Ain Shams University
Faculty of Computer & Information Sciences
Computer System Department

Mobile Robotic Arm

By

Morad Omar Sayed [Computer System]

Omar Ahmed Abdelmohaimen [Computer System]

Israa Hassan Mohamed [Computer System]

Shery Raafat Fekry [Computer System]

Omar Osama Mohamed [Computer System]

Ali Yasser Mohamed [Computer System]

Under Supervision of

Prof . Mahmoud Fayez
Computer System Department,
Faculty of Computer and Information Sciences,
Ain Shams University.

T.A. Ahmed Darwieesh
Computer System Department,
Faculty of Computer and Information Sciences,
Ain Shams University.

June 2024

Table of Contents

Acknowledgements

All praise and thanks to ALLAH, who provided us with the ability to complete this work. We hope to accept this work from us.

We are grateful to our *parents* and *our families* who are always providing help and support throughout the whole years of study. We hope we can give that back to them.

We also offer my sincerest gratitude to our supervisors, *Prof. Dr. Eman Shabaan, prof. Mahmoud Fayez and T.A Ahmed Darwieesh* who have supported us throughout our thesis with their patience, knowledge and experience.

Finally, We would like to thank our friends and all the people who gave us support and encouragement.

Abstract

In the medical field, particularly in drug management and infection control, ensuring precision and efficiency is crucial for patient care. Dealing with patients affected by viruses and infections underscores the need for innovative solutions to minimize human exposure to pathogens and enhance accuracy in medical processes. To address these challenges, this project introduces the Robotic Mobile Arm, a versatile robotic system designed for both manual and autonomous pick-and-place tasks in healthcare environments.

The Robotic Mobile Arm integrates several advanced components: a mobile app for manual control, a Raspberry Pi as the main control unit, and an Arduino for executing tasks. The system is equipped with a YOLOv8 model trained to recognize various medications, ensuring precise drug handling. Additionally, the arm features environmental awareness through sensors, enabling it to navigate specific routes autonomously. In autonomous mode, the arm can detect drug boxes in its path and trigger an alert sound via a buzzer.

By automating repetitive and potentially hazardous tasks, the Robotic Mobile Arm enhances accuracy and reduces the risk of human error and exposure to infections. The integration of these advanced technologies ensures reliable performance, allowing medical staff to concentrate on more critical tasks. The results demonstrate that the Robotic Mobile Arm improves procedural consistency and patient safety, showcasing its potential to transform healthcare practices and enhance outcomes.

List of Figures

Chapter One

Introduction

In recent years, the integration of robotics in the medical field has transformed various aspects of healthcare, leading to advancements in surgical procedures, diagnostics, and patient care. One of the pivotal innovations in this domain is the development of robotic arms, which offer precision, reliability, and efficiency. This project focuses on the design and implementation of a simple robotic arm tailored for medical applications.

The motivation behind this project stems from the increasing demand for minimally invasive procedures and the necessity for precision in delicate medical tasks. Traditional manual methods often face limitations in terms of precision, fatigue, and consistency. Robotic arms, however, can operate with unparalleled accuracy and repeatability, significantly reducing the margin of error in medical procedures.

Our project aims to create a robotic arm specifically designed for pick-and-place tasks in medical applications. This includes handling delicate tissues, organising medical instruments, and assisting with the precise placement of items during surgical procedures. The robotic arm is engineered to be user-friendly, cost-effective, and easily integrable into existing medical infrastructure. It incorporates advanced technologies, such as motion control, sensor integration, and intuitive user interfaces, to enhance its functionality and ease of use.

This documentation outlines the design process, implementation details, and potential applications of the robotic arm in the medical field. It also discusses the challenges encountered during the development phase and the solutions implemented to overcome them. By the end of this project, we aim to demonstrate the viability and advantages of using a simple robotic arm in medical applications, highlighting its potential to improve patient outcomes and optimise healthcare.

1.1 problem Definition

In the fast-evolving medical field, the need for precision, reliability, and efficiency in handling delicate tasks is paramount. Traditional manual methods often fall short due to limitations such as human error, fatigue, and variability in performance, which can negatively impact patient outcomes and procedural consistency. Several specific problems have been identified that highlight the necessity for an innovative solution:

1. **Efficiency in Repetitive Tasks:**
   - **Current Challenge:** Medical environments often involve repetitive tasks such as organising surgical instruments, placing materials, and preparing for procedures. These tasks can become monotonous and lead to decreased human efficiency and increased likelihood of errors.
   - **Implications:** Fatigue and loss of focus during repetitive tasks can compromise the speed and quality of medical procedures, impacting overall workflow and patient care.
2. **Consistency in Medical Procedures:**
   - **Current Challenge:** Variability in human performance leads to inconsistency in the execution of medical procedures. This can affect the reliability of outcomes and the standardization of care.
   - **Implications:** Inconsistent procedural outcomes can result in unequal patient experiences and variable success rates, undermining the overall quality of healthcare services.
3. **Integration with Existing Infrastructure:**
   - **Current Challenge:** Many advanced robotic solutions available today are expensive, complex, and difficult to integrate into existing medical settings. This poses a barrier to their widespread adoption in healthcare facilities.
   - **Implications:** High costs and integration challenges can limit access to advanced robotic technologies, particularly in resource-constrained environments, thereby restricting the benefits these technologies can offer.

This project addresses these critical issues by developing a robotic arm specifically designed for pick-and-place tasks in medical applications. The proposed robotic arm aims to:

- **Enhance Precision:** By utilizing advanced motion control and sensor integration, the robotic arm will perform delicate tissue handling with unparalleled accuracy, minimizing the risk of damage.
- **Improve Efficiency:** Automating repetitive tasks will reduce human fatigue and increase overall efficiency, ensuring a smoother and faster workflow in medical environments.
- **Ensure Consistency:** The robotic arm will provide consistent performance in medical procedures, standardizing outcomes and enhancing the reliability of patient care.
- **Facilitate Integration:** Designed to be user-friendly and cost-effective, the robotic arm will seamlessly integrate into existing medical infrastructure, making advanced robotic technology accessible to a wider range of healthcare facilities.

Through this project, we aim to demonstrate the feasibility and benefits of employing a simple robotic arm in the medical field, ultimately contributing to improved patient outcomes and optimized healthcare delivery.

1.2 Motivation

The motivation behind developing the Robotic Mobile Arm in the medical field stems from the critical need to enhance precision, efficiency, and consistency in healthcare operations, leveraging its versatile control options to cater to diverse operational requirements. As healthcare continues to evolve, traditional manual methods often prove insufficient, necessitating the integration of advanced robotic solutions. Here are the key motivations guiding this project:

Enhancing Operational Efficiency:

Current Challenge: Repetitive tasks and logistical operations in medical environments can be time-consuming and prone to human error, impacting overall operational efficiency.

Robotic Solution: The Robotic Mobile Arm offers both manual control via a mobile app and autonomous navigation capabilities. This dual functionality optimizes task management, allowing healthcare staff to focus on complex patient care needs, thereby enhancing overall efficiency.

Ensuring Reliable and Adaptable Performance:

Current Challenge: Variability in human performance can lead to inconsistencies in medical procedures, affecting reliability and standardization.

Robotic Solution: Through its integrated IR sensors for route detection, ultrasonic sensors for obstacle avoidance with audible alerts, and a Raspberry Pi-based camera system with machine learning capabilities for medical box detection, the Robotic Mobile Arm ensures precise and adaptable task execution. This enhances procedural consistency and operational reliability across healthcare settings.

Facilitating Seamless Integration with Existing Infrastructure:

Current Challenge: Complex and costly robotic technologies may present barriers to integration within existing healthcare systems.

Robotic Solution: By developing a cost-effective and user-friendly robotic arm, this project aims to facilitate seamless integration with current medical infrastructure. This approach promotes accessibility and scalability, enabling broader adoption of advanced robotic technologies across diverse healthcare facilities.

Improving Work Environment and Patient Care:

Current Challenge: Healthcare professionals often face physical and cognitive strain during demanding procedures, impacting their performance and well-being.

Robotic Solution: Automating repetitive and physically taxing tasks with the Robotic Mobile Arm alleviates strain on medical staff, reducing errors and enhancing job satisfaction. This contributes to a safer and more efficient healthcare environment, ultimately improving patient care outcomes.

Driving Innovation and Adoption of Robotic Technology:

Current Challenge: Slow adoption of technological advancements in healthcare due to complexity and implementation barriers.

Robotic Solution: By demonstrating the practical benefits of robotic automation in healthcare operations, this project aims to accelerate the adoption of innovative technologies. The Robotic Mobile Arm serves as a pioneering example of how robotics can optimize medical workflows, driving continuous improvement in healthcare delivery.

1.3 Objectives

The primary objective of this project is to develop a versatile Robotic Mobile Arm tailored for diverse tasks in medical environments, emphasizing its innovative functionalities and operational capabilities. This overarching goal encompasses the following specific objectives:

Design and Development:

- **Objective:** Design and prototype a Robotic Mobile Arm equipped with advanced mobility and sensing capabilities suitable for dynamic medical environments.
- **Outcome:** Develop detailed specifications and functional prototypes that meet the specific operational requirements of medical applications, integrating state-of-the-art control systems and adaptive sensors.

Precision and Adaptability:

- **Objective:** Ensure the Robotic Mobile Arm can handle intricate medical tasks with exceptional precision and adaptability to varying operational scenarios.
- **Outcome:** Implement robust motion control algorithms and sensor integration strategies to enable reliable and responsive task execution, enhancing operational flexibility and efficiency.

Operational Efficiency and Automation:

- **Objective:** Automate essential tasks such as navigation, obstacle detection, and object recognition to streamline medical workflows and optimise resource utilisation.
- **Outcome:** Demonstrate significant improvements in efficiency through autonomous operation capabilities, freeing up medical personnel for more specialized patient care responsibilities.

Consistent Performance and Reliability:

- **Objective:** Provide a reliable robotic solution capable of consistent performance across different medical procedures and environmental conditions.

- **Outcome:** Achieve uniformity in task execution to minimize variability, ensuring dependable and predictable outcomes in critical healthcare scenarios.

User-Centric Interface Design:

- **Objective:** Develop an intuitive user interface that simplifies interaction and control of the Robotic Mobile Arm, accommodating diverse user needs in medical settings.
- **Outcome:** Create a user-friendly control system accessible via mobile app and onboard interfaces, enhancing usability and facilitating seamless integration into existing healthcare protocols.

Cost-Effectiveness and Accessibility:

- **Objective:** Design the Robotic Mobile Arm to be cost-effective in production and maintenance, ensuring affordability and accessibility across healthcare facilities.
- **Outcome:** Deliver a cost-efficient solution without compromising quality or performance, promoting widespread adoption and scalability in diverse medical settings.

Seamless Integration with Medical Infrastructure:

- **Objective:** Ensure compatibility and ease of integration with existing medical systems and facilities, minimizing deployment complexities.
- **Outcome:** Develop interoperability standards and protocols that enable seamless integration into diverse healthcare environments, supporting efficient deployment and operational readiness.

Safety and Reliability Assurance:

- **Objective:** Prioritize safety features and reliability in the design and operation of the Robotic Mobile Arm to ensure safe usage in sensitive medical environments.
- **Outcome:** Conduct rigorous testing and validation to verify safety protocols and operational reliability under stringent healthcare standards, ensuring patient and user safety.

Scalability and Future-Proof Design:

- **Objective:** Design a modular Robotic Mobile Arm platform capable of accommodating future technological advancements and evolving medical requirements.
- **Outcome:** Create a scalable architecture that supports easy upgrades and adaptation to emerging healthcare needs, maintaining long-term relevance and effectiveness.

Evaluation and Validation in Clinical Settings:

- **Objective:** Conduct comprehensive evaluation and validation of the Robotic Mobile Arm's performance in real-world medical applications, gathering feedback from healthcare professionals.
- **Outcome:** Validate effectiveness, reliability, and user acceptance through extensive clinical trials and feedback mechanisms, ensuring the device enhances medical procedures and patient care outcomes effectively.

1.4 Methodology

Developing the robotic arm for medical applications follows a systematic and structured approach. This project is segmented into several essential phases: requirement analysis, design, development, testing, and evaluation. Each phase is crucial to ensuring the project's success and the creation of a reliable and effective robotic arm tailored for medical use. Below is a detailed outline of the revised methodology:

1. Requirement Analysis

- **Objective:** Identify and document the specific requirements and constraints for the robotic arm in medical applications.
- **Activities:**
    - Conduct a literature review on existing robotic solutions in the medical field.
    - Engage with medical professionals to gather insights on the specific needs and challenges in medical environments.
    - Define the functional and non-functional requirements of the robotic arm.
    - Establish performance metrics and success criteria.

2. Design

- **Objective:** Develop a detailed design plan for the robotic arm, addressing all identified requirements.
- **Activities:**
    - Create initial design concepts and sketches.
    - Select appropriate materials and components for the robotic arm.
    - Design the mechanical structure, ensuring it is capable of precise and delicate operations.
    - Develop the control system architecture, including motion control algorithms and sensor integration.
    - Design an intuitive user interface for easy operation.

3. Development

- **Objective:** Build a working prototype of the robotic arm based on the design specifications.

- **Activities:**
    - Procure necessary materials and components.
    - Assemble the mechanical parts of the robotic arm.
    - Develop and integrate the control system, including software and hardware components.
    - Implement motion control algorithms and calibrate sensors.
    - Develop the user interface software and ensure compatibility with the robotic system.
    - Train a machine learning model to detect medical drug boxes using image data captured by the onboard camera.

4. Testing

- **Objective:** Validate the performance and reliability of the robotic arm through rigorous testing.
- **Activities:**
    - Conduct unit testing on individual components and subsystems.
    - Perform integration testing to ensure all parts work together seamlessly.
    - Carry out functional testing to verify that the robotic arm meets all specified requirements.
    - Test the robotic arm in simulated medical scenarios to evaluate its precision, accuracy, and consistency.
    - Perform safety testing to ensure the robotic arm operates safely in all conditions.

5. Evaluation

- **Objective:** Assess the overall effectiveness and usability of the robotic arm in real-world medical settings.
- **Activities:**
    - Conduct pilot studies in collaboration with medical professionals to gather practical feedback.
    - Analyze performance data and identify any areas for improvement.
    - Refine the design and functionality based on feedback and test results.
    - Document the findings and compile a comprehensive evaluation report.

6. Documentation and Training

- **Objective:** Provide thorough documentation and training materials for users and maintenance personnel.
- **Activities:**
  - Create user manuals and operation guides for the robotic arm.
  - Develop maintenance and troubleshooting guides.
  - Provide training sessions for medical staff on how to operate and maintain the robotic arm.
  - Ensure all documentation is clear, concise, and easily accessible.

7. Deployment and Integration

- **Objective:** Implement the robotic arm in real medical environments and ensure seamless integration.
- **Activities:**
  - Coordinate with healthcare facilities to plan the deployment.
  - Install the robotic arm and integrate it with existing medical infrastructure.
  - Provide on-site support during the initial deployment phase.
  - Monitor performance and address any issues that arise during early use.

8. Continuous Improvement

- **Objective:** Ensure the robotic arm remains effective and up-to-date with advancements in technology.
- **Activities:**
  - Collect ongoing feedback from users.
  - Implement software updates and hardware upgrades as needed.
  - Conduct periodic reviews and performance assessments.
  - Stay informed about new developments in medical robotics and incorporate relevant innovations.

1.5 Time Plan

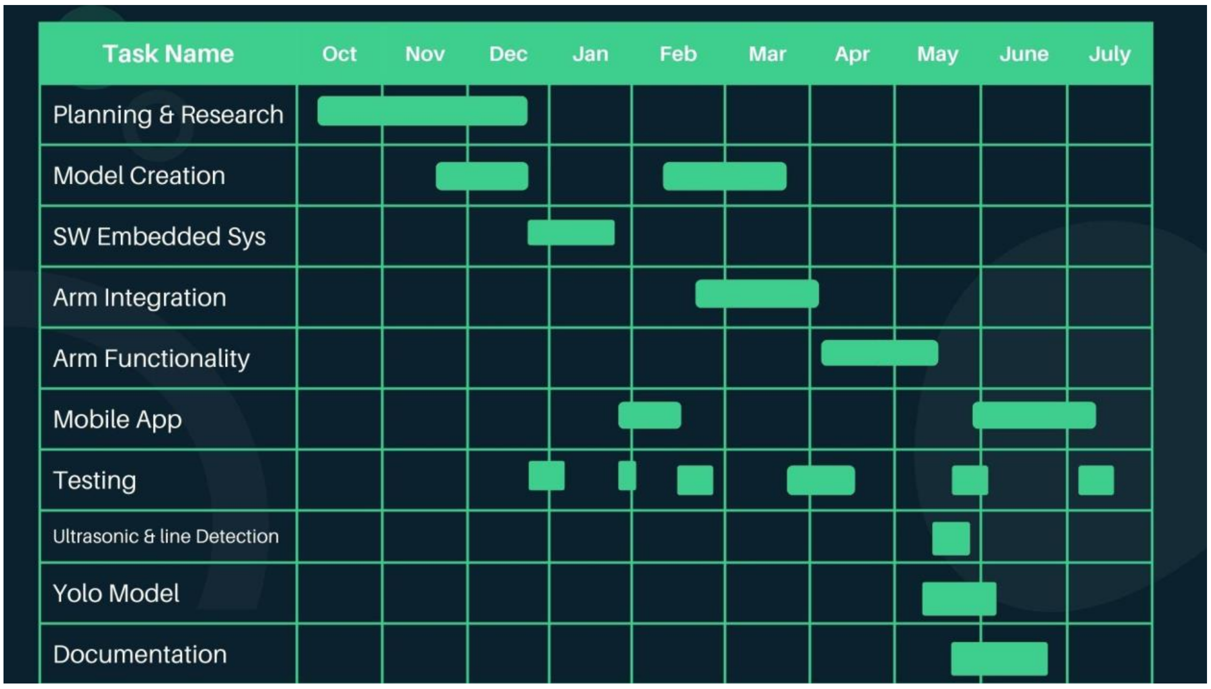


Figure 1.1

1.6 Thesis Outline

This thesis is structured into six main chapters, each detailing a specific aspect of the project, from introduction to conclusion. Below is an overview of each chapter's content and focus:

Chapter 1: Introduction
This chapter provides an overview of the project's background, including the problem definition, motivation, objectives, methodology, and the time plan. It sets the stage for understanding the necessity and scope of the robotic mobile arm designed for medical applications.

Chapter 2: Literature Review
This chapter reviews existing literature and research related to robotic arms, car control systems using Arduino, embedded systems, serial communication between Raspberry Pi and Arduino, Android mobile applications, sensors, and machine learning, particularly the YOLO model for object detection. It establishes the foundation for the project's development by identifying gaps and opportunities in current technologies.

Chapter 3: System Architecture and Methods
This chapter describes the overall system architecture and the methods used in the development of the robotic mobile arm. It covers the hardware and software components, including the integration of various sensors, the control logic, and the communication protocols between different parts of the system.

Chapter 4: System Implementation and Results
This chapter details the implementation process of the robotic mobile arm, including the dataset used, software tools, hardware setup, and experimental results. It provides an in-depth look at the step-by-step development and testing procedures, along with the outcomes of various experiments conducted to validate the system's performance.

Chapter 5: Run the Application
This chapter explains how to run the application, detailing the setup and configuration required for both manual and automatic modes of the robotic mobile arm. It includes user instructions for operating the mobile app, connecting to the hardware, and troubleshooting common issues.

Chapter 6: Conclusion and Future Work
The final chapter summarizes the project's findings, highlighting the achievements and limitations. It discusses the conclusions drawn from the experimental results and proposes potential future work to further enhance the robotic mobile arm's capabilities and applications in the medical field.

References
This section lists all the references and sources cited throughout the thesis, providing a comprehensive bibliography of the literature, tools, and technologies used in the project.

Chapter Two
Literature Review

2.1 Background

In recent years, advancements in robotics and embedded systems have revolutionized various industries, including healthcare, by introducing automated solutions to enhance efficiency and precision in complex tasks. Understanding the foundational technologies behind these innovations is crucial for developing applications such as a Robotic Mobile Arm controlled by Arduino, interfaced with Raspberry Pi, and managed via an Android mobile app.

Robotic Arms: Robotic arms are mechanical devices designed to mimic human arm movement and perform tasks autonomously or under remote control. They are equipped with joints, actuators, and end-effectors that enable them to manipulate objects with precision. These arms are widely used in manufacturing, medicine, and research due to their ability to perform repetitive tasks consistently and accurately.

Arduino-Based Control Systems: Arduino is an open-source electronics platform based on easy-to-use hardware and software. It is commonly used for prototyping and developing embedded systems due to its versatility, affordability, and extensive community support. Arduino boards are equipped with microcontrollers that can interface with sensors, actuators, and other electronic components, making them ideal for controlling robotic systems.

Embedded Systems: Embedded systems are specialized computing systems designed to perform dedicated functions within larger systems or devices. They typically consist of microcontrollers or microprocessors embedded into hardware, often with real-time computing constraints. In the context of robotics, embedded systems play a critical role in controlling movement, processing sensor data, and executing algorithms to achieve desired behaviors.

Serial Communication: Serial communication is a common method used to establish data exchange between microcontrollers, sensors, and other peripheral devices. In robotics projects involving Arduino and Raspberry Pi, serial communication (such as UART) enables these devices to exchange commands, sensor data, and control signals. This communication is essential for coordinating actions between different components of the robotic system.

Android Mobile Apps: Mobile applications developed for Android platforms provide user-friendly interfaces to interact with and control robotic systems remotely. These apps can send commands, receive data from sensors, visualize real-time information, and provide feedback to users. In the context of the Robotic Mobile Arm project, an Android app facilitates intuitive control and monitoring capabilities, enhancing user experience and operational flexibility.

Sensor Interfaces: Sensors are fundamental components in robotics that detect and measure physical quantities such as proximity, temperature, and motion. They provide essential feedback for decision-making and control algorithms. Sensor interfaces in robotic systems involve configuring sensors to interface with microcontrollers (e.g., Arduino), processing sensor data for decision-making, and integrating sensor outputs into higher-level control systems.

Machine Learning in Robotics: Machine learning enables robotic systems to learn from data, make decisions, and improve performance over time without explicit programming for every scenario. In robotics, ML algorithms are applied to tasks such as object detection, path planning, and decision-making, enhancing autonomy and adaptability in dynamic environments. The integration of ML allows robots to perceive and respond to their surroundings intelligently, crucial for tasks requiring object recognition and navigation.

YOLO (You Only Look Once) Trained Model: YOLO is a state-of-the-art real-time object detection system that uses convolutional neural networks (CNNs) for fast and accurate detection of objects in images. Unlike traditional object detection methods that classify objects in multiple stages, YOLO performs detection in a single step, making it highly efficient for real-time applications. YOLO models are trained on large datasets to recognize and localize objects within an image, enabling robots equipped with vision systems to identify and interact with objects autonomously.

Application in the Robotic Mobile Arm: In the context of the Robotic Mobile Arm project, a YOLO trained model integrated into the Raspberry Pi's processing unit enables the robotic system to perceive its environment visually. Using a camera mounted on the Robotic Mobile Arm, the YOLO model processes live video feed to detect medical boxes or other relevant objects in its path. This capability enhances the robot's ability to autonomously navigate, locate objects of interest, and perform pick-and-place tasks effectively in medical settings.

Advantages of ML in Robotics:

- **Enhanced Perception:** ML algorithms improve object recognition and classification accuracy, crucial for tasks requiring precise interaction with diverse objects in complex environments.
- **Real-time Decision-making:** The efficiency of YOLO models allows robotic systems to make rapid decisions based on real-time visual data, enabling swift and responsive actions.
- **Adaptability:** ML facilitates learning and adaptation to new environments and scenarios, making robotic systems more versatile and capable of handling diverse tasks autonomously.

Chapter Three

3.1 System Architecture

Our robotic arm system for pick-and-place tasks in medical settings is structured into three primary layers: the Application Layer, the Communication Layer, and the Physical Layer. Each layer is integral to the functionality and efficiency of the overall system, with specific roles and components designed to ensure smooth operation.

1. Application Layer

The Application Layer is responsible for user interaction and initiating the control process. This layer includes a mobile application that facilitates the control and monitoring of the robotic arm.

- **Mobile Application:**
    - **Initiate Connection:**
        - Establishes a connection with the Raspberry Pi, which hosts the server.
        - Provides a user interface for setting up the connection, ensuring it is secure and reliable.
    - **Send Data:**
        - Allows users to input commands and data, which are then transmitted to the server.
        - Provides real-time feedback on the status and actions of the robotic arm.
        - Choose Control Method**:** Allows users to select a control mode or method for the system (self controlling or with mobile app).
    - **Features:**
        - Touchscreen controls, visual feedback, task customization, and monitoring tools.
        - User-friendly design to ensure ease of use for medical professionals.

2. Communication Layer

The Communication Layer acts as the bridge between the Application Layer and the Physical Layer, managing data transfer and command processing.

- **Raspberry Pi (Running Flask Server):**
    - **Receive Data:**
        - The Flask server hosted on the Raspberry Pi receives data and commands from the mobile application.
        - Processes incoming requests and ensures they are properly formatted for further transmission.
    - **Transmit Data:**
        - Sends processed data to the Arduino Uno for execution.
        - Handles communication protocols and ensures data integrity and security during transmission
    - **Automatic System (Object Detection):**
        - Incorporates a YOLO model for object detection
        - The car can move by line detection with IR sensor

3. Physical Layer

The Physical Layer encompasses the hardware components that perform the actual pick-and-place tasks.

- **Arduino Uno:**
    - **Receive Data from Raspberry Pi:**
        - Receives commands and data from the Flask server running on the Raspberry Pi.
        - Interprets the received data and translates it into actionable commands for the motors.
    - **Send Commands to Motors:**
        - Controls the motor drivers based on the commands received.
        - Ensures precise execution of tasks as per the input from the application layer.

- **Motor Drivers:**
  - **Control Motors:**
    - Interface between the Arduino Uno and the motors.
    - Regulate the power and movement of the motors to perform the desired actions.
- **Motors:**
  - **Execution:**
    - Perform the physical movements required for pick-and-place tasks.
    - Operate with high precision to handle delicate and repetitive tasks in the medical environment.

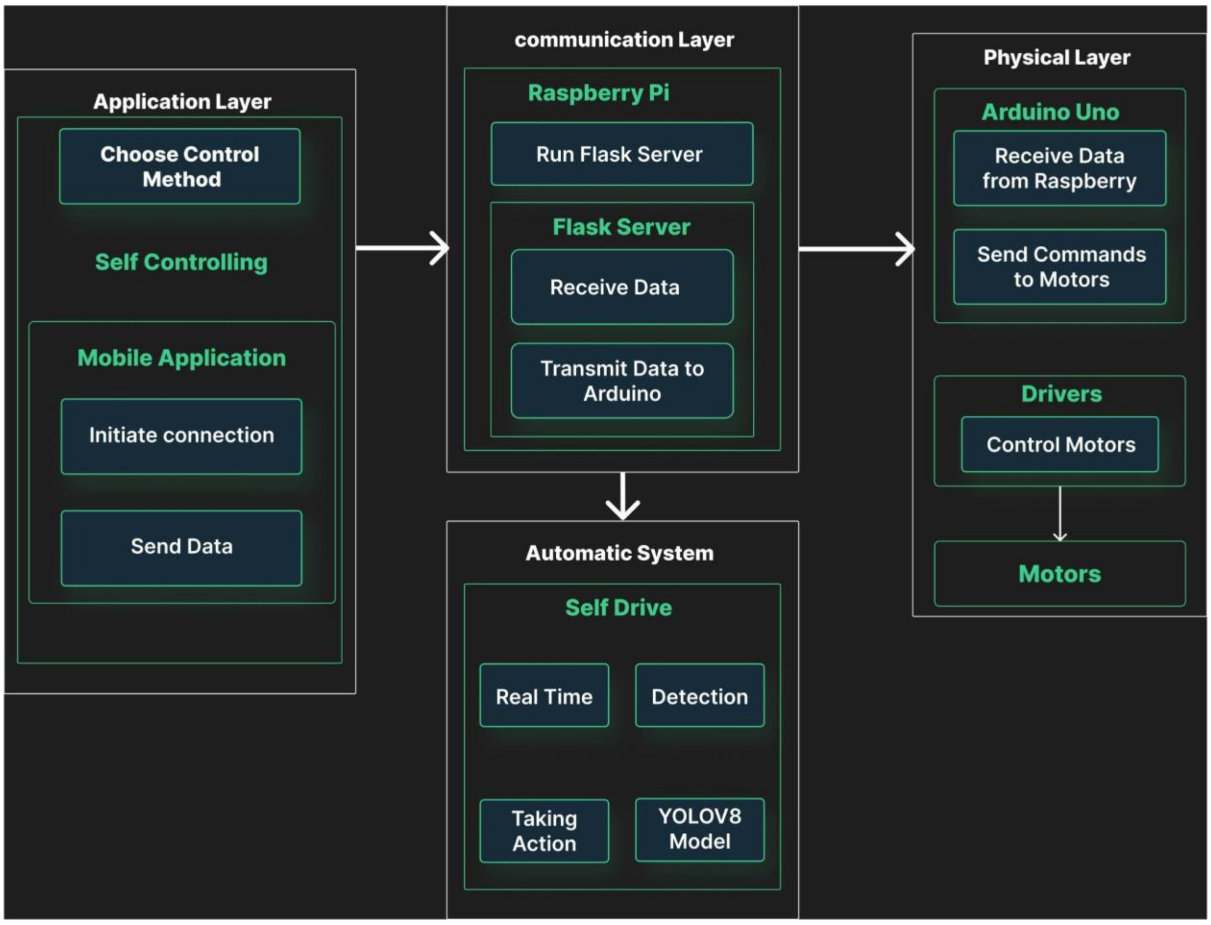


Figure 3.1

3.2 Description of methods and procedures used

The development and implementation of the robotic arm for pick-and-place tasks in medical settings involve several well-defined methods and procedures. These procedures ensure that each component functions seamlessly within the system architecture, providing reliable and precise operations. The methods and procedures can be divided into the following stages:

1. System Design and Planning

- Objective: Define the system architecture and plan the integration of components.
- Activities:
  - Requirement Analysis: Identify the specific needs and constraints of the medical environment. Gather inputs from healthcare professionals to understand the tasks the robotic arm will perform.
  - Component Selection: Choose appropriate hardware and software components for each layer of the system (Application, Communication, Physical).
  - Architecture Design: Develop a detailed design of the three-layer architecture, outlining the roles and interactions of each component.

2. Development of Mobile Application (Application Layer)

- Objective: Create a user-friendly interface for controlling the robotic arm.
- Activities:
  - UI/UX Design: Design an intuitive graphical user interface (GUI) that allows users to initiate connections and send commands to the robotic arm.
  - Connectivity Implementation: Develop functionality to establish and maintain a secure connection with the Raspberry Pi server.
  - Command Interface: Implement features to input, send, and receive data related to pick-and-place tasks.
  - There is two ways of control (self controlling or mobile app)

3. Setting Up Flask Server on Raspberry Pi (Communication Layer)

- Objective: Facilitate data communication between the mobile application and the Arduino.
- Activities:
    - Flask Server Development: Write and configure the Flask application to handle HTTP requests from the mobile application.
    - Data Processing: Implement logic to receive, parse, and format data from the mobile application for transmission to the Arduino.
    - Security Measures: Ensure secure data transmission using encryption and authentication protocols.

4. Integration with Arduino Uno (Physical Layer)

- Objective: Control the motors and execute commands received from the Raspberry Pi.
- Activities:
    - Firmware Development: Write the Arduino code to interpret data received from the Raspberry Pi and control motor drivers accordingly.
    - Motor Control Logic: Implement precise motor control algorithms to ensure accurate execution of pick-and-place tasks.
    - Sensor Integration: Incorporate sensors for feedback and error detection to enhance precision and safety.

5. Motor and Driver Configuration (Physical Layer)

- Objective: Ensure the motors operate correctly and efficiently.
- Activities:
    - Driver Setup: Connect motor drivers to the Arduino Uno and configure them to control the motors based on input signals.
    - Calibration: Calibrate the motors to ensure they move accurately and smoothly. Adjust parameters such as speed, torque, and range of motion.
    - Testing: Perform extensive testing to validate the motor control system, ensuring that the motors respond correctly to commands.

6. System Integration and Testing

- Objective: Ensure all components work together seamlessly.
- Activities:
    - Component Integration: Assemble the mobile application, Flask server, Arduino, motor drivers, and motors into a cohesive system.
    - End-to-End Testing: Conduct comprehensive tests to verify that data flows correctly from the mobile application through the Flask server to the Arduino and finally to the motors.
    - Functional Testing: Perform specific pick-and-place tasks to ensure the robotic arm operates as expected in real-world scenarios.
    - Performance Evaluation: Assess the system's performance in terms of speed, accuracy, and reliability. Identify and resolve any issues.

7. Deployment and User Training

- Objective: Deploy the robotic arm system in the medical setting and train users.
- Activities:
    - Installation: Set up the robotic arm system in the designated medical environment. Ensure all connections and components are correctly installed.
    - Training: Provide training sessions for medical staff to familiarize them with the operation of the mobile application and the robotic arm. Cover aspects such as initiating connections, sending commands, and handling common issues.
    - Documentation: Prepare comprehensive user manuals and technical documentation to support ongoing operation and maintenance.

8. Monitoring and Maintenance

- Objective: Ensure the long-term reliability and efficiency of the system.
- Activities:
  - Continuous Monitoring: Implement a monitoring system to track the performance and health of the robotic arm.
  - Routine Maintenance: Schedule regular maintenance checks to inspect and service the robotic arm, ensuring it remains in optimal working condition.
  - Updates and Improvements: Periodically update the software and firmware to incorporate new features, security patches, and performance enhancements based on user feedback and technological advancements.

Chapter Four

System Implementation and Results

4.1 Dataset:

Mobile Captured Pharmaceutical Medication Packages of prescription and over-the-counter drug information and images. It combines data from pharmaceutical companies, the Food and Drug Administration (FDA), the National Institutes of Health, and the Department of Veterans Affairs.

1) A total of 3900 medication pack images were captured at different random angles. There are 26 images per pack.

2) Backgrounds are made of paper, cardboard, and plastic.

3) No transparent, reflexive, or patterned backgrounds are used.

4) 300 images were captured while enabling the flashlight.

5) 3600 images were captured in different lighting conditions without using a flashlight. Some images were even captured at dawn time to try the worst lighting conditions.

6) Pharmaceutical packs only were captured. Nothing was captured without its pack.

7) Captured medication packs contain tablets, capsules, syrups, creams, ampoules, gels, drops, ointments, and other types. Two types of special medical tapes were also captured.

8) 150 different pharmaceutical packages were captured. If there is a difference in the dose, size, or type, it is considered a different pack.

9) Some images were captured by different people after receiving the instructions to simulate a real patient experience.

We prepared dataset with different shape of the same drug box to give us the best accuracy

Sample from the dataset:

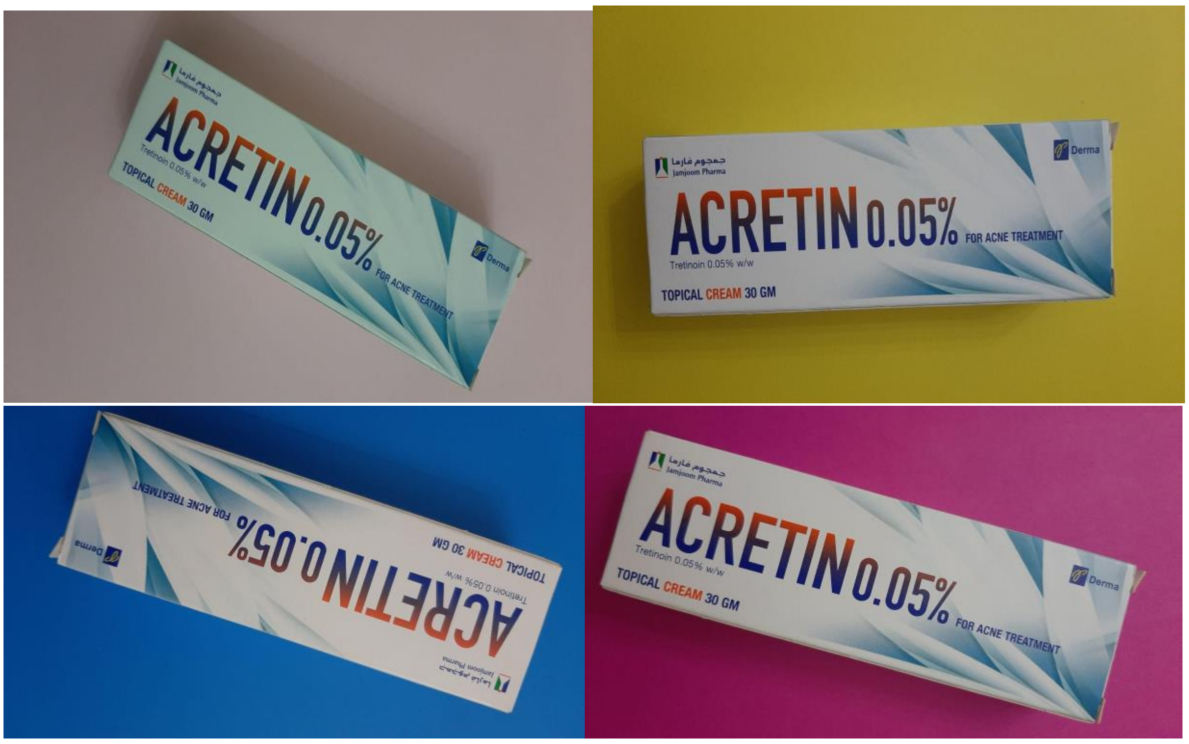


Figure 4.1

4.2 Description of Software Tools Used:

- **PyCharm**: This IDE is a dedicated Python Integrated Development Environment (IDE) providing a wide range of essential tools for Python developers, tightly integrated to create a convenient environment for productive Python, web, and data science development. The main issue of using PyCharm is it takes up more space than other text editors, which degrades the functionality of code. The community version is idle for python development only and does not allow the use of other programming languages. The professional version is somewhat expensive. PyCharm was released to the market of the Python-focused IDEs to compete with PyDev (for Eclipse) or the more broadly focused Komodo IDE by Active State. The beta version of the product was released in July 2010, with the 1.0 arriving 3 months later History. PyCharm was released to the market of the Python-focused IDEs to compete with PyDev (for Eclipse) or the more broadly focused Komodo IDE by Active State. The beta version of the product was released in July 2010, with 1.0 arriving 3 months later.

- **Android Studio:**
  Android Studio is the official Integrated Development Environment (IDE) for Android application development, created by Google. It provides a comprehensive suite of tools for designing, coding, testing, and debugging Android apps. The IDE offers a user-friendly interface, powerful code editor, and integrated emulator. You can download the latest version from the official website, and it supports Windows, macOS, and Linux.

- **Arduino IDE:**

  The Arduino IDE is an open-source software that simplifies writing code and uploading it to your Arduino board. It provides a user-friendly interface for programming and communication with Arduino hardware. You can download the latest version from the official website, and it supports Windows, Mac OS, Linux, and even a portable version.

- **Raspbian**:

  A Unix-like operating system based on the Debian Linux distribution for the Raspberry Pi family of compact single-board computers.
  The main issues of Raspbian(bullseye) are that the media pipe isn't compatible with this version so we can replace it with a buster version.
  Raspberry Pi OS was first developed by Mike Thompson and Peter Green as Raspbian, an independent and unofficial port of Debian to the Raspberry Pi. The first build was released on July 15, 2012.

- **VNC Viewer**:

  VNC Viewer is a remote access software that allows users to control other devices running VNC Server from a distance. It supports multiple platforms including Windows, Linux, macOS, Raspberry Pi, iOS, Android, and HTML5-enabled browsers. VNC Viewer can manage connectivity between devices using VNC Cloud or establish direct connections within a network. For secure usage, it emphasises the importance of strong passwords and encryption.

- **Geany IDE:**

  Geany IDE: Geany is a lightweight and fast Integrated Development Environment (IDE) that supports multiple programming languages. It offers a simple and user-friendly interface with features like syntax highlighting, code folding, and auto-completion. Geany is highly extensible through plugins and provides a built-in terminal for executing code. You can download Geany for Windows, macOS, and Linux from the official website. It is designed to be minimal yet powerful, making it suitable for both beginners and experienced developers. Its main functions include:

  1. **Code Editing:** Geany supports essential code editing features such as cut and paste, search (including limited expressions), replace, indentation, code folding, syntax highlighting (for over 30 common languages), line wrapping.
  2. **Cross-Platform Compatibility:** Geany works on Linux, Windows, and macOS, making it versatile for developers across different operating systems3.
  3. **Plugin Manager:** It comes with a built-in plugin manager, allowing users to extend its functionality with additional features.
  4. **Syntax Parsing:** Geany includes a real syntax parser, which aids in viewing methods and inner classes in Java source files.

4.3 Step-up Configuration (Hardware):

- **Raspberry Pi 3**

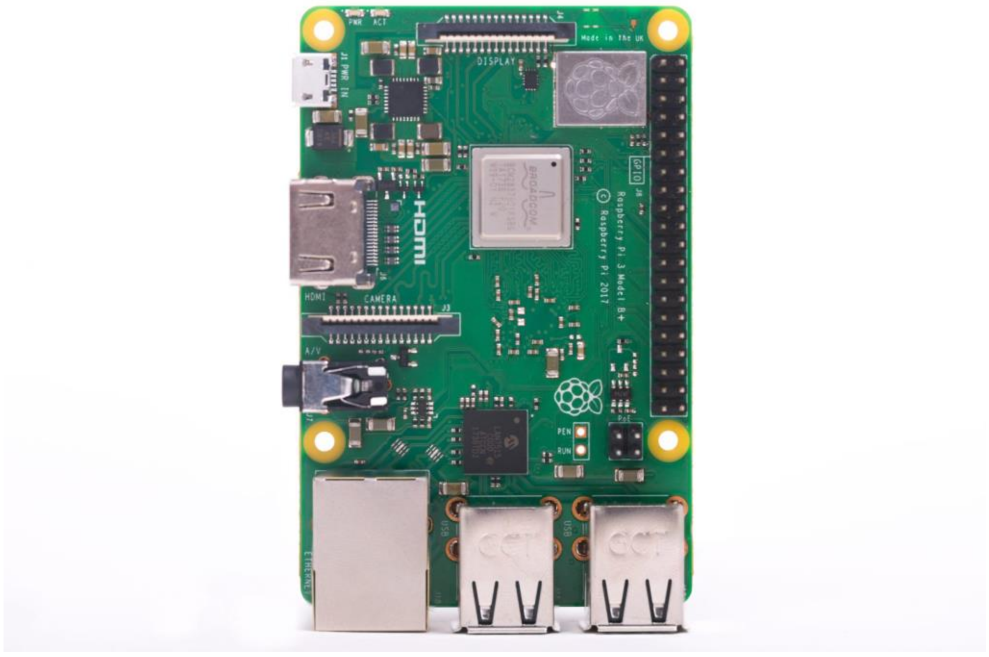


Figure 4.2

The Raspberry Pi 3 is a compact, affordable, and versatile single-board computer developed by the Raspberry Pi Foundation. It features a 1.2GHz 64-bit quad-core ARM Cortex-A53 processor, 1GB of RAM, and built-in Wi-Fi and Bluetooth capabilities. The Raspberry Pi 3 supports various operating systems, including Raspbian, a Debian-based OS. It is ideal for projects ranging from basic computing tasks to complex embedded systems, robotics, and IoT applications. The board has multiple GPIO pins, USB ports, HDMI output, and a microSD slot for storage, making it a popular choice for hobbyists and educators alike.

- Raspberry Pi Camera v2:

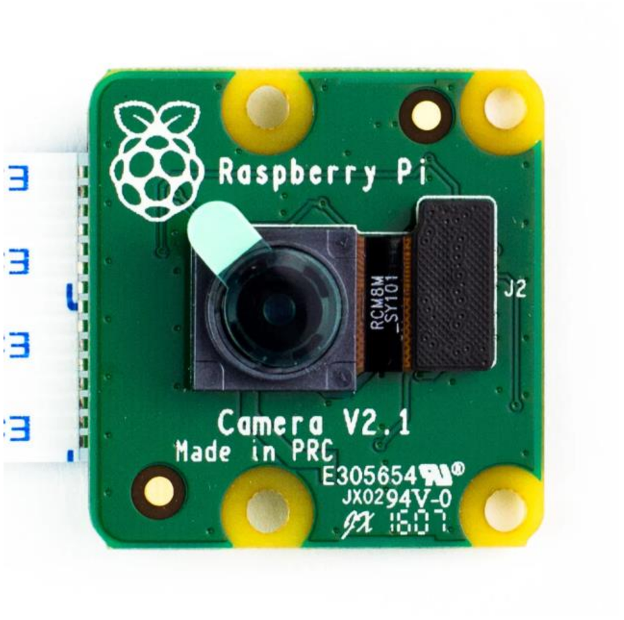


Figure 4.3

The Raspberry Pi Camera v2 is a high-quality, 8-megapixel camera module designed specifically for the Raspberry Pi. It is capable of capturing still images with a resolution of 3280 x 2464 pixels and recording 1080p HD video at 30 frames per second. The camera connects to the Raspberry Pi via the CSI (Camera Serial Interface) port using a flexible ribbon cable. It is compatible with various Raspberry Pi models, making it suitable for a wide range of projects, including surveillance, photography, and computer vision applications. The Camera v2 is supported by the official Raspberry Pi operating system, Raspbian, and is easily programmable using libraries such as picamera and OpenCV.

- Arduino Uno

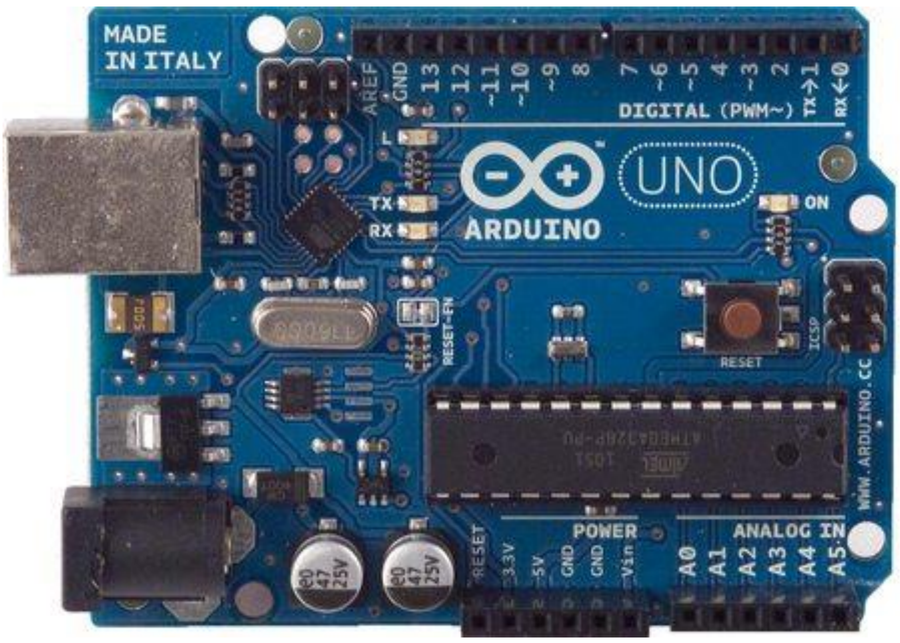


Figure 4.4

The Arduino Uno is a popular open-source microcontroller board based on the ATmega328P microcontroller. It features 14 digital input/output pins (6 of which can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header, and a reset button. The Uno is the ideal board for beginners and educational purposes, offering a simple yet powerful platform for creating a wide range of interactive electronic projects. It is programmable with the Arduino IDE, which supports Windows, macOS, and Linux, and provides an easy-to-use interface for writing and uploading code to the board. The Uno is widely used in hobbyist and professional projects, from basic LED blinking to complex robotics and IoT applications.

• Drivers

- PCA9685 (Servo driver)

The PCA9685 is a 16-channel PWM controller designed for driving servos and LEDs. It offers 12-bit resolution for precise control of up to 16 servos or LEDs via an I2C interface

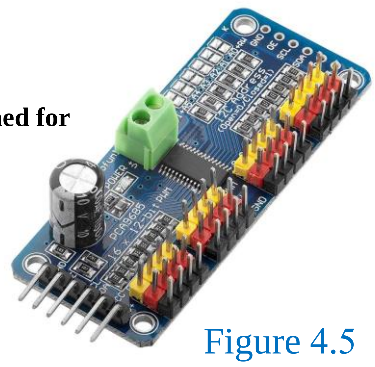


Figure 4.5

-L298 Motor Driver

The L298 is a dual H-bridge motor driver IC capable of driving two DC motors or a single stepper motor. It supports a wide range of supply voltages (up to 46V) and can deliver up to 2A of continuous current per channel, with peak currents up to 3A.

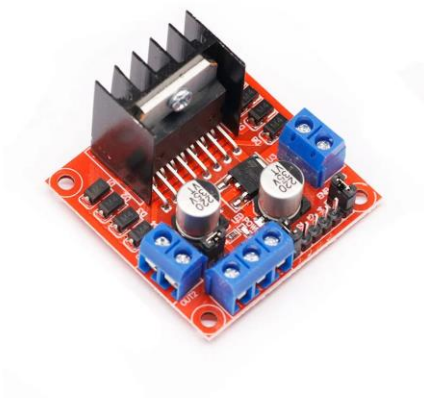


Figure 4.6

• Motors:

- Servo Motor:

Figure 4.7

- Metal Gear Servo Motor

Figure 4.8

- DC Gear Motor (with Wheel 65mm):

Figure 4.9

- **Arm model:**

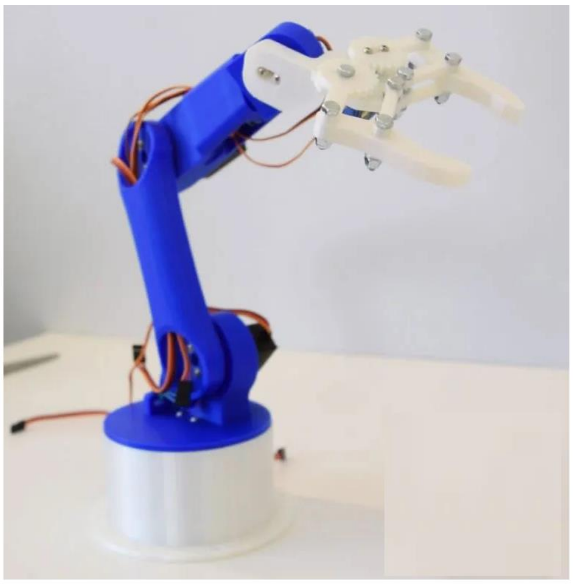


Figure 4.10

- **Car model:**

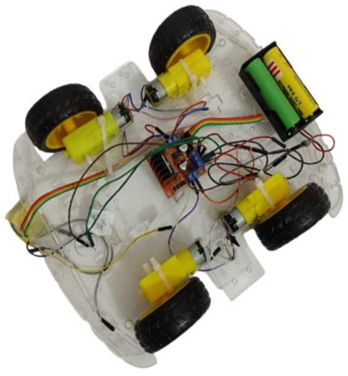


Figure 4.11

- **Integrated model (Car + Arm):**

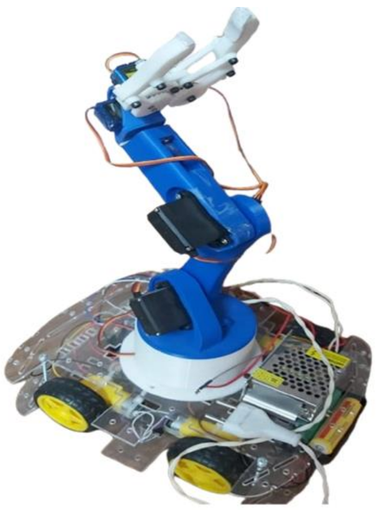


Figure 4.12

- **5V Power Supply:**

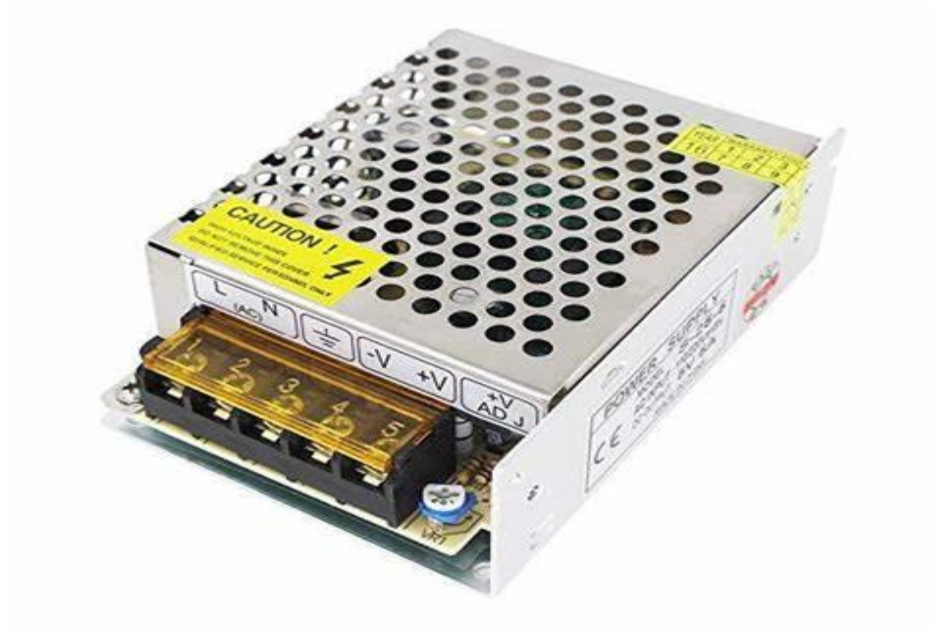


Figure 4.13

- **Batteries:**
  - **Lithium Battery 3.7 V**

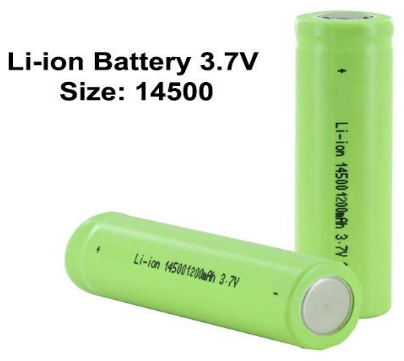


Figure 4.14

4.4 Experimental and Results

- **Car Base:**

  - The car base of the Robotic Mobile Arm system is designed to facilitate smooth and accurate movement. It features a sturdy platform equipped with four motors connected to a motor driver that controls their operation. This configuration ensures stable and precise mobility. Additionally, the car base is fitted with two infrared (IR) sensors, which are used for line detection. These sensors detect the contrast between the predefined path (dark line) and the surrounding surface, enabling the car to follow the line autonomously. This combination of motors and IR sensors provides reliable and efficient navigation, making the car base an essential part of the robotic system for performing tasks in various environments.

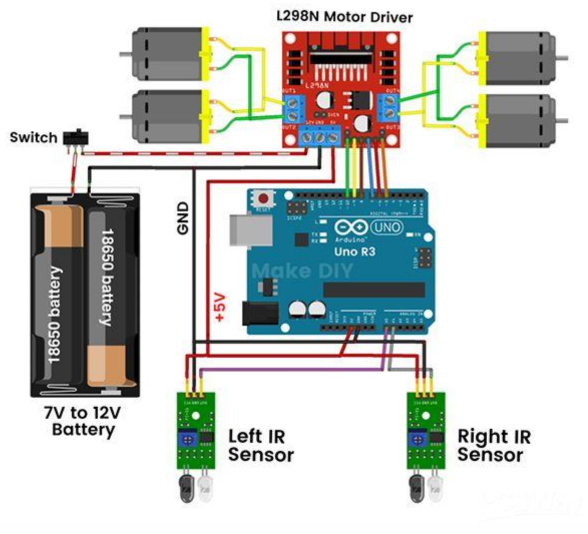


Figure 4.15

  - Result:

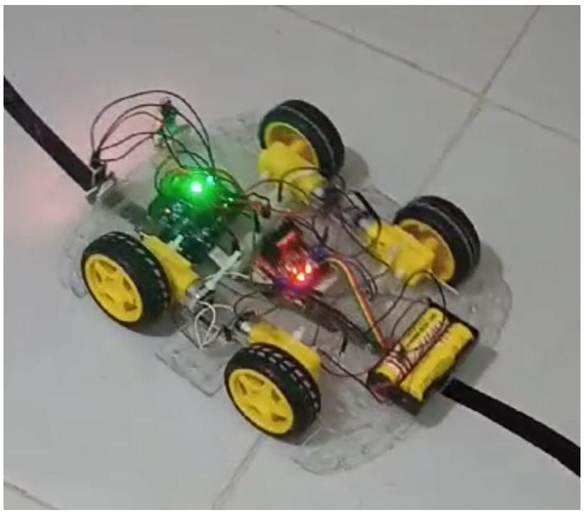


Figure 4.16

- **Robotic Arm**
  - The robotic arm is a 5-degree-of-freedom (DOF) system, designed for precise and versatile operation in medical applications. It consists of six servos that control the following movements: gripper, wrist roll, wrist pitch, elbow, shoulder and waist. Each servo is responsible for moving a specific part of the arm, allowing it to perform tasks with high accuracy. The servos are connected to a servo driver, which manages their movements and ensures coordinated control. The servo driver is, in turn, connected to an Arduino Uno. The gripper, located at the end of the arm, is capable of picking up and handling objects, making the robotic arm suitable for tasks such as organizing instruments and handling delicate tissues in medical settings

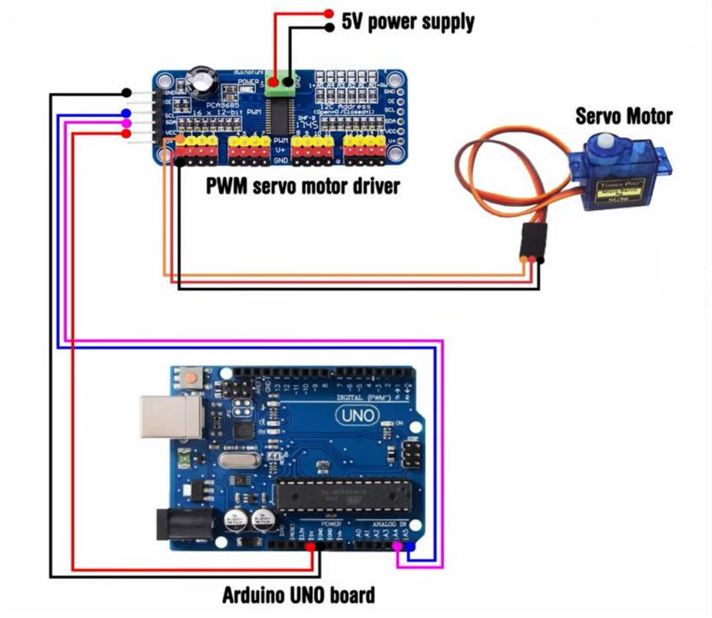


Figure 4.17

  - Result:

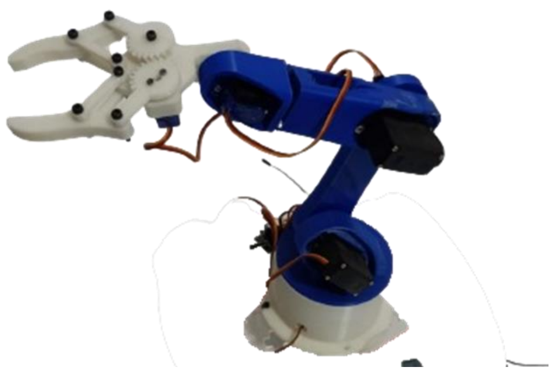
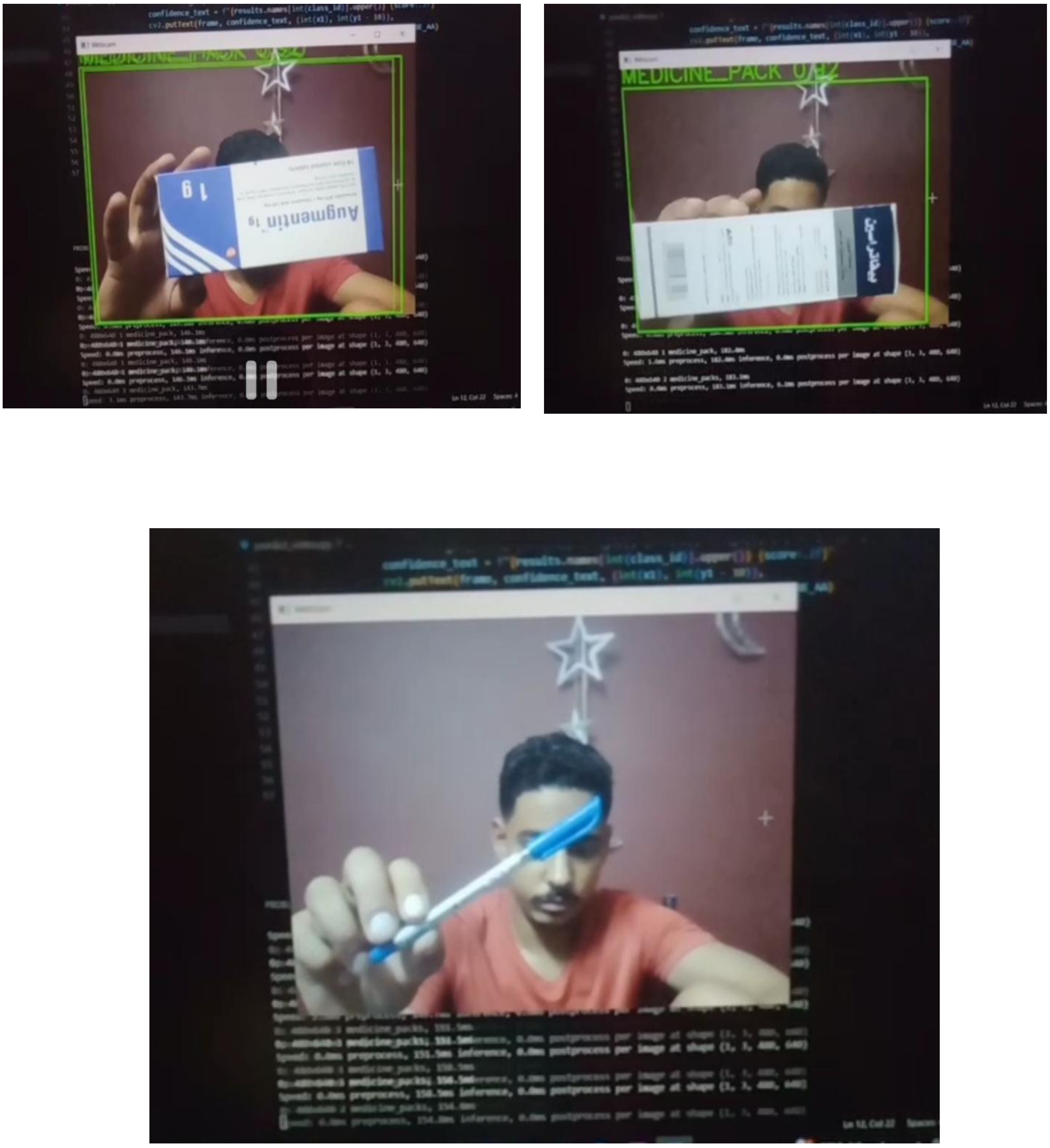


Figure 4.18

- **YOLO Model:**
    - This project involves real-time object detection using a custom-trained YOLO (You Only Look Once) model. The objective is to utilize the YOLO model to detect objects captured via camera and display bounding boxes around the detected objects with their respective labels and confidence scores. The model is loaded from a pre-trained weight file and performs detection in real-time, drawing rectangles and displaying the class label along with the confidence score for each detected object.
- **Problems Faced:**
    - Several challenges were encountered during the development and deployment of this project. One of the main problems was the adjustment of the number of epochs during the training phase. The model's performance was significantly impacted when the number of epochs was not adequately tuned. Training for too few epochs resulted in an underfitted model, which failed to learn the necessary features from the dataset, leading to poor detection accuracy. Conversely, training for too many epochs led to overfitting, where the model performed well on the training data but failed to generalize to new, unseen data.
    - Another critical issue was the insufficient number of images used for training. A limited dataset restricted the model's ability to generalize and recognize various objects accurately. This limitation often resulted in low confidence scores and inaccurate detections. To improve the model's performance, it was essential to augment the dataset with more images representing different scenarios and object variations. Expanding the dataset and properly tuning the number of epochs during training are crucial steps to achieving robust and reliable object detection performance in real-time applications.



```
confidence_text = f"{results.names[int(class_id)].upper()} {score:.2f}"
cv2.putText(frame, confidence_text, (int(x1), int(y1 - 10)),
```

Figure 4.19

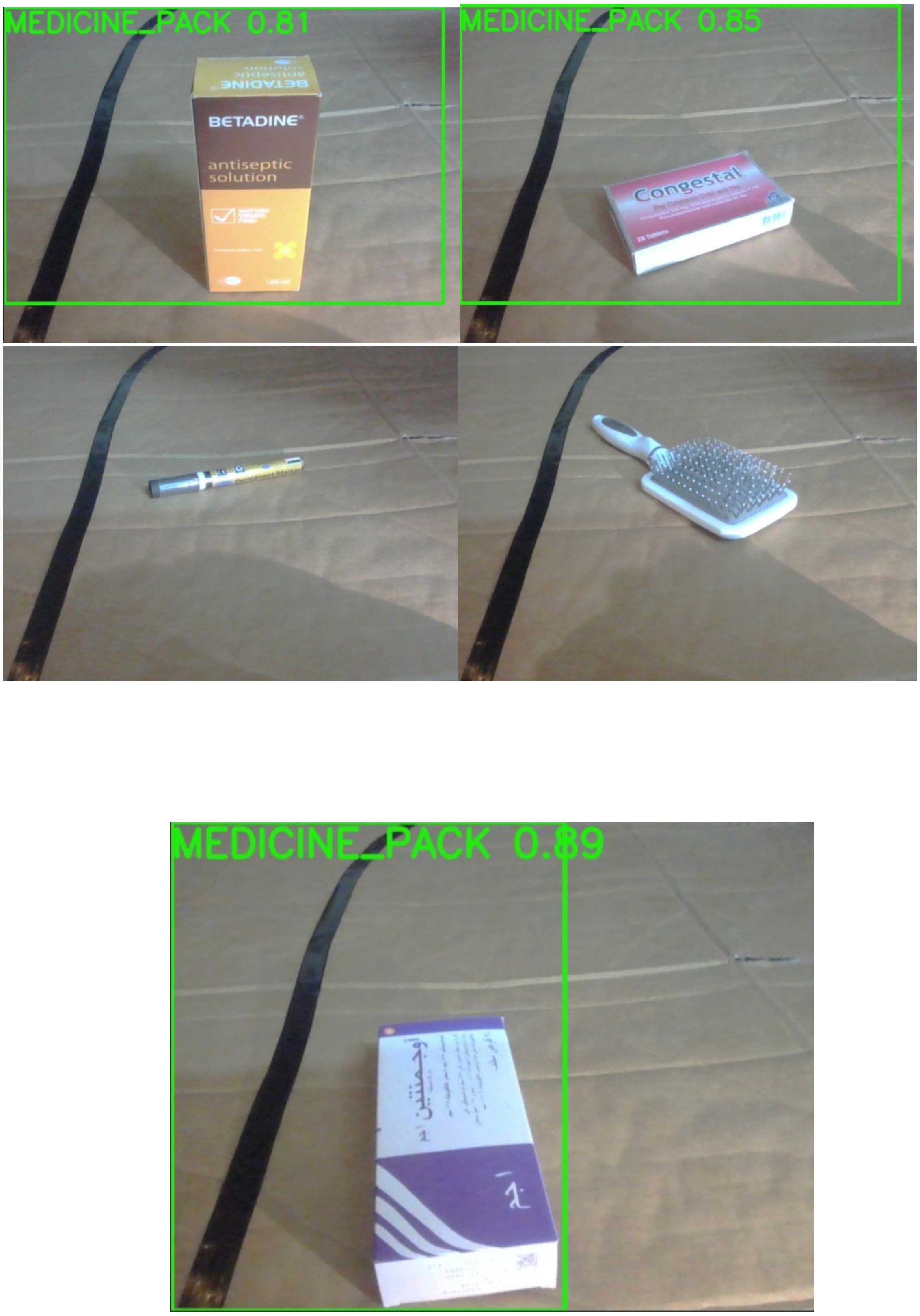


Figure 4.20

- **Mobile Application**
  - The mobile application for this project serves as an essential interface for controlling the Robotic Mobile Arm and its integrated car. It enables users to connect the robotic arm and car to a Wi-Fi network by entering the IP address, ensuring reliable and remote operation within the healthcare environment. Users can manually control the movements of the robotic arm and car in real-time.

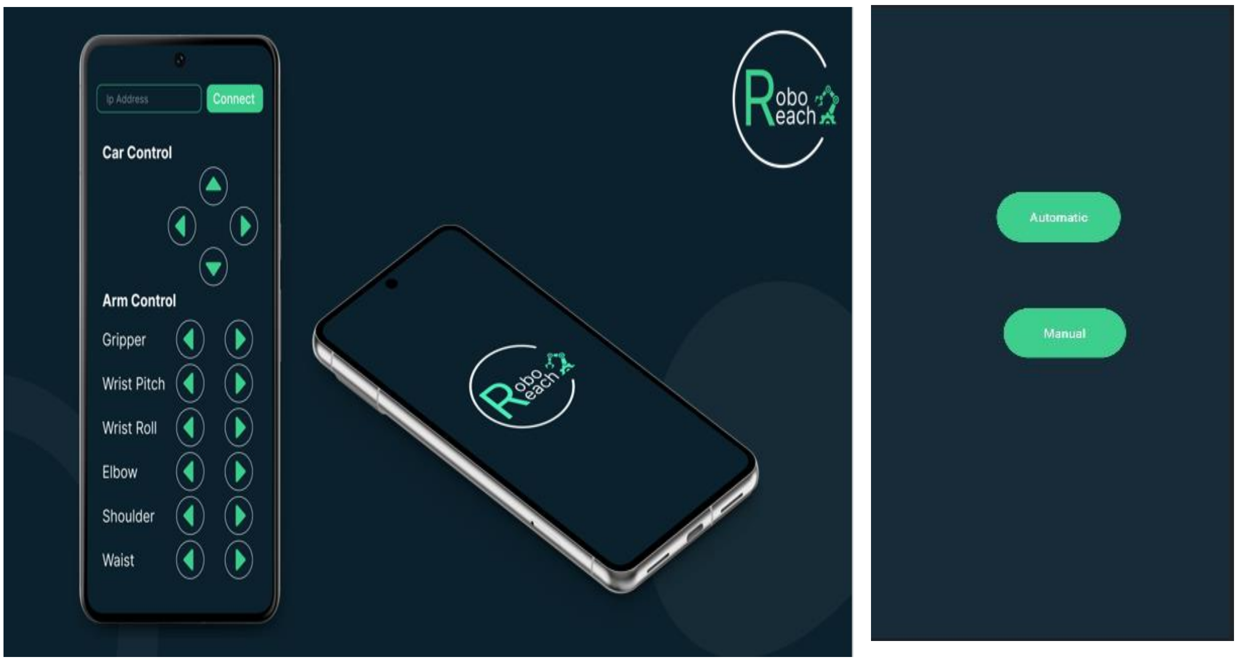


Figure 4.21

- **Arm & Car Model (Results)**

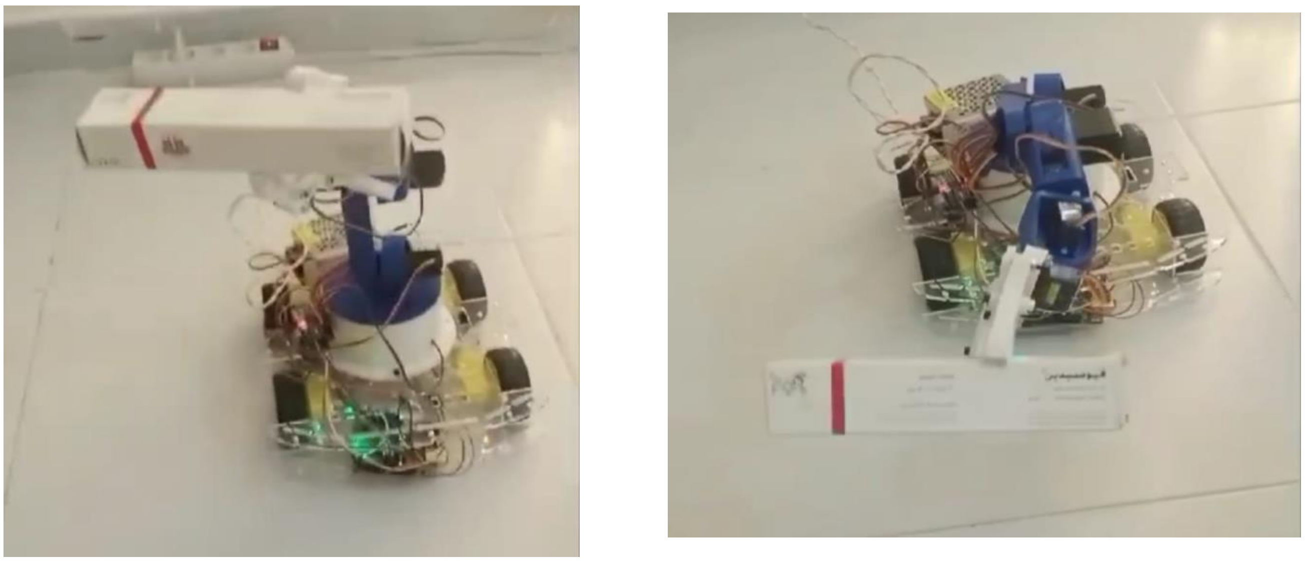


Figure 4.22

- **Car Line Detection:**
  - Car line detection using IR sensors enables autonomous navigation of the car by detecting and following predefined paths. Infrared (IR) sensors are strategically placed on the car to detect the contrast between the dark line and the surrounding surface. As the car moves, the IR sensors continuously scan the surface beneath it. When the sensors detect the line, they send signals to the control system, which adjusts the car's steering to keep it aligned with the path. This method ensures precise and reliable navigation, making it ideal for applications in healthcare, manufacturing, and other environments where automated guidance is necessary.

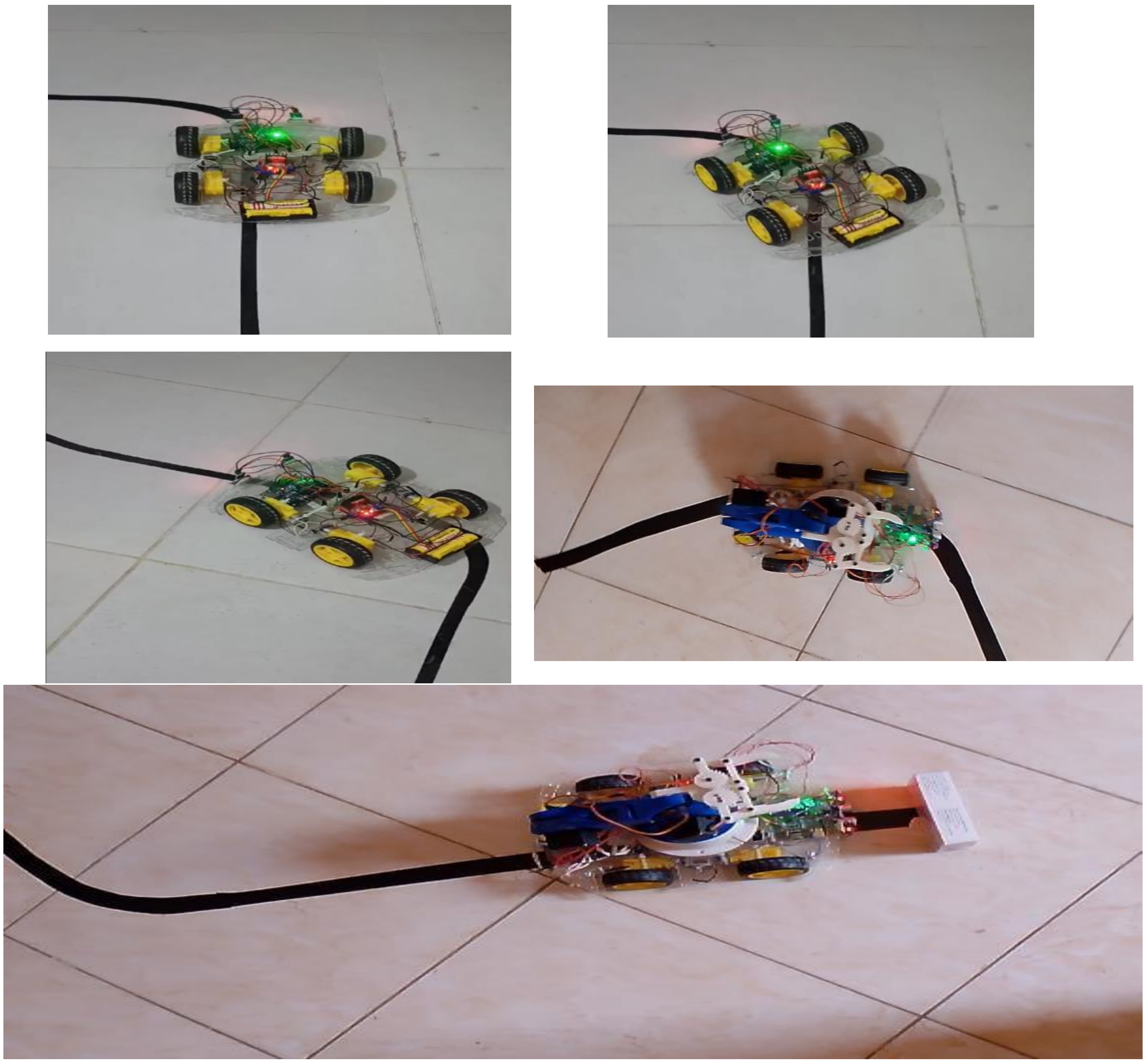


Figure 4.23

Chapter Five

Run the Application

The following steps describe in detail how to control the model manually:

- **Start the application**
- **Connect to Wi-Fi:**
  - Enter the IP address of the of the network to initiate connection between the application and raspberry pi.
  - Press connect button.
- **Control the Car:**
  - Move the car in the direction of what you want to pick.
- **Control the Arm:**
  - The model has 5 motors each motor controls the movement of a part, (gripper, wrist pitch, wrist roll, elbow, shoulder, and waist).
  - Pick the desired object by the gripper.
  - Place in its correct place.

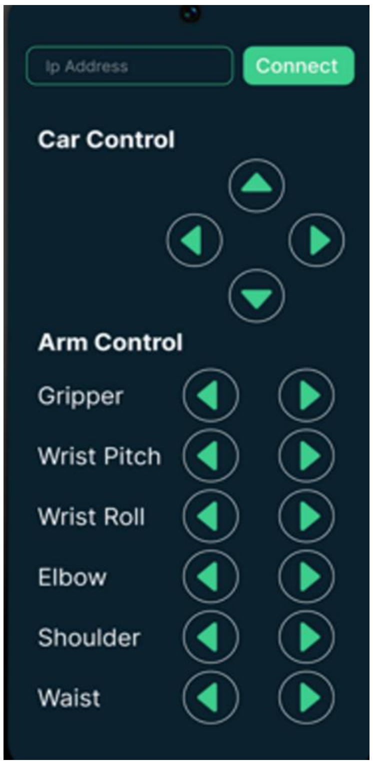


Figure 5.1

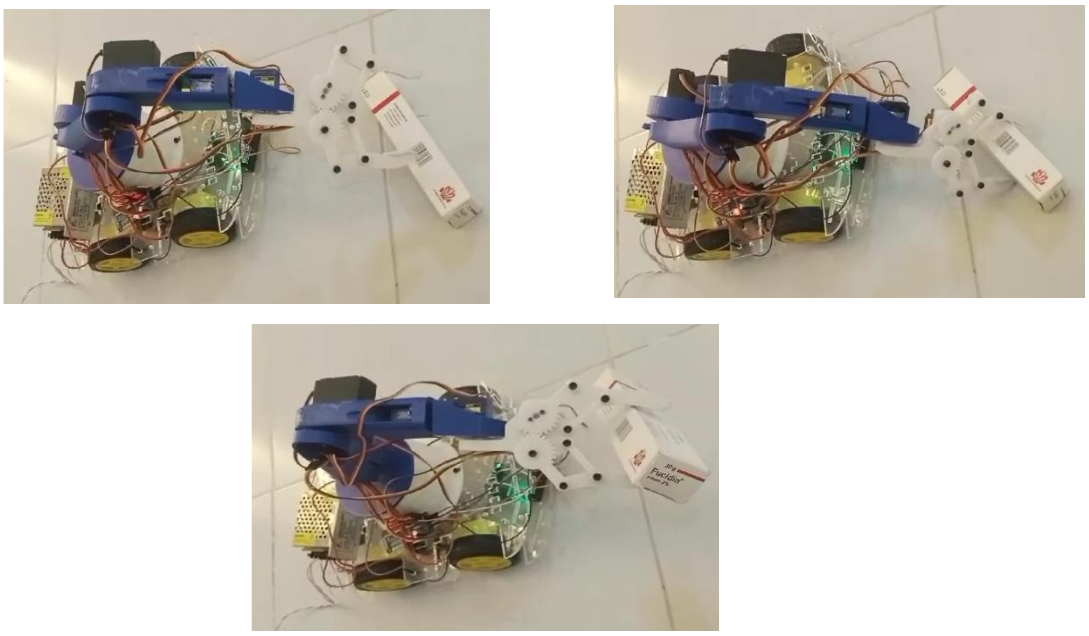


Figure 5.2

The car moves automatically if there is no serial communication:

- IR sensors detect the contrast between the dark line and the surrounding surface. The car moves on the dark line.

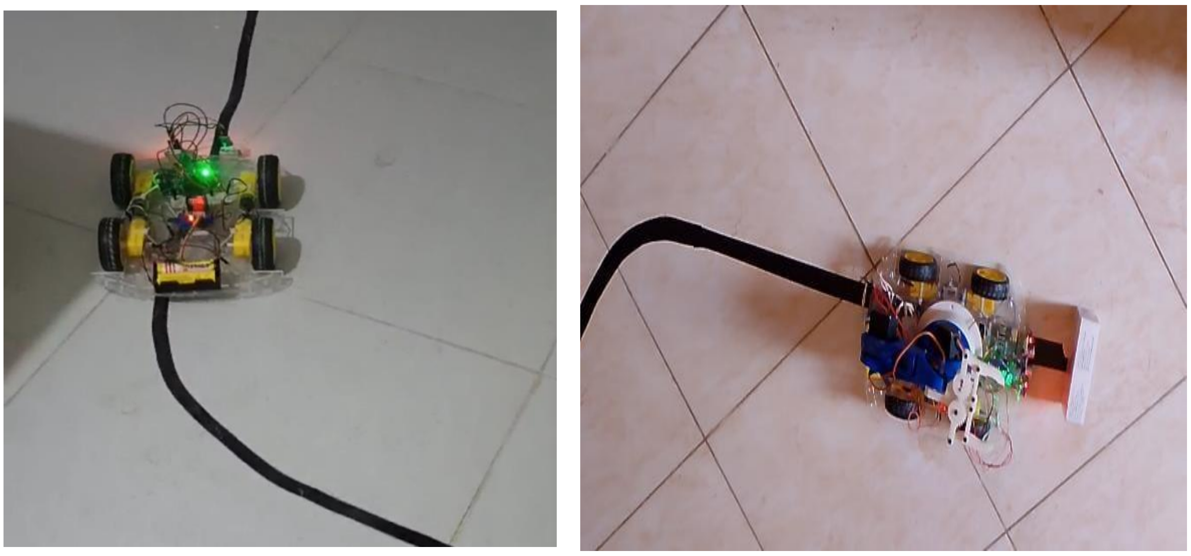


Figure 5.3

Object detection:

- Once the arm is connected to the Raspberry Pi, the attached camera begins capturing images. If the camera detects a medicine box, the system sends a signal via serial communication to stop the arm's movement. Simultaneously, the buzzer is activated to alert the user of the detection. This feature ensures precise handling of medications and prevents errors, enhancing the overall efficiency and safety of the robotic system in medical applications.

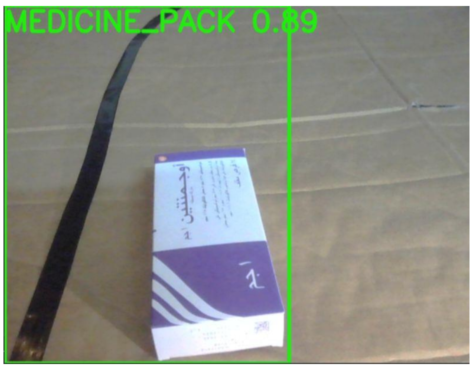


Figure 5.4

Chapter Six

6.1 Conclusion

This project successfully developed a cost-effective robotic arm for pick-and-place tasks in the medical field. By automating repetitive tasks, the robotic arm enhances precision and reduces errors, addressing common challenges faced by traditional manual methods. The system's three-layer architecture—comprising the Application Layer, Communication Layer, and Physical Layer—ensures efficient control and reliable performance. The integration of advanced technologies allows medical staff to focus on more critical tasks, ultimately improving patient care. The positive results demonstrate the potential of robotic assistance in healthcare, paving the way for further innovations in medical robotics.

6.2 Future Work

To further enhance the capabilities and impact of the robotic arm in medical settings, several areas of future work are recommended:

1. **Advanced Sensor Integration:** Incorporate additional sensors, such as force and tactile sensors, to improve the robotic arm's ability to handle even more delicate and complex tasks with greater precision and safety.
2. **AI and Machine Learning:** Implement artificial intelligence and machine learning algorithms to enable the robotic arm to learn from its tasks and improve its performance over time. This can lead to better adaptability and efficiency in various medical procedures.
3. **Expanded Functionality:** Extend the robotic arm's capabilities beyond pick-and-place tasks to include more complex functions, such as suturing, tissue manipulation, and assisting in minimally invasive surgeries.

4. **Enhanced User Interface:** Develop a more intuitive and interactive user interface for the mobile application, potentially incorporating voice commands and augmented reality to make the system more accessible and easier to use for medical staff.
5. **Teleoperation and Remote Control:** Enable remote operation of the robotic arm, allowing medical professionals to perform tasks from a distance. This can be particularly useful in telemedicine and during infectious disease outbreaks where direct contact should be minimized.
6. **Clinical Trials and Validation:** Conduct extensive clinical trials to validate the effectiveness and reliability of the robotic arm in real-world medical settings. Gather feedback from healthcare professionals to further refine and improve the system.
7. **Cost Reduction and Scalability:** Explore ways to reduce the cost of the robotic arm and make it more scalable, ensuring that it can be widely adopted in various healthcare facilities, including those in resource-constrained environments.
8. **Integration with Other Medical Systems:** Ensure seamless integration with existing medical equipment and electronic health records (EHR) systems to provide a comprehensive and cohesive workflow in the healthcare setting.

Reference

· **Wireless Robotic Arm**

Mohd Ashiq Kamaril Yusoff, Reza Ezuan Samin (2012)

· **Internet Controlled Robotic Arm**

Wan Muhamad Hanif Wan Kadir, Reza Ezuan Samin& Babul Salam Kader Ibrahim
25 August 2012

· **Evolution of robotic arms**

Michael E. Moran
May 2007

· **User Interface and Integration in Medical Robotics**:

- Lum, M. J. H., et al. (2009). "Teleoperation in Surgical Robotics – Network Latency Effects on Surgical Performance". *Robotics and Automation Systems*, 57(4), 419-428. DOI: 10.1016/j.robot.2008.11.005
- Stoianovici, D., et al. (1997). "A Modular Surgical Robotic System for Image Guided Percutaneous Procedures". *Lecture Notes in Computer Science*, 1205, 404-410. DOI:

· **Model Creation**:

- Redmon, J., & Farhadi, A. (2018). **YOLOv3: An Incremental Improvement**. arXiv preprint arXiv:1804.02767.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). **Deep Learning**. MIT Press.
- Sutton, R. S., & Barto, A. G. (2018). **Reinforcement Learning: An Introduction**. MIT Press.
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). **Deep learning**. Nature, 521(7553), 436-444.
- Gonzalez, R. C., & Woods, R. E. (2018). **Digital Image Processing**. Pearson.