Design And Fabrication of Color Sorting Machine Based on Computer Vision

Ibrahim Ahmed Alrushidy (1, *)
Yousef Abobakr Bahumid (1)
Ramzi Abdulsalam Bin Shujaa (1)
Ahmed Ehab Abdullah (1)
Ammar Ali Kurd (1)
Mohammed Saeed Moqbel Abdullah (1)

Received: 21/04/2025
Revised: 26/05/2025
Accepted: 27/05/2025

© 2025 University of Science and Technology, Aden, Yemen. This article can be distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

© 2025 جامعة العلوم والتكنولوجيا، المركز الرئيس عدن، اليمن. يمكن إعادة استخدام المادة المنشورة حسب رخصة مؤسسة المشاع الإبداعي شريطة الاستشهاد بالمؤلف والمجلة.

[1] University of Science and Technology – Yemen
*Corresponding Author's Email: a.kurd@ust.edu

Design And Fabrication of Color Sorting Machine Based on Computer Vision

Ibrahim Ahmed Alrushidy
University of Science and Technology – Yemen
alrushidy.ibrahim@gmail.com

Yousef Abobakr Bahumid
University of Science and Technology – Yemen
yousef.abobakrb@gmail.com

Ramzi Abdulsalam Bin Shujaa
University of Science and Technology – Yemen
,
ramzi2014.99@gmail.com

Ahmed Ehab Abdullah
University of Science and Technology – Yemen
ahmedehabm31@gmail.com

Ammar Ali Kurd
University of Science and Technology – Yemen
a.kurd@ust.edu

Mohammed Saeed Moqbel Abdullah
University of Science and Technology – Yemen
m.saeed@ust.edu

Abstract— Manual sorting of objects by color can be exhausting and time-consuming, resulting in high labor costs and the potential for errors. At the same time, achieving accurate automation of this sorting process poses a challenging engineering task. This study presents a real-time method utilizing image processing of video frames collected by a camera and processed by an onboard Raspberry Pi computer. Unlike systems relying on color sensors or AI, this work achieves comparable accuracy using cost-effective, fast color identification with different colors' shades. An algorithm was specifically developed using the OpenCV package for this objective. The sorted objects are then transferred by a robotic gripper mechanism to their respective bins. The mechanical side will have a robotic arm and conveyor belt. The proposed design will be proven to be an effective and economical solution based on its capacity to pass tests set to evaluate its performance, where it exhibited a 90% accuracy under ideal operating conditions. The strategy being proposed is ideal for industrial applications that require color-sorting machinery, such as the fruit packaging and food processing sectors.

Keywords— Color-sorting, image processing, robotic arm, real-time processing, object detection, mechatronics, conveyor belt, computer vision, Raspberry Pi, OpenCV."

Introduction

In industrial automation, image processing has become crucial for many applications. This integration has paved the way for creative solutions that boost efficiency and precision. One significant application of this technology is categorizing things based on their color. For instance, in industrial automation applications requiring color detection such as sorting objects by color, image processing techniques and algorithms can be utilized to achieve this purpose. This will effectively speed up the sorting process, reduce costs and minimize human errors [1]. Most industrial applications demand sorting in real time, which in turn leads to employing cameras and using sophisticated algorithms and sensors [1]. This will streamline production line conditions for diverse products and as more companies adopt automation, the demand for image processing will continue to grow.

Although the color detection problem appears to be the same in all applications, there are differences in this problem that arise from the specific needs of the production process itself, such as the speed and accuracy of detection. This results in a number of detection methods, including the use of artificial intelligence [3, 4, 12], simple color thresholding techniques [2, 5], or specialized color sensors [6–8].

Studies on color detection with the use of AI or color thresholding approaches involve microprocessor-based solutions such as the usage of Raspberry Pi, generally with image processing software libraries such as OpenCV [2, 9-13]. On the other hand, investigations using dedicated color sensors adopt microcontroller-based solutions such as Arduino boards [6-8].

Dedicated color sensors offer cost advantages; however, they are inadaptable and have a limited accuracy, which renders them inadequate for dynamic industrial environments. AI-based solutions are more accurate and highly adaptable but demand higher computational performance, which leads to an increase in cost and limiting speed if operated on modest hardware. In this work we propose an automated production line (robotic arm and conveyor belt) control system using a color thresholding method based on hue, saturation, and value (HSV) color space thresholding on a microprocessor-based solution employing Raspberry Pi and OpenCV. The suggested system aims to establish a balance between cost, accuracy, and adaptability. Studies on color detection using AI or color thresholding approaches involve microprocessor-based solutions such as Raspberry Pi, typically with image processing libraries like OpenCV [2, 9-13].

I. Methodology

This section presents the materials, methods, and overall flow of the system development. The study outlines the core hardware and software components employed in the system's development, including the Raspberry Pi 4 Single-Board Computer (SBC), Arduino Uno microcontroller development board, Arducam Camera Module, Python programming language, and OpenCV library. Each component plays a critical role in enabling the functionality and integration of the sorting process.

The design methodology is divided into three principal phases:

1. Embedded System Design: Focused on electronic control and software integration.

2. Mechanical Design: Pertaining to the physical assembly and actuation mechanisms.
3. System integration: In this final phase the work from the previous two phases is integrated and tested.

The sequential flow of system design and integration is illustrated in Figure 1, which provides a structured overview of the developmental workflow.

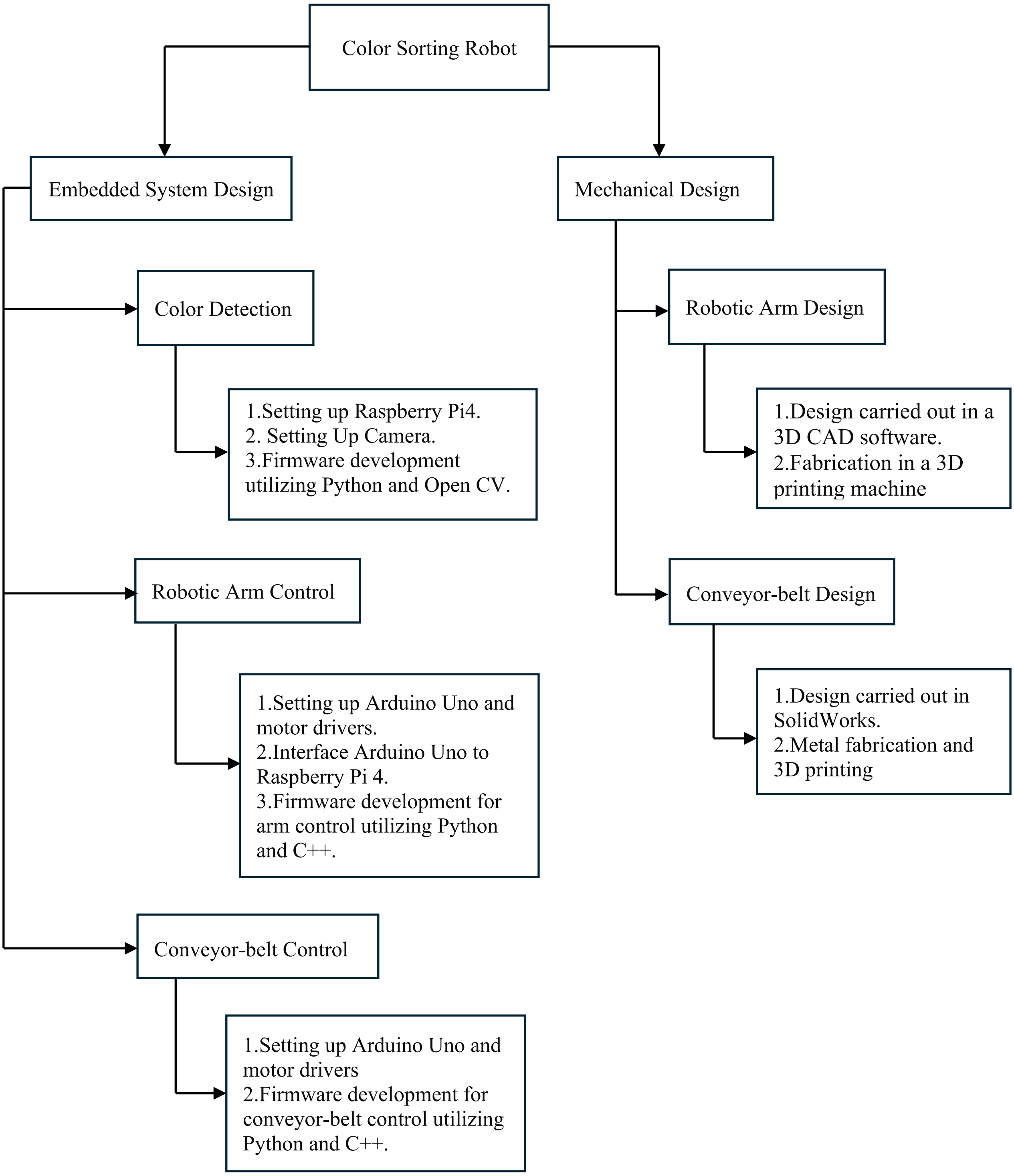

Figure 1 System design flow

Proposed embedded system

The embedded system architecture comprises a Raspberry Pi 4 serving as the master controller, with an Arduino Uno functioning as a slave controller to interface with motor drivers and regulate the mechanical actuation of the robotic arm and conveyor belt. As illustrated in Table 1, a comparative analysis of various Raspberry Pi models confirms that the Raspberry Pi 4 fulfills the computational demands necessary for real-time image processing tasks.

Table 1 Comprehensive comparison of various raspberry pi models [14]

Model	CPU	RAM	USB Ports	Ethernet	WiFi	Bluetooth	GPIO Pins	HDMI Ports	Size	Power
Raspberry Pi Model B	Single-core 700MHz	512MB	2	Yes	No	No	26	1	85.60mm x56mm	5V
Raspberry Pi 2	Quad-core 900MHz	1GB	4	Yes	No	No	40	1	85.60mm x56mm	5V
Raspberry Pi Zero	Single-core 1GHz	512MB	1	No	No	No	40	mini HDMI	65mmx30mm	5V
Raspberry Pi 3 Model B+	Quad-core 1.4GHz	1GB	4	Yes	Yes	Yes	40	1	85.60mm x56mm	5V
Raspberry Pi 4 Model B	Quad-core 1.5GHz	2-8GB	4	Yes	Yes	Yes	40	2 micro HDMI	85.60mm x56mm	5V

For image acquisition in color detection, a compatible camera module was selected to ensure seamless integration with the Raspberry Pi 4. This study employed a 5-megapixel (MP) camera from Arducam, which interfaces with the Raspberry Pi 4 via the Mobile Industry Processor Interface (MIPI). While a lower-resolution camera could theoretically suffice, the marginal cost difference rendered higher specifications a more practical choice.

Color recognition process

The system performs real-time color identification using HSV color space thresholding and contour detection algorithms implemented through OpenCV. As shown in Figure 2.

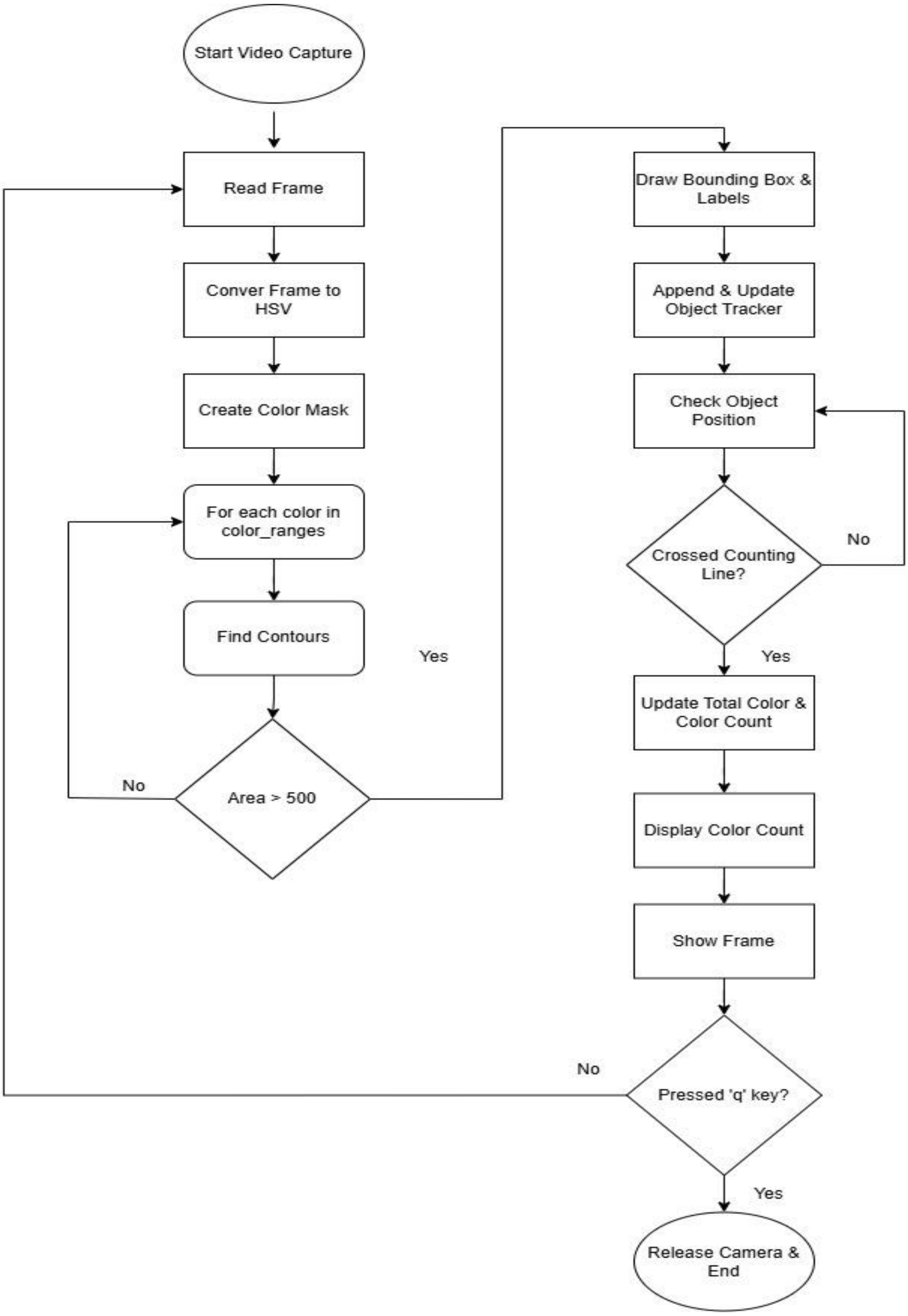

Figure 2 Color detection flowchart

Proposed mechanical structure

The mechanical assembly comprises two primary components:

1. The robotic arm designed using CAD software and manufactured via 3D printing.
2. The conveyor-belt system was CAD-modeled and fabricated via metalworking processes.

Robotic arm

The robotic arm used for this project is a standard programmable mechanical system composed of multiple rigid segments connected by motorized joints, forming a kinematic chain. Each joint—actuated by servo motors—enables rotational or translational movement, granting the arm its range of motion.

Conveyor Belt

The conveyor belt system transports materials sequentially for recognition in an assembly-line configuration. As illustrated in Figure 3, the complete system architecture is presented through detailed computer-aided design (CAD) schematics, which provide comprehensive visualization of the integrated components and their spatial relationships.

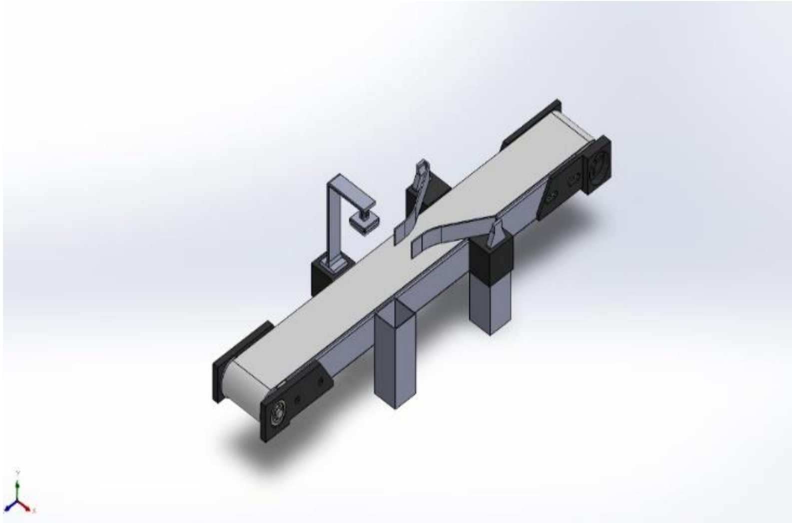

Figure 3 3D view of the complete conveyor belt

Its continuous motion is achieved through a closed-loop mechanism driven by two rollers positioned at opposite ends of the belt assembly. Figure 4 presents the computer-aided design (CAD) schematic of the roller component, illustrating its structural and functional details.

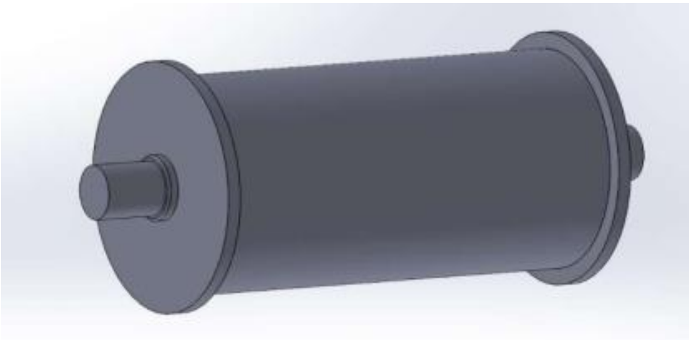

Figure 4 3D view of the roller design

System integration

To test the system, it was decided to sort based on three colors: yellow, green, and blue. However, the system can be adaptable to other colors easily. The overall flow of the system is depicted in Figure 5.

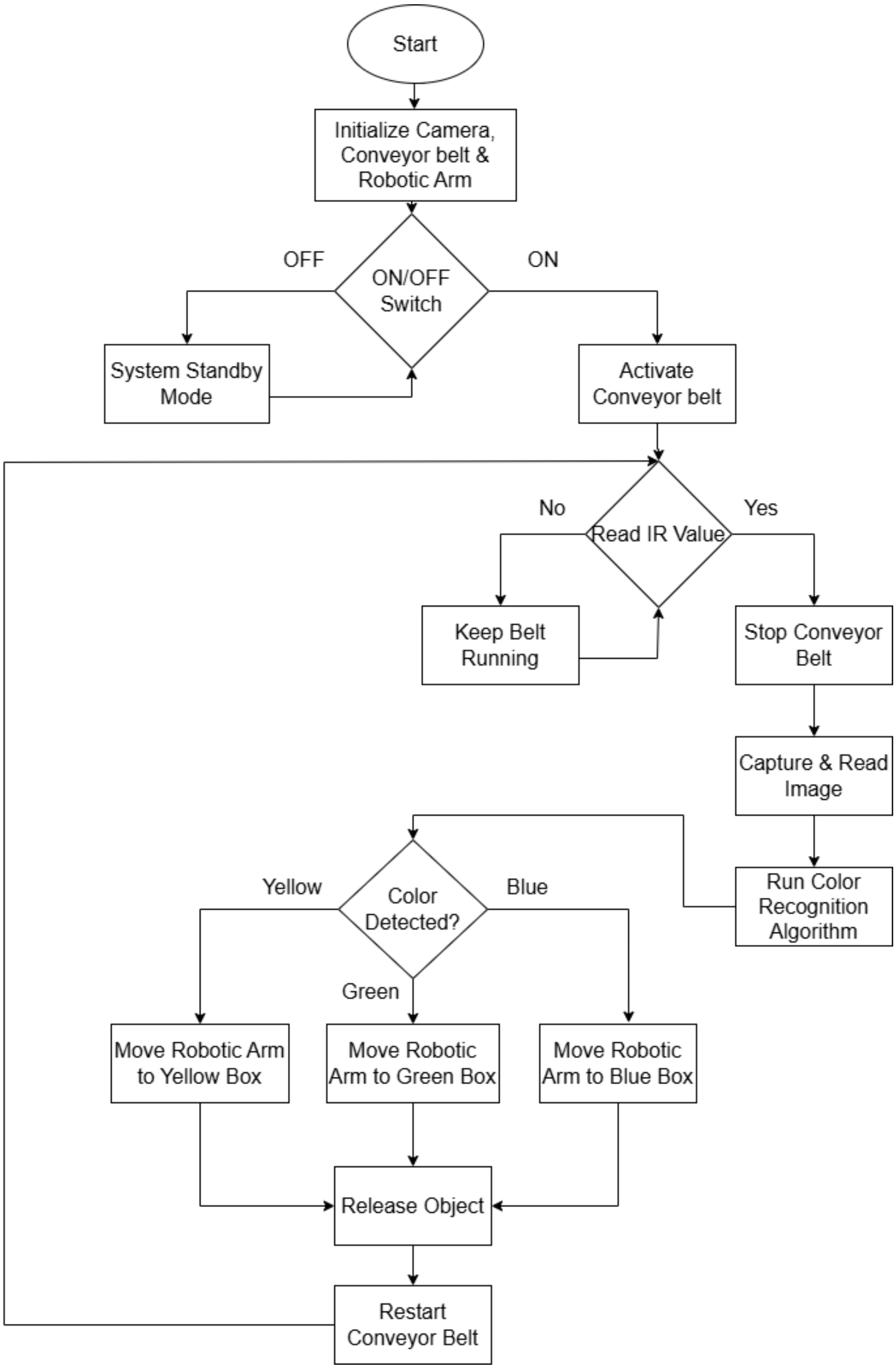

Figure 5 System integration flowchart

II. RESULTS

The color-based sorting robotic system was successfully designed, implemented, and evaluated. The complete system is shown in Figures 6.

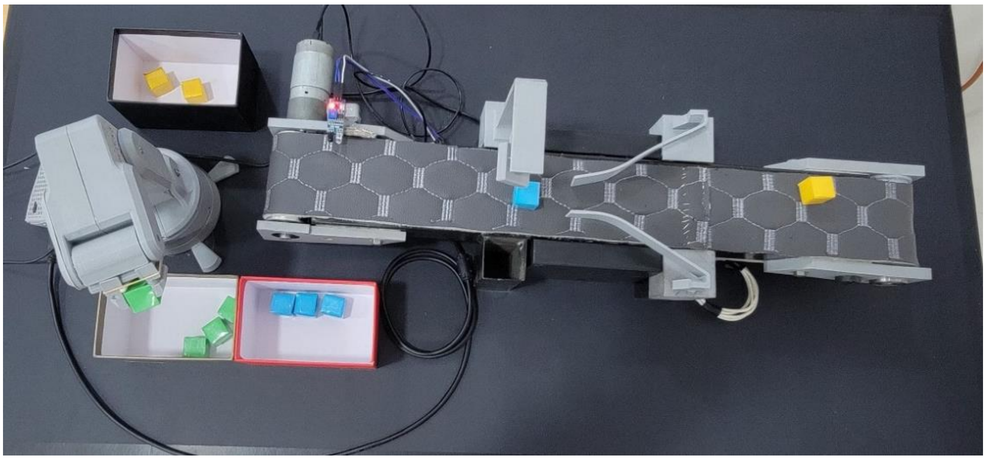

Figure 6 Prototype of the computer vision-based sorting system performing real-time color classification.

Through systematic testing, the system demonstrated the capability to autonomously classify and sort objects based on color with an average processing rate of one item every five seconds. The system exhibited low latency in processing images and executing sorting commands, with an average response time of 0.5 seconds from detection to giving an action command to the arm to pick up the object.

The color detection algorithm was evaluated under various lighting conditions. The algorithm demonstrated high accuracy, successfully identifying and categorizing ~90% of the items correctly under standard lighting conditions. Under challenging lighting scenarios, such as low light and varying light intensity, the accuracy might decrease to ~85%, indicating the algorithm's robustness and reliability.

The conveyor belt maintained a somewhat consistent speed, ensuring smooth movement of items to the detection area. Some mechanical failures were observed during testing where the conveyor belt had some aligning issues where the object would swing to the sides, which made the object not land where the robot arm grip was. The issue was solved by designing a funneling system on each side to keep the object in the middle, as seen in Figure 7, highlighting a more functional design.

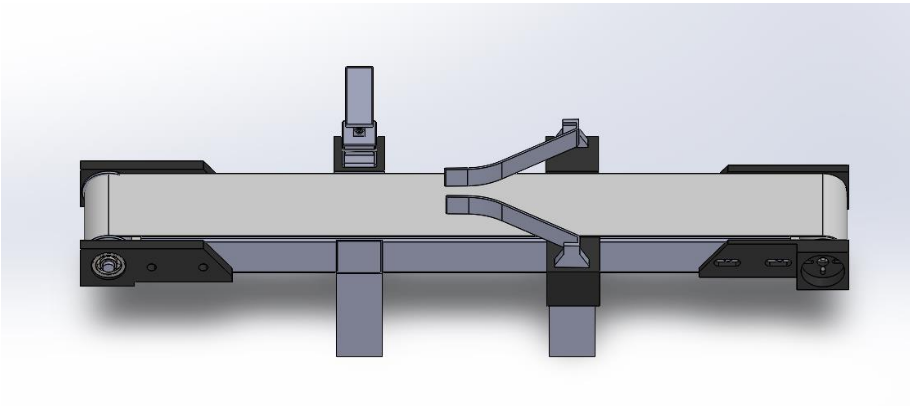

Figure 7 Top view of the convyer belt showing the implementation of the funneling system

III. Discussion

The primary objectives of this work were successfully achieved. The development and implementation of a functional color-based sorting robot, a reliable color detection algorithm, and the seamless integration between mechanical and embedded systems were all realized. It was demonstrated that it is possible to create an efficient and affordable sorting system using readily available and cost-effective components.

The high accuracy of the color detection algorithm under standard conditions validates the approach taken. The slight drop in performance under challenging lighting conditions suggests that further refinements could be made, such as implementing more advanced preprocessing techniques or adaptive thresholding to enhance robustness. The algorithm's performance indicates its suitability for industrial applications where lighting conditions can be controlled to some extent.

IV. Conclusion

This work demonstrated the design and implementation of a color-sorting machine based on a color thresholding algorithm; the design demonstrated a balance between accuracy, adaptability, and cost by using affordable computing boards and nonproprietary open-source computer vision libraries. The system successfully operated the conveyor belt and robotic arm to sort objects based on their color at a rate of 12 items per minute (one item every five seconds).

By employing a programmable solution based on image processing, this work can be adaptive to diverse industrial environments by updating the code base developed in this process. In any future effort, one could build upon this work to expand functionality to include multi-feature detection. Another area for improvement is in boosting system throughput by employing a sorting mechanism other than a robotic arm, which would limit the adaptability of the system but increase its throughput.

In this project, the system was designed to detect and sort objects based on three colors: yellow, green, and blue. However, the same method can be improved to detect more colors by adjusting the HSV values in the code. Additionally, integrating shape recognition using contour approximation would enable the sorting of objects based not only on color but also on geometric shape (e.g., cubes, cylinders, spheres). This improvement will make the system more useful for different industrial applications that require sorting by more than one feature.

V. References

[1] Y. Jia, G. Yang and J. Saniie, "Real-time color-based sorting robotic arm system," 2017 IEEE International Conference on Electro Information Technology (EIT), Lincoln, NE, USA, 2017, pp. 354-358, doi: https://doi.org/10.1109/EIT.2017.8053385.

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

[3] Medojevic, I., Veg, E., Joksimovic, A., & Ilic, J. (2022). Promotion of Color Sorting in Industrial Systems Using a Deep Learning Algorithm. *Applied Sciences*, *12*(24), 12817, doi: https://doi.org/10.3390/app122412817

[4] M. Haggag, S. Abdelhay, A. Mecheter, S. Gowid, F. Musharavati and S. Ghani, "An Intelligent Hybrid Experimental-Based Deep Learning Algorithm for Tomato-Sorting Controllers," in IEEE Access, vol. 7, pp. 106890-106898, 2019, doi: https://doi.org/10.1109/ACCESS.2019.2932730.

[5] M. Abdullah-Al-Noman, A. N. Eva, T. B. Yeahyea, and R. Khan, " Computer Vision-based Robotic Arm for Object Color, Shape, and Size Detection," Journal of Robotics and Control (JRC), vol. 3, no. 2, pp. 180–186, 2022, doi: https://doi.org/10.18196/jrc.v3i2.13906

[6] A. E. Elwakeel, Y. S. A. Mazrou, A. A. Tantawy, A. M. Okasha, A. H. Elmetwalli, S. Elsayed, and A. H. Makhlouf, "Designing, optimizing, and validating a low-cost, multi-purpose, automatic system-based RGB color sensor for sorting fruits," Agriculture, vol. 13, no. 9, p. 1824, 2023, doi: https://doi.org/10.3390/agriculture13091824

[7] A. R. M. Khairudin, M. H. A. Karim, A. A. Samah, D. Irwansyah, M. Y. Yakob, and N. M. Zian, "Development of colour sorting robotic arm using TCS3200 sensor," in 2021 IEEE 9th Conference on Systems, Process and Control (ICSPC), 2021, pp. 108–113, doi: https://doi.org/10.1109/ICSPC53359.2021.9689114

[8] S. Sinha, S. K. Suman, and A. Kumar, "Color Sensor-Based Object Sorting Robotic Arm," in Computing Algorithms with Applications in Engineering: Proceedings of ICCAEEE 2019, Springer, 2020, pp. 169–180, doi: https://doi.org/10.1007/978-981-15-2369-4_16

[9] S. N. Kulkarni and S. Kumar Singh, "Object Sorting Automated System using Raspberry Pi," 2018 3rd International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2018, pp. 217-220, doi: https://doi.org/10.1109/CESYS.2018.8724056.

[10] R. R. Mhaski, P. B. Chopade and M. P. Dale, "Determination of ripeness and grading of tomato using image analysis on Raspberry Pi," *2015 Communication, Control and Intelligent Systems (CCIS)*, Mathura, India, 2015, pp. 214-220, doi: https://doi.org/10.1109/CCIntelS.2015.7437911.

[11] V. Pereira, V. A. Fernandes and J. Sequeira, "Low cost object sorting robotic arm using Raspberry Pi," 2014 IEEE Global Humanitarian Technology Conference - South Asia Satellite (GHTC-SAS), Trivandrum, India, 2014, pp. 1-6, doi: https://doi.org/10.1109/GHTC-SAS.2014.6967550.

[12] P. Yadav, M. Uikey, P. Lonkar, S. Kayande and A. Maurya, "Sorting of Objects Using Image Processing," 2020 IEEE International Conference for Innovation in Technology (INOCON), Bangluru, India, 2020, pp. 1-6, doi: https://doi.org/10.1109/INOCON50539.2020.9298360.

[13] I. B. Mustaffa and S. F. B. M. Khairul, "Identification of fruit size and maturity through fruit images using OpenCV-Python and Rasberry Pi," Proceeding 2017 Int. Conf. Robot. Autom. Sci. ICORAS 2017, vol. 2018-March, pp. 1–3, 2018, doi: https://doi.org/10.1109/ICORAS.2017.8308068.

[14] Raspians. (n.d.). Raspberry Pi models comparison. Retrieved from https://raspians.com/raspberry-pi-models-comparison/