بسم الله الرحمن الرحيم

AN-NAJAH NATIONAL UNIVERSITY

FACULTY OF ENGINEERING AND INFORMATION TECHNOLOGY

Computer Engineering Department

Hardware Graduation Project

Sorting System

students
Ahmad Saadeh
Ahmed Awwad

Advisors
Dr.Hikmat Darawsheh

Presented in partial fulfillment of the requirements for Bachelor's degree in Computer Engineering.

Jan,2025

Acknowledgments

We would like to express our heartfelt gratitude to all those who provided support and guidance during this project. Special thanks go to our advisor Dr. Hikmat Darawsheh for his invaluable guidance. Also, for our families and friends for their encouragement.

Disclaimer

This report was written by the students Ahmad Saadeh and Ahmed Awwad at the Computer Engineering Department, Faculty of Engineering, An-Najah National University. It has not been altered or corrected, other than editorial corrections, as a result of assessment and it may contain language as well as content errors. The views expressed in it together with any outcomes and recommendations are solely those of the student(s). An-Najah National University accepts no responsibility or liability for the consequences of this report being used for a purpose other than the purpose for which it was commissioned.

Contents

List of Figures

Chapter 1

Introduction

The present world is full of scientific novelties in the form of high-tech technologies that have made humans' lives easy and resulted in increasing standards of living. The operations need automation with minimal human intervention. This project aims to develop and implement a computerized automated sorting line that will have advanced image processing with color recognition technology to sort and ensure exact accuracy and efficiency in the proper sorting of different products and categorization for the same.

1.1 Problem Statement

In delivery systems, manual sorting of products is very time-consuming, cumbersome, and error-prone. This may further cause delays, raise operation costs, and result in customer dissatisfaction. An automated system can considerably reduce this problem by providing faster and more accurate alternatives.

1.2 Objectives

The objectives of this project are:

- Design a capable automatic sorting system to identify and sort items based on color-coded labels.
- Design and develop a color-detecting system using a camera based on enhanced image-processing techniques.

- To optimize the efficiency of the sorting process, reduce errors, and minimize human interference.
- A user-interface website for monitoring the sorting process in real-time, including analytics for optimization.
- Automatic item filling: Mechanism for filling items that need to be precisely placed into the sorted boxes.
- Conveyor-based transport mechanism: for moving the boxes that have been filled. efficiently to the unloading area.

1.3 Scope of the Project

The project involves the design of a working prototype of an automated sorting system for small- to medium-scale delivery businesses. This system is targeted to:

- Recognize and sort items using color-coded labels.
- Automatically send items to their respective destinations according to their label information.
- Monitor and visualize in real-time using a user-friendly web-based application.
- Fill sorted items into boxes efficiently and transport them smoothly to the unloading area.

1.4 Significance of the Project

By adopting this automated sorting system, delivery companies can experience several key benefits, including:

- Lowering operational costs by reducing the need for manual labor in sorting tasks.
- Improve delivery times by minimizing sorting errors and delays.

- Enhancing delivery speed by minimizing errors and delays in the sorting process.
- Enabling businesses to scale their operations effectively to keep up with increasing demand.
- Boosting customer satisfaction through more reliable, accurate, and timely deliveries. This system offers a practical solution to help businesses stay competitive and meet evolving logistical challenges.

Chapter 2

Literature Review

This project focuses on enhancing the sorting systems and algorithms. Many sorting systems and algorithms have been used in factories, projects, and research. In 2010 an academic paper was published for a machine that sorts lemons according to their color or shape. Color and size are the most important features for accurately classifying and sorting citrus. Because of the ever-growing need to supply high-quality food products quickly, automated grading of agricultural products is getting special priority among many farmer associations [4].

2.1 Overview of Sorting Systems

Sorting problems and sorting system machines are not a new problem. A lot of research confirms that this topic is important. Production lines in many sectors, especially the food sector, depend on sorting things according to various factors. A study on color image processing based intelligent fruit sorting system made in 2004 said that Sorting is one of the important tasks in the production line and has an appreciable effect on the homogeneity of products. The use of vision to increase productivity in automated sorting systems is of interest to a large number of researchers[3].

2.2 Color Detection in Sorting Systems

Color detection is one of the main approaches used in the automated sorting of objects. In this method, the features of an object are examined to sort them based on specific attributes such as color-coded labels. It has been shown that using image processing libraries like OpenCV or even hardware accelerators such as DepthAI increases the speed and accuracy of color recognition.

DepthAI usage in color detection in sorting systems has grown because it possesses advanced features for real-time processing and depth perception. DepthAI improves the accuracy of color classification in the sorting of objects because it integrates AI-powered computer vision with neural networks. This system can operate in different light environments, increasing sorting task reliability. Also, Because DepthAI can process visual data at very high speeds it can easily be integrated into conveyor belt systems to achieve fast and accurate color-based sorting in practical real-world situations.

Nowadays various automation techniques are being adopted and researched for the increase in productivity, better accuracy, eliminating human errors, and safety. Machine Vision is one such advancement in automatic systems. Machine vision performs the tasks that are equivalent to human vision. It helps to automate the systems where there are limitations for the human vision like detecting various shades of colors or determining precise dimensions thus permitting human employees to serve in more appropriate positions[2].

2.3 Industrial Automation Techniques in Sorting Systems

Alaameri et al. (2024) developed a computer vision system for color sorting of objects using PLC (Programmable Logic Controller) systems with TIA Portal and Factory I/O software for automation and simulation. Color sensors were used to identify the characteristics of objects on the conveyor belt, and the PLC controlled the sorting unit [1]. This technique produced a robust solution for industrial automation by integrating SCADA (Supervisory Control and Data Acquisition) with HMI (Human-Machine Interface), thus facilitating monitoring and control of the system in real-time. While the system proved reliable and efficient in sorting materials bound by the predetermined set of parameters, its reliance on PLC hardware makes it expensive and not very optimized for small to medium-scale applications. Although this research serves as a very good reference in the use of industrial tools for sorting automation, it reveals the gap for cheaper and more flexible options, as well as the innovative alternatives like image processing and machine vision that this project offers.

2.4 Gaps in the Literature

Although great strides have been made toward the creation of automated sorting systems, several challenges still exist:

- A large proportion of existing systems are not scalable, and thus are not appropriate for small to medium-sized enterprises.
- The addition of real-time monitoring and control functions into the sorting systems is still at a very early stage.
- Most of the research concentrates on the technical side of the sorting systems but very little is done about how these systems are incorporated into the logistics practice.

A review of the literature draws attention to the changing sorting systems and the technologies associated with this change. Nevertheless, there is a gap for a more developed and cheaper system for small and medium delivery companies. This project is designed to address the gaps by offering a color-based sorting system with web real-time monitoring and control applications.

Chapter 3

Methodology

A smart sorting system is an automation system for sorting products based on their color or any other image processing purpose. This system is a combination of software and hardware to get the items sorted and transport the sorted items to their destination. Displaying all data of sorted items on a web page.

3.1 System Architecture

The smart sorting system architecture involves integrating a hardware and software system that enables effortless and effective sorting of products.

- **Hardware:** The DepthAI camera for color detection, the conveyor belt for moving the products, and actuators for the automated diversion of the products to the intended destination make up the physical system. The camera captures images of color-coded stickers placed on the products which are transported on the conveyor belt. The products are sorted by the Raspberry Pi 5 after the camera has captured the sticker images and sent signals to the Arduino Mega to move a servo motor to handle the item knowing its position on the belt according to the interrupt coming from the LDR laser sensors. Raspberry serves as the main control unit, processes the data from the cameras, and commands the actuators to sort the products.

- **Software:** The software components of the system include the image processing algorithms that are contained in the OpenCV, which segments the images captured by the DepthAI camera and extracts the colors of the stickers. A Firebase database enables the capturing of real-time sorted items and supports monitoring and analysis. Moreover, a web application based on a Firebase database is used to display all data of the sorted things in the system.

These components work in unison and form a simple workflow whereby the camera captures the image, the Raspberry Pi interprets the color, and the actuators perform the sorting process according to the specified instructions.

3.2 How it Works

1. The system is powered by a PC power supply that gives power to all components.

2. After pressing the power on the button a push-button is used to toggle the sleep state of the system which is on sleep mode initially.

3. By pressing the push button the code started by turning the DepthAI Camera on and then sending a signal to the Arduino to start the DC motor that moves the conveyor belt.

Figure 3.1: Power Button and Push Button to Turn the System ON and OFF

4. When the conveyor belt starts moving the system is ready to sort items.

5. The camera starts processing objects and adds them to 3 queues to be handled later. Also, the Python code adds them to the Firebase Database.
6. When an LDR laser sensor placed next to a servo motor arm reads a signal it makes an interrupt on the Raspberry.
7. The Python code on the Raspberry handles the interrupt by sending a signal to the Arduino Mega to move the servo if the object should be handed by this arm next to the IR that made this interrupt.
8. If the object is pushed by any arm in the belt it will cross another LDR laser sensor that will count it and add it to the Firebase database.
9. When any box reaches its limit of items it will send a signal to Arduino.
10. The Arduino will immediately stop the belt and start moving the boxes to free them up and return them.
11. When the boxes reach the belt again the belt will start moving again to continue sorting the items.

3.3 Hardware Design

The design of hardware concentrates on the integration of components that guarantee accurate item handling, efficient color detection, and sorting. The major components utilized in the project are as follows:

- **Conveyor Belt:** A motor-driven conveyor belt aids in the movement of items along the sorting path. It is powered by a DC motor and controlled by an Arduino Mega so that the operation is continuous and steady. (see figure 3.2)

Figure 3.2: Conveyor Belt That Objects Placed On to Be Sorted

- **Camera System:** A Luxonis OAK-D camera is fastened onto the top of the conveyor to grab pictures of the items. Its main task is to recognize stickers that are used for determining the sorting of products. Images of items along with color codes will be processed by a Raspberry Pi 5 attached to the camera.

Figure 3.3: Luxonis OAK-D Camera

- **LDR Laser Sensors:** These sensors are mounted on the conveyor belt at intervals to sense the outgoing items and send interrupts for sorting to take place accurately. This also aids in the tracking and positioning of the items.

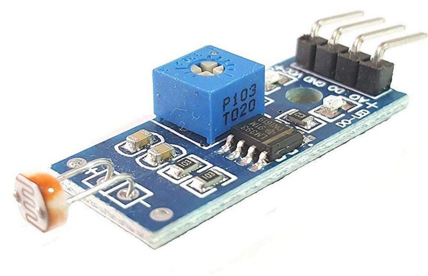

Figure 3.4: LDR Photo-resistor Light Detection Sensor Module

- **Actuators:** Servo motors working as actuators are used for the diversion of outgoing items to specific bins according to the estimated colors. The Arduino Mega sends signals generated in response to input from the Raspberry Pi to the servo motors.

  Also, 2 DC motors have been driven using the Arduino signal through the BTS7960 motor driver (see Figure 3.5).

Figure 3.5: BTS7960 High Current 43A H-Bridge Motor Driver

  Those DC motors are responsible for the conveyor belt movement and box transportation.

- **Raspberry Pi 5:** The Raspberry Pi 5 component is a single-board computer that serves as the main processing unit for the system. It receives images from the DepthAI camera, processes them, interfaces with the Arduino Mega, and issues commands to the sorting actuators. Also, it constantly updates the Firebase database with sorting data.

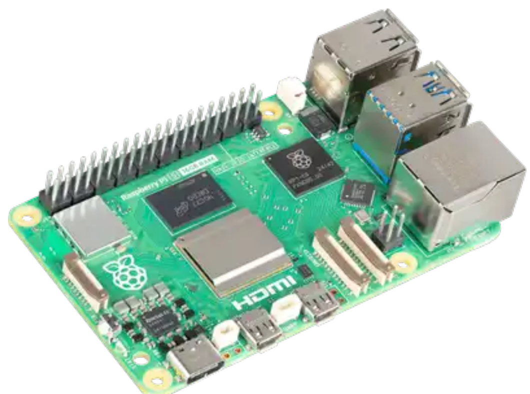

Figure 3.6: Caption

- **Arduino Mega:** This microcontroller is responsible for hardware-level control such as turning on and off the motor of the conveyor belt, moving the servo motors, and listening to the Raspberry Pi commands. Also, it's responsible for the transportation of the system to turn on and off the motor that moves the boxes according to limit switch signals and turning off the conveyor belt motor while transportation.

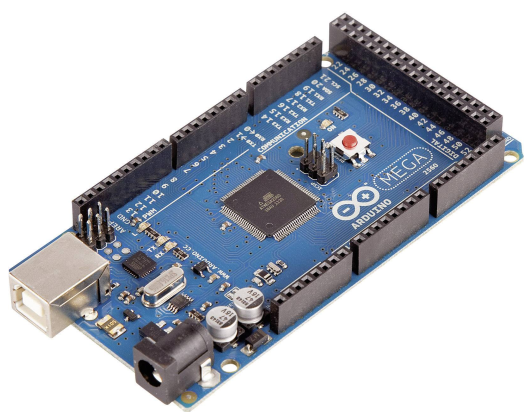

Figure 3.7: Arduino Mega

- **Limit Switches:** Limit switches are connected directly to the Arduino and its function is to tell the Arduino whether the boxes reach the destination to take them back or reach the belt to turn the belt on again.

Figure 3.8: Limit Switch

- **Power Supply:** A PC power supply is used to ensure that all the components can be reliably powered, which in turn enables the motors, sensors, and micro-controllers to function. It converts the 220 AC power to 12, 5, and 3.3 Volt DC to drive the components.

Figure 3.9: Caption

By combining these hardware components, it is possible to design an advanced sorting system that contains all the functions needed for item transportation, detection, processing, and destination-based sorting. All of them give a guarantee at the very least system-level accuracy of the sorting algorithm.

3.4 Software Development

The development of the smart sorting system is very strategic and methodical. This will allow the software to be easy to strengthen in the future if necessary. The core of the sorting system is to combine sensor data acquisition, color identification, actuator control, and communication with the Firebase cloud databases so that items are sorted automatically by color.

1. **Sensor Integration and Data Handling:**There are 6 infrared IR sensors connected to the Raspberry Pi through gpio pins. Those sensors' function is to detect if there is an item crossing and send an interrupt to the Raspberry Pi. To make sure that all interrupts have been read directly, every interrupt is handled inside a possess alone. That means that there are 6 alive possess that are waiting for the interruption from the IR sensors.

2. **Color Detection Algorithm:**The camera Function is also in a possess itself to make sure that every object crossing under the camera will be detected. The camera detected 3 colors at this time (Red, Green, and Blue) and did actions according to them.

   What happens when the camera detects an item is to give it a character. R for red, M for green, and L for blue. Those characters refer to right middle left respectively.

   The characters are then processed to be a queued system consisting of three queues of boolean. Every queue is filled with a boolean true or false and handled using one of the IR processes.

3. **IR Process Handling:**Every IR gives an interrupt to its process if an item is crossing it. Then the process just enqueues one item from its queue. if the item is true and the IR for example is IR1, the process sends R to the Arduino. If it's false or the queue is empty the process

does nothing. Same for IR2 and sending M to the Arduino and IR3 and sending L to the Arduino.

4. **Controlling Actuators:**When a serial input comes to the Arduino it checks the character. If it is R, M, or L a signal to a servo motor will be sent to move and move back making it pushing the item into its box. The Arduino doesn't know if the object is the one that has to be pushed. It's just doing what it has to do according to the serial input.

   For DC motors it's completely different. The conveyor belt motor is OFF by default. As mentioned in the 3.2 section, when the push button is pressed the belt starts moving. This happened because the Raspberry sends serial communication to the character S to the Arduino making it directly start the DC motor.

   For the other DC motor that is responsible for moving the boxes, Also the Arduino is the controller for it. When the Arduino gets the G command, it stops the belt and starts moving the motor forward waiting for the limit switch signal, then backward until it returns to the second limit switch the motor will be OFF and the belt will move again.

5. **Using Firebase for Data Storage:**Firebase is mainly used to store the data about sorted items in a real-time database. Most of the processes reach and modify the Firebase database. The camera sets the count of the items it reads into the firebase. Then when the servo handles an item and the item crosses the IR sensor that counts the number in the boxes, the process of this IR adds one to the count of the boxes and updates it in the firebase. In the main process, the system checks the item count. Every box has a capacity count and when reaching this count the main process sends G to the Arduino and frees the box count on Firebase.

6. **Real-Time Synchronization:** Every code reset clears the data in the database to zero. So the system always starts from zero. Also to make life easy the database is synchronized in real-time and the data and current status of the process can be seen from anywhere.

7. **Using Process for Efficiency:**The system before possess was too slow and handling it was impossible. This leads to the processing technique where everything is running independently and reaching the current

states across global queues or variables. That is the reason behind using the Raspberry Pi for IR interrupts where it's possible to divide the system into a multi-process program. This was impossible if the interrupts were connected with the Arduino.

8. **Web Application and Data Display:**The web Application is only for data tracking. All data is displayed in a real-time. every update on the sorting process, new item detected, new item added to the box, or boxes free up process applied, updates the Firebase. The web application listens to the changes and updates directly (Real-time Database Listener).

The software was such a challenge if the Arduino is the processor for everything and the Raspberry only for image processing, this leads to the idea that it is very important to multi-process the system to make sure everything is working together in a parallel automated system. Also, to make the tracking of the system easier, the web page[5] has an instance update for the processed object in the system.

Figure 3.10: Website for data display and tracking system

3.5 Testing and Evaluation

To make sure that the system is working efficiently and to track all the processes some tests have been made:

- **Color Detection Accuracy:** The system was tested using colored boxes (Red, Green, and Blue). The intensity around the camera was a challenge. To solve this problem a cover was added around the camera making the intensity almost constant.
- **Sorting Speed:** The sorting speed relies on the efficiency of the IR sensors and its time to detect and send interrupts. As the system is multi-process then the speed is not a challenge. The challenge is to synchronize the time between detecting the item by IR and sending the serial to the Arduino.
- **Web Interface Usability:** While making all the tests the website was always open to make sure that the tracking system was working properly. Every time a mistake is found in the tracking system leads to a problem in the IR reading or color detection, not the website itself.

Those tests are used to improve the system and make sure that everything is working as it has to.

Chapter 4

Results and Analysis

This section will discuss the outcome of the system from design to development to testing. Also, the system's key features, strengths, and challenges.

4.1 System Functionality Overview

The system is designed and implemented to classify objects based on image processing techniques. The image processing technique used in this system detects the item and classifies it according to the color in an automated workflow. The key functionalities are:

- **Color Detection:** Using Luxonis OAK-D camera to detect items. Python code processes the colors of every item.
- **Automated Sorting:** The Python code on the Raspberry Pi sends signals to the Arduino at the right time to move the servo motors, stop the belt, or free up the boxes.
- **Real-Time Data Handling:** While the system processes the items everything is updated on Firebase in real-time making the system easily tracked.

4.2 Observed Strengths

The system's strengths were noticed during the tests and could be summarized into several things:

1. **Integration of Components:** Combining Raspberry Pi with Arduino Mega and connecting them with a real-time database on Firebase gives the system the ability to handle complex tasks and do them in parallel without affecting other tasks.

2. **Parallel Processing:** The strength of using Raspberry Pi gives the system the ability to multi-task and execute code lines in parallel. This was impossible using Arduino only or other systems that don't provide multi-tasks.

3. **Real-Time Synchronization:** Using Firebase's real-time database the system keeps track of everything happening and knows when an error occurs directly. This makes it possible for the system to be monitored.

4. **Scalability:** The system modularity makes it easy to scale, debug, and maintain. For a sample project like this system, there are only 3 sorting lines. But if there are more than three objects the system could be scaled easily by adding another process for handling the fourth item or the n^{th} item.

4.3 Challenges and Limitations

The development process was not easy at all. Some challenges make the system implementation harder which are:

1. **Environmental Sensitivity:** While setting the colors and trying to get them correct. The light intensity around the camera was causing some problems. This leads to covering the camera with a curtain to make the lighting somewhat constant.

2. **Synchronization Complexity:** Any additional delay in the camera processing, IR interrupt, DC motors movements, servo motors movements, serial communication, or Firebase communication, if it's not as expected, makes the system sorting errors and gives wrong outputs. This challenge makes every delay time to be counted.

3. **Hardware Dependency:** Having a lot of hardware components makes the system easy to fail and the classifying process sorts items wrong if any component fails or does not work as expected.

4.4 Expected Performance

According to the development and testing process, the system is expected to deliver:

- **Accurate Sorting:** All items should be sorted correctly and transported to their destination.
- **Real-Time Monitoring:** The Firebase with the website deployed on it should be displaying data correctly and at the same time an update occurs in the system.
- **Efficiency:** The system processes should work in parallel without any delay due to using multi-process programming on the Raspberry Pi.

4.5 Future Considerations

In the development of the system, these suggestions could be useful:

- **Enhanced Camera Functionality:** The usage of advanced techniques may facilitate the introduction of additional classification categories or improved accuracy.
- **Redundant Sensors:** The system's tolerance to hardware malfunctions could be commensurately improved through the addition of backup sensors.
- **Modular Scalability:** Making the system ready to add more sensors and servos by adding ready-to-plug interfaces.
- **Improved User Interface:** Making the web interface more useful by adding features to check sensors working and system errors.
- **Adding Arm for special propose:** Some objects will be affected by the pushing system arm. Implementing an arm at the end of the line that handles those specials will be a good choice for a general item classification.

Although the system has no quantitative measurement, it is obvious that the automated sorting system is a strong solution for the classification problem.

For time, accuracy, and cost. In this system, all needed is a worker to put items on the line and the sorting will be applied more accurately, and tracking errors in sorting will be easy according to the web-based tracking system. Future improvements will focus more on the availability, the speed of the system, and the modularity of the system.

Chapter 5

Discussion

The system in its development takes more effort in making decisions about what to use, why, and how. The system has many challenges in its development cycle. It was not easy to get out with a fully automated system considering the complexity of the system.

5.1 Design Considerations

The efficiency, automation, scalability, and reliability talked the system development to many decisions:

- **Modular Architecture:** The system consists of tow controllers. Arduino is used for hardware-level handling. It handles all motors in addition to the limit switches While Raspberry Pi focuses more on Firebase data processing and handling multi-process.

- **Real-Time Monitoring:** A web application that has the items processed and counting for all boxes in real-time makes the system tracking easier and can be debugged. Tracking the system is one of the most important things in such a system.

- **Multi-Process Programming:** Using multi-process in Raspberry Pi gives the system the ability to make fast decisions for object classification. The camera reads in parallel with processing other items in the line. The Firebase processing also does not affect the system performance.

5.2 Reflections on Observed Outcomes

The primary objective of the system has been achieved with some positive results:

1. **Automation Efficiency:** The system is working without human effort except for putting an item on the line. The item takes its way from the belt to the sorted area without human intervention.

2. **Real-Time Responsiveness:** High speed with real-time multi-processing makes the system able to do more than one task at a time.

3. **Usability and Monitoring:** Tracking the system through a web-based application in real-time.

5.3 Challenges and Lessons Learned

Despite the system's strengths, several challenges were encountered:

- **Environmental Factors:** It was challenging to make a color-based system classification while the colors differ according to the room lighter. Making the system take colors in ranges solves this problem with a core above the camera detection area.

- **Hardware Reliability:** The dependency on multiple hardware components makes it risky. Any failure in the system's hardware affects the system.

- **Synchronization Complexity:** The calculation of the delays needed for everything was not easy so the software-to-hardware implementation needs to be sure that everything takes its time without exceeding a limit.

- **Scalability Limitations:** While the system is scalable for a limited number of modules. Adding more than the limit will lead the system to lag and sort inefficiently. This will lead to scaling, or upgrading the controllers.

5.4 Broader Implications

The innovative smart sorting system is a true game changer that enables automated sorting systems for logistics, manufacturing, and even recycling. Using advanced technologies such as real-time databases or multi-process programming, the system enables sophisticated automation where hardware talks to the software and vice versa.

Yet, the main success lies beyond technological advancements, and this project focuses on and strongly supports:

- **Scalability and Flexibility:** Automated systems should be created with the potential for future growth so that they can be adjusted depending on the expansion of the client's organizational functions.
- **Resilience to Failures:** One of the most important requirements for industrial systems is the development of errors that respond even when a unit fails.
- **User Accessibility:** The inclusion of monitoring dashboards in the system upon which the user interfaces of the system is to real-time, the usability of other systems can greatly improve.

5.5 Future Directions

Considering the lessons learnt and the outcomes that are reported, the following directions are suggested for future work:

• Enhanced Object Classification: Expanding the system to detect and classify certain additional object monikers such as size, shape, or texture would broaden its applicability.

• Improved Fault Tolerance: By adopting redundancy on critical components such as sensors and actuators, the system's accuracy can be boosted.

• Scalable Design: Modifying the system to add extra sorting categories or bins may make the system suitable for different kinds of industries.

• Advanced Monitoring: Adding a bit of predictive analytics to the web interface could aid tremendously in evaluating system performance and other bottlenecks.

The development of the smart sorting system suggests the possibility of merging these elements with practical application in automation of sorts.

These objectives were said to be fully satisfied by the system, however the challenges confronted offer grounds for progress of the system. Proposing these limitations and persuing these avenues after bridge these possibilities might style the system into a better improved moderated solution for automated sorting tasks.

Chapter 6

Conclusions and Recommendations

6.1 Conclusions

With this project, an automatic sorting technique has been created which allows for the combination of software with hardware. It demonstrates an effective sorting mechanism. Because the system is able to sort and classify objects according to their respective colors as well as monitor the progress with a web interface while multi-tasking, it is a huge leap toward industrial automation.

The critical conclusions derived from this project include the following considerations:

- **Automation Feasibility:** Using Raspberry Pi alongside Arduino microcontroller was useful for accomplishing simultaneous real-time data management and controlling the hardware.

- **Real-Time Monitoring:** There was incorporation of Firebase, which enables systematic operations to be carried out in real time, which was necessary for monitoring, correcting and system transparency.

- **Process Efficiency:** By employing a multi-process approach and design, the system was allowed to perform numerous tasks at the same time without much delay, thereby increasing efficiency.

- **Scalability Potential:** Even though the present system works and is efficient, there is great opportunity for enhancement, including adding more sorting parameters or increasing the system's scale of operation.

  The results of this project highlight the effectiveness of combining modern technology with strategic design to address practical challenges in automated sorting systems.

6.2 Recommendations

There are few positives and few limitations concerning the system developed through display techniques and thorough testing. This feedback, however, contemplates the progress as a whole and measures the efficiency of the design techniques.

6.2.1 System Improvements

- **Improved Object Recognition:** Add on features that unit might recognize such as dimensions and contour and pattern of an object. This will allow the system to be used in a wide range of applications.
- **Controllable Lighting:** Incorporate lights or filters that can adjust independently so that the surrounding light has less of an effect on the performance of the color detection system.
- **Redundant Components Inclusion:** Redundant components such as sensors and actuators can be added to increase tolerance to faults, enhance reliability and decrease outage time caused by hardware failure.
- **Advanced Analytic Inclusion:** Greater advanced analytics features should be added for performance monitoring to discover trends, bottlenecks, and maintenance prediction.

6.2.2 Future Research Directions

- **Machine Learning:** Investigate methods for integrating object detection machine learning algorithms into the existing features of the system for better precision.

- **Scalability:** Look at ways the system can be made more powerful by developing new algorithms and purchasing additional processing hardware for increased productivity.

6.2.3 Deployment on System

- **Documentation and User Training:** Develop user guides and conduct informative training sessions to enable effective functioning and upkeep of the system.
- **Testing for Environmental Conditions:** Ensure the system performs satisfactorily under various moisture, heat, and lighting conditions for tested robustness.
- **Compliance for Regulation:** Ensure the system has the necessary features and limitations to fit the various legal constraints in the industry for commercial use.

6.3 Final Thoughts

The smart sorting system is a big step toward improving the efficiency and automation of high-volume sorting tasks. The results indicate a successful implementation of the project through the proper interaction of hardware and software integration and a broad-based approach to modular and scalable system design.

Bibliography

[1] Karrar J Alaameri, Ali J Ramadhan, Ahmed Fatlawi, and Zainab S Idan. Design of a new sorting colors system based on plc, tia portal, and factory i/o programs. *Open Engineering*, 14(1):20220547, 2024.

[2] Vedang Chauhan, Saurin Sheth, Bhavesh R Hindocha, Rushabh Shah, Parth Dudhat, and Pratyush Jani. Design and development of a machine vision system for part color detection and sorting. In *Proceedings of Second International Conference on Signals, Systems & Automation (ICSSA)*, pages 90–93, 2011.

[3] Guo Feng and Cao Qixin. Study on color image processing based intelligent fruit sorting system. In *Fifth world congress on intelligent control and automation (IEEE Cat. No. 04EX788)*, volume 6, pages 4802–4805. IEEE, 2004.

[4] M Khojastehnazhand, M Omid, A Tabatabaeefar, et al. Development of a lemon sorting system based on color and size. *African Journal of Plant Science*, 4(4):122–127, 2010.

[5] Ahmad Saadeh and Ahmed Awwad. Smart sorting system. `https://smart-sorting-system.web.app/`, 2025. Accessed: 2025-01-26.

- URL: `https://docs.luxonis.com`.
- URL: `https://smart-sorting-system.web.app/`