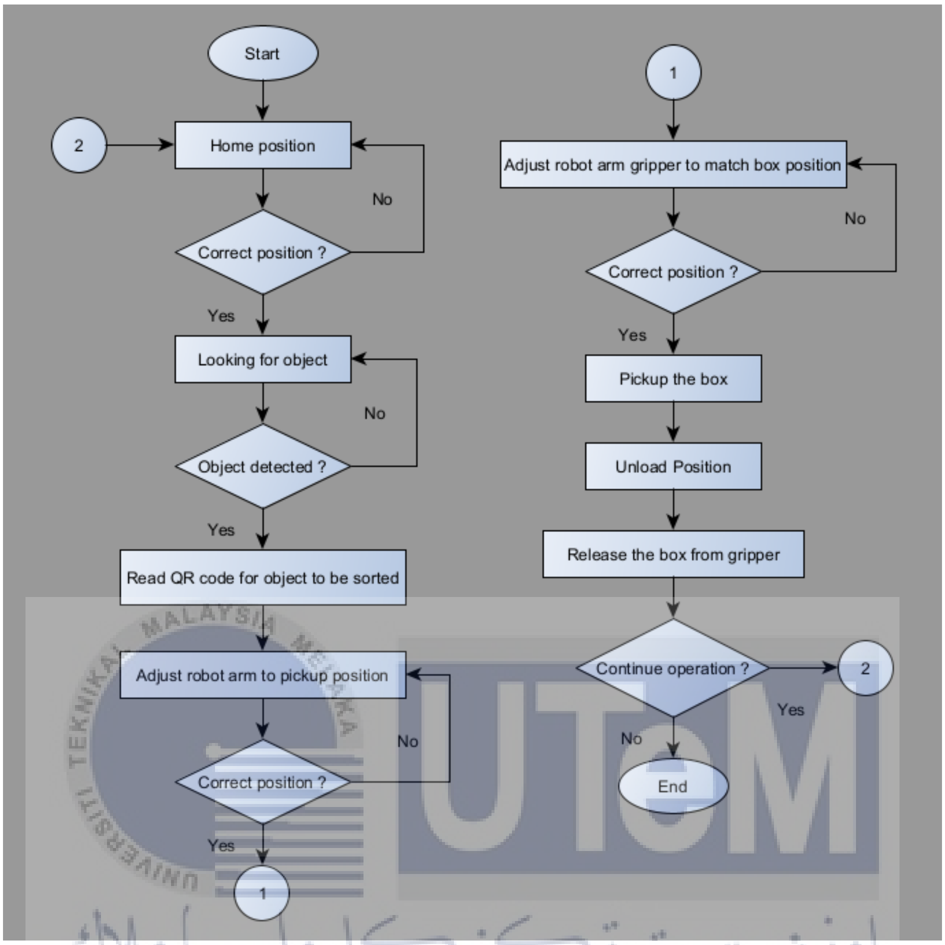

Figure 3.8 Robot Arm Workflow.

3.8 Limitation Of Proposed Methodology

The proposed methodology for the sorting robot system presents several limitations. Environmental sensitivity may impact the robot's performance under varying conditions, potentially affecting its robustness. The accuracy of QR code detection and decoding using OpenCV may be influenced by code complexity and environmental factors, such as lighting. Processing speed could be a concern, particularly during high-throughput scenarios. Dependency on internet connectivity for Node-RED integration poses a risk to real-time monitoring and control in the absence of a stable connection. Precise calibration of servos and sensors is crucial for accurate robot movements, and any misalignment could compromise sorting accuracy. Cost constraints associated with specific hardware components might limit scalability. The manual-style approach to using OpenCV, Node-RED, and hardware configuration may pose a learning curve for users. Security concerns regarding IoT functionalities and potential maintenance challenges are also significant considerations. Addressing these limitations through iterative testing and refinement is crucial for enhancing the overall effectiveness and practicality of the proposed methodology.

3.9 Summary

This chapter focuses on the development of the sorting robot, encompassing vital aspects like Raspberry Pi camera integration, QR code detection using OpenCV, and wiring configurations for servo motors and sensors. It serves as a practical guide, detailing the setup of each component, providing clear instructions on the integration of hardware and software. This chapter lays the groundwork for the efficient functioning of the sorting robot, guaranteeing the smooth integration of technologies that are essential to its functioning.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Final Design

Figure 4.1 Sorting Robot.

The final hardware design of the sorting robot is a meticulous integration of key components. The Raspberry Pi camera module acts as the robot's eyes, capturing images of objects for processing. OpenCV, employed for QR code detection, enhances the robot's perceptual abilities. In the wiring configuration, servos and sensors are strategically connected, optimizing motion control and object recognition. This hardware design ensures a cohesive system, empowering the robot to efficiently navigate and sort objects. The meticulous arrangement and connection of these components form the backbone of the sorting robot's physical structure, contributing to its overall effectiveness in industrial applications.

4.2 Result

This section delves into the outcomes of key components within the integrated system. From the sorting mechanism to the OpenCV in QR code recognition, each facet contributes to the overall success of the project. The seamless integration of Node-RED establishes a foundation for real-time monitoring and control. The collective performance of these elements underlines the achievement of project objectives, showcasing the prowess of computer vision, motion control, and IoT technologies in optimizing sorting processes.

4.2.1 Sorting Mechanism

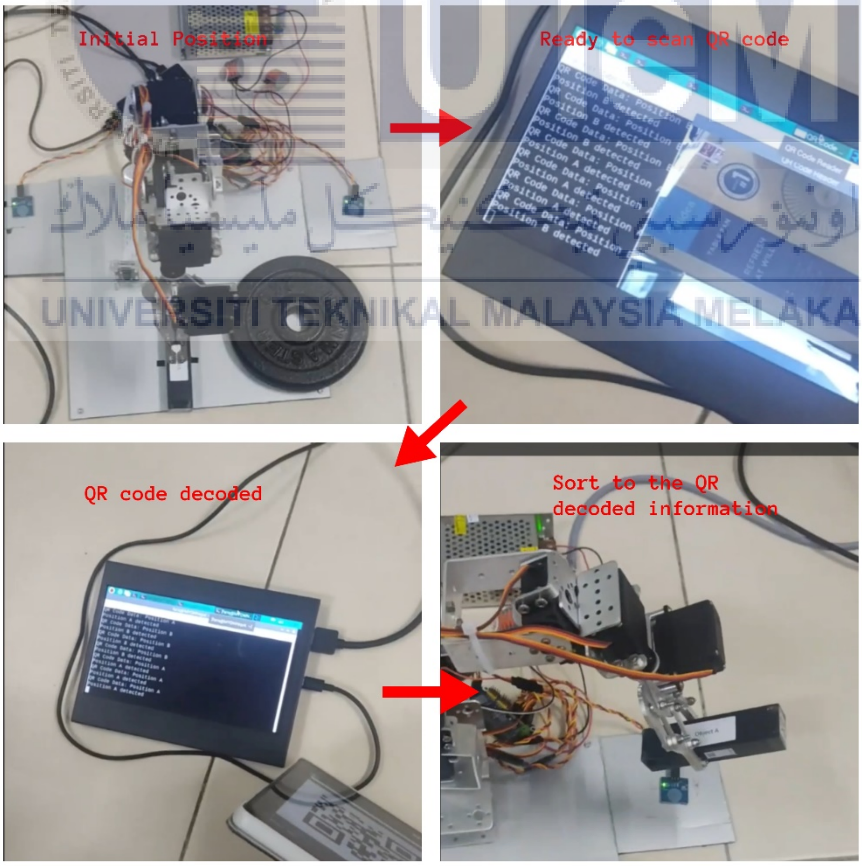

Figure 4.2 Sorted object.

From the image above, is the mechanism where the robot arm will be ready on initial condition. At the same time, the camera is ready to scan any QR code. If the correct QR code is detected, it will start to sort based on the QR code data decode.

4.2.2 Software mechanisms

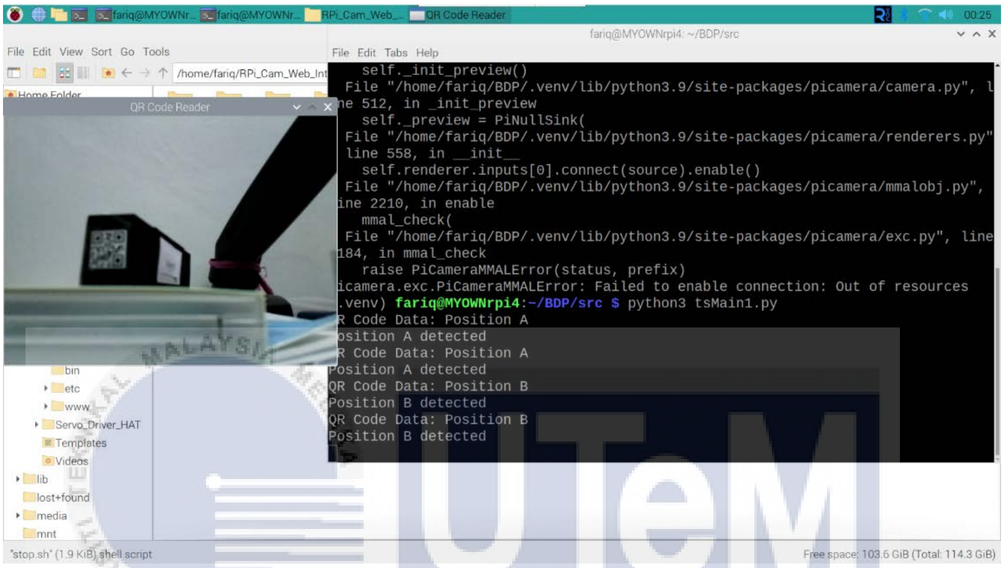

Figure 4.3 Project script.

This image displays the camera view from the OpenCV window GUI, prepared to scan a QR code. PyZbar will decode the information and present it on the terminal.

4.2.3 IoT dashboard

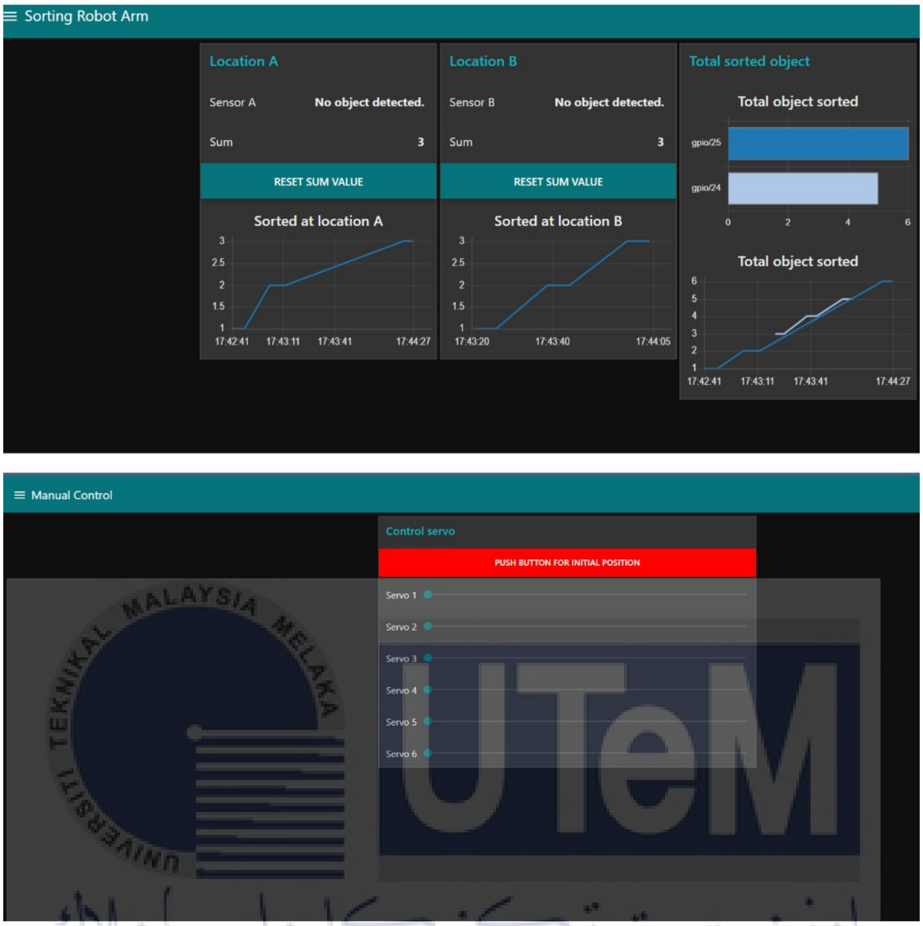

Figure 4.4 Node-RED dashboard.

The top image shows the automatic sorting section on the Node-RED dashboard. In the automatic sorting section, there are three parts: Location A, Location B, and Total Sorted Objects. Location A and Location B represent the output from the capacitive sensor, which operates based on the principles of capacitance and detects changes in capacitance caused by the proximity or touch of an object. This means that if the sensor detects an object on either side, it will increment the count on the Node-RED dashboard. If the reset sum value is pushed, it will reset the sum value. Additionally, there is a graph to display the number of sorted objects per hour. The Total Sorted Objects part, it adds the sum of both sorted objects on Location A and Location B. The image at the bottom shows the Manual Control section, which has only one part. It includes six sliders to control the PWM of the servo motor and a button to return the robot arm to its initial position.

4.3 Issue Encountered

The issue I encountered is that the camera's megapixel count is too low to capture a small and detailed image. The image becomes blurry to the point where it can't decode the QR code. I noticed this issue when scanning a larger QR code, it can be read without any problem because OpenCV can recognize the pattern of the QR code. Due to the size limitation of the gripper on the robot arm, I am restricted in the objects I can use. Therefore, with small objects, I can only attach a small QR code. To address this, a better camera with higher megapixels is needed. However, for this project, the budget has already exceeded its limit. As a solution, I used the image of the QR code from my phone, where the image is brighter and much larger, making it easier to recognize with the 5-megapixel camera module.

4.4 Analysis

This analysis is extracted from the Node-RED dashboard and can be observed in the image below. In Figure 4.5, I am attempting to measure the sorting time frequency for sorting 5 objects to Location A and other 5 objects to Location B. From the graph plotted in Figure 4.5, we can observe that the sorting time remains consistent for each instance. We can use the formula 'Time to sort one item = (Total sorting time) / (Number of items).' From the graph plotted for Location A, the time taken to sort 5 objects is approximately 54 seconds. Dividing this time by the total objects sorted (5), the time to sort one item is calculated as 10.8 seconds. In Location B, the 5th sorted object stays a bit longer, as I was slightly delayed in picking up the object from the sensor. However, for the past 4 items sorted, the time remains the same.

Figure 4.5 Sorting time frequency.

In Figure 4.6, I am attempting to measure the sorting robot's movement from the initial position to Location A and then back to the initial position and onward to Location B. From the graph, it is evident that it takes 109 seconds to complete the sorting of 10 objects between these two locations. Applying the same formula, we can observe that it takes approximately 10.9 seconds to sort one object, which is nearly the same for both the 5 objects to Location A and the 5 objects to Location B.

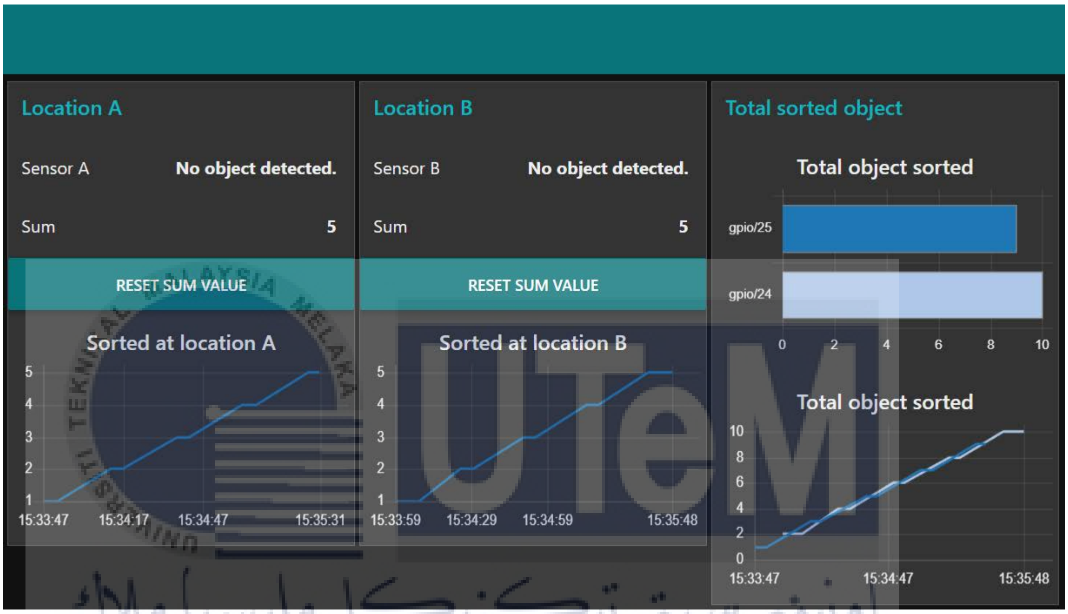

Figure 4.6 Sorting time frequency (Initial to A, Initial to B).

CHAPTER 5

CONCLUSION AND RECOMMENDATION

5.1 Conclusion

In conclusion, the project focused on the integration of computer vision and motion control for sorting robots with IoT integration. The development and implementation of this integrated system have demonstrated significant advancements in automation and efficiency across various industries. By combining computer vision technology, which enables robots to perceive and interpret visual information, with precise motion control mechanisms, the sorting robots are capable of accurately identifying and sorting objects with minimal human intervention. The integration of IoT further enhances the capabilities of these robots by enabling seamless connectivity and data exchange with other devices and systems. This connectivity allows for real-time monitoring, remote control, and data analysis, leading to improved decision-making and optimization of sorting processes. The use of IoT also facilitates the collection of valuable data that can be utilized for predictive maintenance, performance analysis, and process optimization. The successful implementation of this integrated system offers numerous benefits, including increased productivity, reduced errors, enhanced safety, and cost savings. Furthermore, it sets the foundation for future advancements in robotics and automation, paving the way for more sophisticated and intelligent sorting systems. Overall, the integration of computer vision, motion control, and IoT in sorting robots represents a significant technological breakthrough that revolutionizes the sorting industry, empowering businesses to achieve higher levels of efficiency, accuracy, and competitiveness.

5.2 Potential for Commercialization

The integration of computer vision and motion control for sorting robots with IoT integration presents substantial prospects for commercialization. The increasing demand for advanced robotic systems in logistics, manufacturing, and supply chain management underscores the commercial viability of this technology. In logistics, automated sorting enhances operational efficiency, reducing errors and streamlining processes. Moreover, the application of IoT facilitates real-time monitoring and data-driven decision-making, providing valuable insights into sorting operations. The potential commercial avenues extend to industries requiring precision automation, such as e-commerce, where efficient sorting directly impacts customer satisfaction and operational costs. To leverage this potential, strategic partnerships, licensing agreements, and customization services can be explored. However, it is crucial to ensure ethical and sustainable practices in the commercialization process, promoting responsible use of technology and contributing to both economic growth and environmental well-being. Balancing innovation with ethical considerations will be essential for the successful and socially responsible commercialization of the integrated system.

5.3 Future Works

For future enhancements, refining the precision of the Integration of Computer Vision and Motion Control for Sorting Robots with IoT Integration can be achieved through the implementation of a potentiometer that mirrors the movement of each robot arm joint. This can be accomplished by integrating parallel axis gears at each joint, connecting them to the respective potentiometer. Through this approach, the microcontroller gains accurate data of the robot arm's positional coordinates, enhancing overall movement accuracy. The use of robot kinematics also can improve the motion control of the robot arm.

REFERENCES

[1] M. Javaid, A. Haleem, R. P. Singh, S. Rab, and R. Suman, “Exploring impact and features of machine vision for progressive industry 4.0 culture,” 2021, doi: 10.1016/j.sintl.2021.100132.

[2] W. Zhang, J. Mei, and Y. Ding, “Design and Development of a High Speed Sorting System Based on Machine Vision Guiding,” *Phys Procedia*, vol. 25, pp. 1955–1965, Jan. 2012, doi: 10.1016/J.PHPRO.2012.03.335.

[3] M. Abdullah-Al-Noman, A. N. Eva, T. B. Yeahyea, and R. Khan, “Computer Vision-based Robotic Arm for Object Color, Shape, and Size Detection,” *Journal of Robotics and Control (JRC)*, vol. 3, no. 2, 2022, doi: 10.18196/jrc.v3i2.13906.

[4] Institute of Electrical and Electronics Engineers, *ICCAR 2018 : 2018 4th International Conference on Control, Automation and Robotics : proceedings : April 20-23, 2018, Auckland, New Zealand.*

[5] B. Xuan Duong *et al.*, “Deep Learning based 6DOF Robot Arm Control with Computer Vision.” [Online]. Available: www.ijert.org

[6] A. S. Shaikat, S. Akter, and U. Salma, “Computer Vision Based Industrial Robotic Arm for Sorting Objects by Color and Height,” *Journal of Engineering Advancements*, vol. 01, no. 04, 2020, doi: 10.38032/jea.2020.04.002.

[7] V. D. Cong, L. D. Hanh, L. H. Phuong, and D. A. Duy, “Design and Development of Robot Arm System for Classification and Sorting Using Machine Vision,” *FME Transactions*, vol. 50, no. 1, pp. 181–192, 2022, doi: 10.5937/fme2201181C.

[8] H. Meddeb, Z. Abdellaoui, and F. Houaidi, “Development of surveillance robot based on face recognition using Raspberry-PI and IOT,” *Microprocess Microsyst*, vol. 96, Feb. 2023, doi: 10.1016/j.micpro.2022.104728.

[9] M. E. El-sayed, A. W. Youssef, O. M. Shehata, L. A. Shihata, and E. Azab, “Computer vision for package tracking on omnidirectional wheeled conveyor: Case study,” *Eng Appl Artif Intell*, vol. 116, Nov. 2022, doi: 10.1016/j.engappai.2022.105438.

[10] N. Boysen, S. Schwerdfeger, and M. W. Ulmer, “Robotized sorting systems: Large-scale scheduling under real-time conditions with limited lookahead,” *Eur J Oper Res*, vol. 310, no. 2, pp. 582–596, Oct. 2023, doi: 10.1016/j.ejor.2023.03.037.

[11] W. Lu and J. Chen, “Computer vision for solid waste sorting: A critical review of academic research,” *Waste Management*, vol. 142. Elsevier Ltd, pp. 29–43, Apr. 01, 2022. doi: 10.1016/j.wasman.2022.02.009.

[12] W. Zhang, J. Mei, and Y. Ding, “Design and Development of a High Speed Sorting System Based on Machine Vision Guiding,” *Phys Procedia*, vol. 25, pp. 1955–1965, 2012, doi: 10.1016/j.phpro.2012.03.335.

[13] “Computer vision apply for medical image processing | Download Scientific Diagram.” Accessed: Jun. 10, 2023. [Online]. Available: https://www.researchgate.net/figure/Computer-vision-apply-for-medical-image-processing_fig1_318288922

[14] I. Dincer, “Renewable energy and sustainable development: a crucial review”, Accessed: Jun. 10, 2023. [Online]. Available: www.elsevier.com/locate/rser

[15] K. Kjeldsen, M. Nodeland, A. Fagerstrøm, and S. Pawar, “The relative impact of QR codes on omnichannel customer experience and purchase intention,” in *Procedia Computer Science*, Elsevier B.V., 2023, pp. 1049–1056. doi: 10.1016/j.procs.2023.01.383.

[16] S. Asif and P. Webb, "Kinematics Analysis of 6-DoF Articulated Robot with Spherical Wrist," *Math Probl Eng*, vol. 2021, 2021, doi: 10.1155/2021/6647035.
[17] A. T, S. M, and V. B, "Kinematic Analysis of 6 DOF Articulated Robotic Arm," *International Research Journal of Multidisciplinary Technovation*, pp. 1–5, Jan. 2021, doi: 10.34256/irjmt2111.
[18] "The 5 Most Important Considerations for Motion Applications » Electronicsmedia." Accessed: Jun. 10, 2023. [Online]. Available: https://www.electronicsmedia.info/2017/11/06/5-important-considerations-motion-applications/
[19] M. Taghizadeh, A. Ghaffari, and F. Najafi, "Improving dynamic performances of PWM-driven servo-pneumatic systems via a novel pneumatic circuit," *ISA Trans*, vol. 48, no. 4, pp. 512–518, 2009, doi: 10.1016/j.isatra.2009.05.001.
[20] "Block Diagram of Internet of Things (IoT) The Figure 1 shows the block... | Download Scientific Diagram." Accessed: Jun. 10, 2023. [Online]. Available: https://www.researchgate.net/figure/Block-Diagram-of-Internet-of-Things-IoT-The-Figure-1-shows-the-block-diagram-of_fig1_337209480
[21] H. Kareemullah, D. Najumnissa, M. S. M. Shajahan, M. Abhineshjayram, V. Mohan, and S. A. Sheerin, "Robotic Arm controlled using IoT application," *Computers and Electrical Engineering*, vol. 105, Jan. 2023, doi: 10.1016/j.compeleceng.2022.108539.
[22] C. Rattanapoka, S. Chanthakit, A. Chimchai, and A. Sookkeaw, "An MQTT-based IoT Cloud Platform with Flow Design by Node-RED," *RI2C 2019 - 2019 Research, Invention, and Innovation Congress*, Dec. 2019, doi: 10.1109/RI2C48728.2019.8999942.
[23] A. Sinan Cabuk, "Experimental IoT study on fault detection and preventive apparatus using Node-RED ship's main engine cooling water pump motor," *Eng Fail Anal*, vol. 138, Aug. 2022, doi: 10.1016/j.engfailanal.2022.106310.
[24] "Raspberry Pi 4 Model B 4Gb." Accessed: Jun. 13, 2023. [Online]. Available: https://www.robot-advance.com/EN/art-raspberry-pi-4-model-b-4go-2640.htm
[25] "Amazon.com: KEYESTUDIO for Raspberry Pi Camera, 5MP 1080p Camera Module with OV5647 Sensor Video Webcam for Raspberry Pi Model A/B/B+, Pi 2, Raspberry Pi 3, 3B+, Pi 4B Ras Pi 4 Zero W Camera with Flex Cable : Electronics." Accessed: Jun. 13, 2023. [Online]. Available: https://www.amazon.com/Keyestudio-Camera-Module-Raspberry-Model/dp/B073RCXGQS
[26] "DIYMORE MG996R Torque Digital Metal Gear Servo Fit for Helicopter Car Boat Model N4 | Shopee Malaysia." Accessed: Jun. 13, 2023. [Online]. Available: https://shopee.com.my/DIYMORE-MG996R-Torque-Digital-Metal-Gear-Servo-Fit-for-Helicopter-Car-Boat-Model-N4-i.145270449.8306260633?sp_atk=f9a6766c-024b-41f1-99b6-41d9d715cdb3&xptdk=f9a6766c-024b-41f1-99b6-41d9d715cdb3
[27] "DIYMORE ROT3U 6DOF Aluminium Robot Arm Mechanical Robotic Clamp Claw for Arduino DIY | Shopee Malaysia." Accessed: Jun. 13, 2023. [Online]. Available: https://shopee.com.my/DIYMORE-ROT3U-6DOF-Aluminium-Robot-Arm-Mechanical-Robotic-Clamp-Claw-for-Arduino-DIY-i.145270449.7166918402?sp_atk=d7dc1335-90a4-48fc-a3c4-0afb0a514f39&xptdk=d7dc1335-90a4-48fc-a3c4-0afb0a514f39

APPENDICES

Appendix A Coding for main.py

```python
import cv2
from pyzbar.pyzbar import decode
from picamera.array import PiRGBArray
from picamera import PiCamera
from qr_utils import decode_qr_code,overlay_text
from pwmLocationV2 import sort2A,sort2B,homePos

def read_qr_code():
    camera = PiCamera()
    camera.resolution = (640, 480)
    rawCapture = PiRGBArray(camera)

    # Start a preview using only OpenCV GUI window
    for frame in camera.capture_continuous(rawCapture, format="bgr", use_video_port=True):
        image = frame.array

        qr_data = decode_qr_code(image)

        if qr_data is not None:
            overlay_text(image,qr_data)
            break  # Stop the loop if a QR code is found

        # Display the frame in OpenCV GUI window
        cv2.imshow("QR Code Reader", image)

        key = cv2.waitKey(1) & 0xFF  # Check for 'q' key to quit
        if key == ord("q"):
            break

        rawCapture.truncate(0)  # Clear the stream for the next frame

    # Clean up
    camera.close()
    cv2.destroyAllWindows()

    return qr_data

if __name__ == "__main__":
    while True:
        qr_data = read_qr_code()
        if qr_data == "Position A":
            print("Position A detected")
            sort2A()
            homePos()
        elif qr_data == "Position B":
            print("Position B detected")
            sort2B()
            homePos()
        else:
            pass
```

Appendix B Coding for qr_utils.py

```python
import cv2
from pyzbar.pyzbar import decode
from picamera.array import PiRGBArray
from picamera import PiCamera

def decode_qr_code(frame):
    # Convert the frame to grayscale
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    # Use the pyzbar library to decode QR codes
    qr_codes = decode(gray)

    # Loop over the detected QR codes
    for qr_code in qr_codes:
        qr_data = qr_code.data.decode("utf-8")
        print("QR Code Data:", qr_data)
        return qr_data

    return None

def overlay_text(frame, text, position=(10, 30), font_scale=1, font_thickness=2, font_color=(0, 255, 0)):
    font = cv2.FONT_HERSHEY_SIMPLEX
    cv2.putText(frame, text, position, font, font_scale, font_color, font_thickness, cv2.LINE_AA)
```

Appendix C   Coding for pwmLocationV2.py (Home position)

```
def homePos():
    pwm. setServoPulse (0,1485)
    pwm. setServoPulse (1,1606)
    pwm. setServoPulse (2,1266)
    #time.sleep(1)
    pwm. setServoPulse (3,1130)
    pwm. setServoPulse (4,1538)
    pwm. setServoPulse (5,1096)
    time.sleep(1)
```

Appendix D Coding for pwmLocationV2.py (Sort to Location A)

```python
def sort2A():
    pwm. setServoPulse (0,1485)
    pwm. setServoPulse (1,1606)
    pwm. setServoPulse (2,1266)
    #time.sleep(1)
    pwm. setServoPulse (3,1130)
    pwm. setServoPulse (4,1538)
    pwm. setServoPulse (5,500)
    time.sleep(1)

    pwm. setServoPulse (0,1485)
    i=1606
    for x in range(i,1210,-15):
        pwm. setServoPulse (1,x)
        time.sleep(0.02)
    pwm. setServoPulse (2,1266)
    #time.sleep(1)
    pwm. setServoPulse (3,1130)
    pwm. setServoPulse (4,1538)
    pwm. setServoPulse (5,500)
    time.sleep(1)

    pwm. setServoPulse (0,1485)
    pwm. setServoPulse (1,1210)
    pwm. setServoPulse (2,1266)
    #time.sleep(1)
    pwm. setServoPulse (3,1130)
    pwm. setServoPulse (4,1538)
    pwm. setServoPulse (5,900)
    time.sleep(1)

    pwm. setServoPulse (0,1485)
    pwm. setServoPulse (1,1606)
    pwm. setServoPulse (2,1266)
    #time.sleep(1)
    pwm. setServoPulse (3,1130)
    pwm. setServoPulse (4,1538)
    pwm. setServoPulse (5,900)
    time.sleep(1)

    i=2435
    for x in range(1485,i,25):
        pwm. setServoPulse (0,x)
        time.sleep(0.02)
    pwm. setServoPulse (1,1606)
    pwm. setServoPulse (2,1266)
    #time.sleep(1)
    pwm. setServoPulse (3,1130)
    pwm. setServoPulse (4,1538)
    pwm. setServoPulse (5,900)
    time.sleep(1)

    pwm. setServoPulse (0,2435)
    i=1606
    for x in range(i,1210,-15):
        pwm. setServoPulse (1,x)
        time.sleep(0.02)
    pwm. setServoPulse (1,1210)
    pwm. setServoPulse (2,1266)
    #time.sleep(1)
    pwm. setServoPulse (3,1130)
    pwm. setServoPulse (4,1538)
    pwm. setServoPulse (5,900)
    time.sleep(1)

    pwm. setServoPulse (0,2435)
    pwm. setServoPulse (1,1210)
    pwm. setServoPulse (2,1266)
    #time.sleep(1)
    pwm. setServoPulse (3,1130)
    pwm. setServoPulse (4,1538)
    i=900
    for x in range(i,500,-15):
        pwm. setServoPulse (5,x)
        time.sleep(0.02)
    time.sleep(1)

    pwm. setServoPulse (0,2435)
    pwm. setServoPulse (1,1606)
    pwm. setServoPulse (2,1266)
    #time.sleep(1)
    pwm. setServoPulse (3,1130)
    pwm. setServoPulse (4,1538)
    pwm. setServoPulse (5,500)
    time.sleep(1)
```

Appendix E Coding for pwmLocationV2.py (Sort to Location B)

```python
def sort2B():

    pwm. setServoPulse (0,1485)
    pwm. setServoPulse (1,1606)
    pwm. setServoPulse (2,1266)
    #time.sleep(1)
    pwm. setServoPulse (3,1130)
    pwm. setServoPulse (4,1538)
    pwm. setServoPulse (5,500)
    time.sleep(1)

    pwm. setServoPulse (0,1485)
    i=1606
    for x in range(i,1210,-15):
        pwm. setServoPulse (1,x)
        time.sleep(0.02)
    pwm. setServoPulse (2,1266)
    #time.sleep(1)
    pwm. setServoPulse (3,1130)
    pwm. setServoPulse (4,1538)
    pwm. setServoPulse (5,500)
    time.sleep(1)

    pwm. setServoPulse (0,1485)
    pwm. setServoPulse (1,1210)
    pwm. setServoPulse (2,1266)
    #time.sleep(1)
    pwm. setServoPulse (3,1130)
    pwm. setServoPulse (4,1538)
    pwm. setServoPulse (5,900)
    time.sleep(1)

    pwm. setServoPulse (0,1485)
    pwm. setServoPulse (1,1606)
    pwm. setServoPulse (2,1266)
    #time.sleep(1)
    pwm. setServoPulse (3,1130)
    pwm. setServoPulse (4,1538)
    pwm. setServoPulse (5,900)
    time.sleep(1)

    i=565
    for x in range(1485,i,-25):
        pwm. setServoPulse (0,x)
        time.sleep(0.02)
    pwm. setServoPulse (1,1606)
    pwm. setServoPulse (2,1266)
    #time.sleep(1)
    pwm. setServoPulse (3,1130)
    pwm. setServoPulse (4,1538)
    pwm. setServoPulse (5,900)
    time.sleep(1)

    pwm. setServoPulse (0,565)
    i=1606
    for x in range(i,1210,-15):
        pwm. setServoPulse (1,x)
        time.sleep(0.02)
    pwm. setServoPulse (1,1210)
    pwm. setServoPulse (2,1266)
    #time.sleep(1)
    pwm. setServoPulse (3,1130)
    pwm. setServoPulse (4,1538)
    pwm. setServoPulse (5,900)
    time.sleep(1)

    pwm. setServoPulse (0,565)
    pwm. setServoPulse (1,1210)
    pwm. setServoPulse (2,1266)
    #time.sleep(1)
    pwm. setServoPulse (3,1130)
    pwm. setServoPulse (4,1538)
    i=900
    for x in range(i,500,-15):
        pwm. setServoPulse (5,x)
        time.sleep(0.02)
    time.sleep(1)

    pwm. setServoPulse (0,565)
    pwm. setServoPulse (1,1606)
    pwm. setServoPulse (2,1266)
    #time.sleep(1)
    pwm. setServoPulse (3,1130)
    pwm. setServoPulse (4,1538)
    pwm. setServoPulse (5,500)
    time.sleep(1)
```