Color Sorting Robot

Harish Thangaraj (21BRS1033), Raahul C (21BRS1044)

Machine learning for robotics (BCSE424L)
at
Vellore Institute Of Technology, Chennai

April 2024

Prof. Rajarajeswari S

Abstract

In recent decades, robots have become indispensable assets in industrial automation, significantly transforming manufacturing processes across various industries. This abstract highlights one such innovative industrial automation process: color sorting objects. The system described utilizes computer vision algorithms and the YOLOV8 architecture to accurately identify and sort objects based on their RGB values. This project explores the development, implementation, and performance evaluation of the color sorting system, demonstrating its potential to streamline production workflows, improve quality control, and enhance overall manufacturing efficiency. Further, we have discussed the incorporation of reinforcement learning to fine tune such object detection models. The combined framework can dynamically adapt to changing environments and improve detection performance under various conditions.

Declaration

We, the authors of this project report, hereby confirm that all the content contained within this document has been generated solely by ourselves. We affirm that we have conducted the methodology presented and composed this report with the utmost integrity and honesty.

Throughout the course of our work, we have consulted various sources of information from the internet and other reference materials to inform our research and analysis which have been acknowledged appropriately.

Furthermore, We would like to thank our professor in charge and institution for providing the necessary guidance and resources to complete our project succesfully.

Harish Thangaraj (21BRS1033), Raahul C (21BRS1044)

Contents

Chapter 1

Motivation

Our primary motivation to implement a color sorting robot focuses on the advantages offered by automated sorting compared to manual sorting in an industrial setting.

Manual sorting :

Manual sorting typically involves human workers visually inspecting items as they pass along a conveyor belt or assembly line, then making decisions on how to categorize them accordingly. While manual sorting offers the advantage of flexibility, allowing workers to adapt to various objects and situations, it is also inherently limited by factors such as human error, fatigue, and throughput capacity. Human workers may struggle to maintain consistency over extended periods, leading to variability in sorting accuracy and efficiency. Moreover, manual sorting processes are often labor-intensive and time-consuming, with the potential for increased operational costs associated with hiring and training personnel

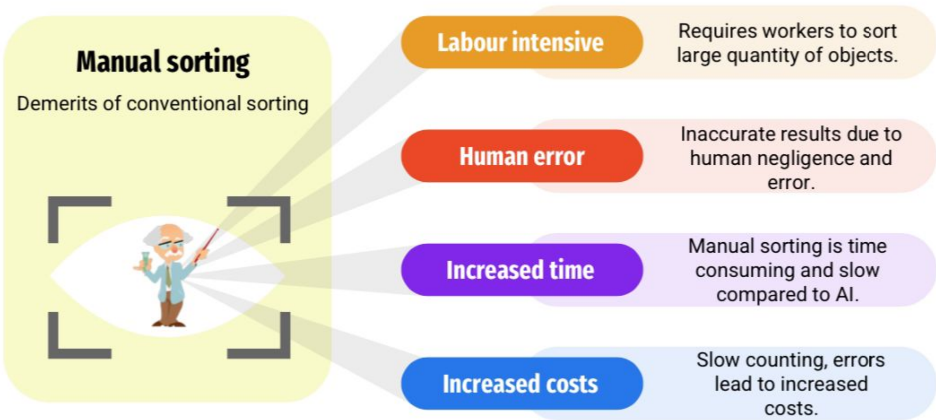

Fig 1. Demerits of manual sorting

Automated sorting :

In contrast, the emergence of color sorting robots represents a transformative shift in industrial automation, offering a viable alternative to manual sorting processes. These robots leverage advanced technologies such as machine vision, artificial intelligence, and robotics to automate the sorting process with precision and efficiency. Color sorting robots are equipped with sophisticated cameras and sensors that enable them to accurately identify objects based on their color attributes in real-time. This capability allows for rapid sorting at high speeds and with minimal errors, leading to increased throughput and productivity. Additionally, color sorting robots can operate continuously without breaks, resulting in improved efficiency and consistency over manual labor.

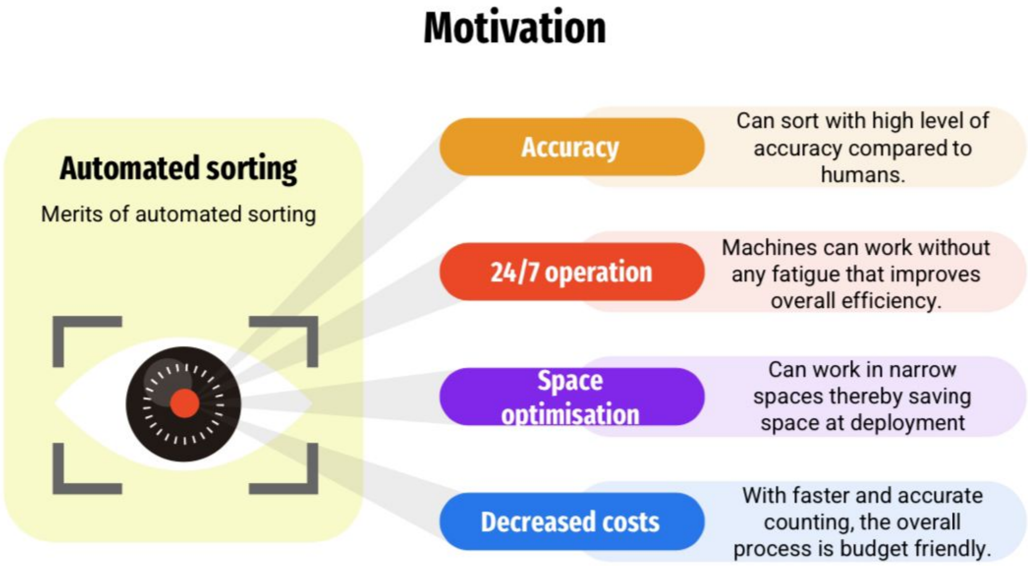

Fig 2. Merits of automated sorting

Chapter 2

Objective

The objective of developing an automated color sorting robot is to create a robust and efficient system capable of accurately identifying and sorting objects based on their color in industrial setting. Specifically the objectives include :

Table 1. Objectives of the project

Objective	**Description**
Enhancing efficiency	Improving the speed and effectiveness of sorting processes.
Increasing productivity	Boosting the output of sorted items within a given time-frame.
Ensuring accuracy	Achieving precise identification and sorting of objects based on color attributes.
Facilitating integration	Seamlessly integrating the color sorting robot into existing industrial workflows.
Reducing labor costs	Minimizing expenditures associated with manual labor and sorting tasks.
Quality control	Maintaining high standards of product quality through meticulous sorting procedures.
Improving consistency	Ensuring uniformity and reliability in sorting operations across different batches.
Promoting safety	Enhancing workplace safety by automating potentially hazardous sorting tasks.

Chapter 3

Related works

3.1 Literature review

The literature review presented in this section aims to provide a comprehensive overview of the research and development efforts in the field of sorting robots. It explores the evolution of sorting technologies, examines the underlying principles of color detection and classification, and discusses the key components and functionalities of modern color sorting systems. Additionally, the review investigates the applications, benefits, and challenges associated with the implementation of color sorting robots in various industrial contexts.

1. Yonghui Jia et al. in their paper, present an end to end color sorting robotic arm that is driven by an arduino micro-controller, servo motors and a computer vision algorithm to sort object in a three-dimensional space. Their research focuses on using RGB based object localization using OpenCV and picking and placing of distinguished objects. At an industrial setting, very specific objects are to be sorted and at high speed which may not be achievable using the discussed method.

2. Uzma Amin1 et al have implemented a hardware centric approach to color sorting using a TCS 3200 color sensor. The PIC micro-controller promises high speed performance with a desirable accuracy at an industrial environment. yet, object localization of specific kind of objects will need additional paradigms to be included.

3. Tuong Phuoc To and Nguyen Truong Thinh have propsed a size based sorting system of objects moving on a conveyor belt. Their methodology aims to classify and sort tomatoes as ripe and raw using the object's position as deciding parameters. This system allows localization of specific objects described by their size and position. Although, the accuracy when an unwanted object of the similar size comes into the picture may not be desirable.

4. Lekha Bhausaheb Kachare in her study has taken a path away from conventional openCV based detection by using MATLAB's computer vision paradigms. The image processing is achieved by using a simple webcam and an ARM based processor. Such a system is preferable on small settings where performance is not a factor to be considered as MATLAB offers relatively

slower image processing and offers fewer inbuilt modules.

5. Malvin Nkomo and Michael Collier have worked on a comprehensive full scale SCARA robotic arm that possesses a on system chip, a programmable color to light frequency chip, color and IR sensors. While this implementation focuses on using hardware modules, it acts as a primary reference to our project. The integration of machine learning paradigms with feed-back based learning and hardware modules can provide to be a powerful framework to large scale sorting.

3.2 Research gap

While existing sorting algorithms predominantly rely on OpenCV libraries or sensor-based techniques for object identification and classification, there remains a notable research gap in the realm of leveraging advanced deep learning architectures for enhanced efficiency and adaptability. Traditional methods often encounter limitations in terms of speed, accuracy, and adaptability to dynamic environments. Our approach aims to address this gap by harnessing the capabilities of the You Only Look Once (YOLO) architecture coupled with OpenCV and reinforcement learning techniques. YOLO architecture, known for its real-time object detection capabilities, promises unparalleled speed and accuracy in identifying objects based on their attributes. By integrating reinforcement learning, our proposed framework enables the sorting robot to dynamically adapt to changing environmental conditions and optimize sorting strategies over time. This presented approach not only enhances the efficiency and accuracy of sorting processes but also lays the groundwork for autonomous and adaptive sorting systems capable of meeting the evolving demands of industrial automation.

Chapter 4

Proposed System

The system introduced here aims to categorize cubes (target objects) according to their color. Initially, the framework is tasked with identifying the target objects within its defined environment. Subsequently, it proceeds to recognize the color of the objects and execute the sorting process accordingly. The development and implementation of this pipeline can be broadly categorized into three modules.

4.1 HSV parameters

A colored cube dataset comprising of 499 cube images was selected and the following pre-processing techniques were applied :

Table 2. Pre-processing techniques applied

Technique	**Description**
Auto-orient	Setting an uniform image orienation
Resize	Stretch to 640x640.
Flip augmentation	Horizontal flip.
Blur augmentation	Upto 2.5px.
Noise augmentation	Up to 5% of pixels.
Cutout augmentation	3 boxes with 10% size each.

The images are further subjected to conversion of the RGB color values of the cubes to the HSV color space. By transforming the color representation from RGB to HSV, Three distinct channels are generated: Hue, Saturation, and Value. The Hue channel encodes information about the dominant color of the cube, while Saturation and Value provide details regarding the intensity and brightness of the color, respectively. Subsequently, the HSV parameters from each image are extracted, creating a feature representation that encompasses both the spatial and color characteristics of the cubes.

4.2 YOLOv8 model

The utilization of YOLO, as discussed in previous sections of this paper, showcases remarkable efficiency while upholding superior accuracy in tasks related to object detection. The model adopts a single-pass architecture, enabling it to predict bounding boxes and class probabilities directly from the entire image in a single evaluation. It also employs a grid-based approach, dividing the input image into a grid and predicting objects within each grid cell simultaneously facilitating

The integration of the generated HSV features into YOLO detection provides improved color discrimination and robust performance of the model. The variations in illumination at an industrial setting can be tackled by adopting this approach where colors are distinguished using a range of values.

Table 2. HSV parameters

Color	Hue (H) Range	Saturation (S) Range	Value (V) Range
Red	$0° - 30°$ or $330° - 360°$	$50\% - 100\%$	$50\% - 100\%$
Green	$90° - 150°$	$50\% - 100\%$	$50\% - 100\%$
Blue	$210° - 270°$	$50\% - 100\%$	$50\% - 100\%$

The dataset was split into 438 training, 42 validation and 19 testing images respectively to train and evaluate the model.

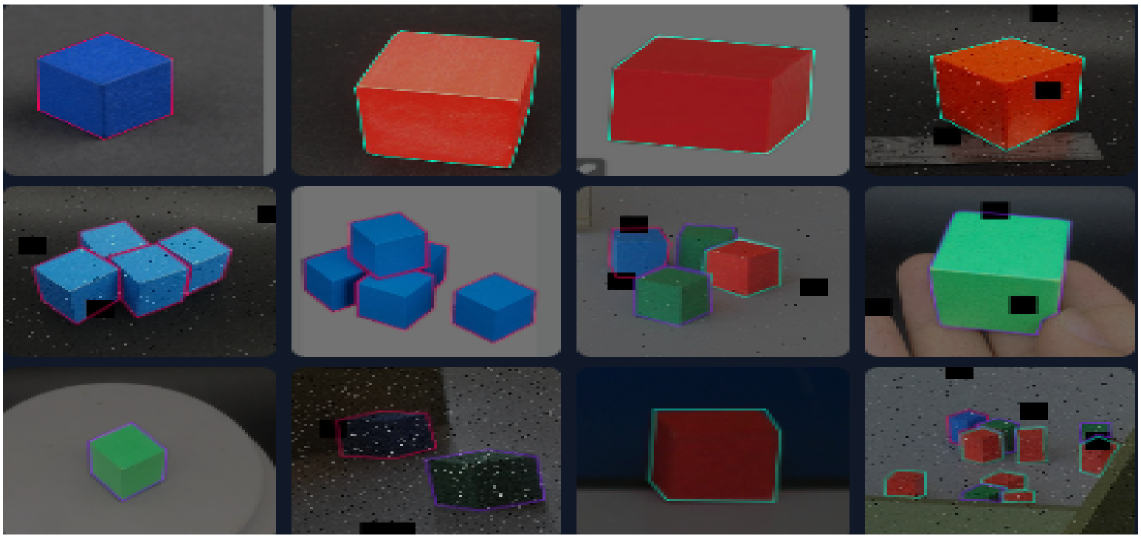

Fig 3. Pre-processed dataset

Following pre-processing and splitting of the dataset, the YOLOv8 model was trained for 10 epochs promising an overall accuracy of 93%. Further, by leveraging OpenCV's computer vision paradigms, we were able to configure tagging of the cubes using a live video stream and plot bounding boxes around the target cube. At a real word setting, this becomes a fundamental requirement for such applications where the system is able to tag and retrieve the target object's coordinates through a live camera/sensor feed. The captured coordinates of the object are returned for further manipulation.

4.3 Robot simulation

The sorting operation was simulated using webot robotic simulation software considering its simplicity and python integration support. A pick and place template was loaded into the simulation environment comprising a conveyor belt of colored cubes (soft objects) and a robot with a grasping tool.

In the simulation environment, a camera sensor was mounted on the robot's end effector, providing a live video feed of the conveyor belt and the surrounding environment. This camera feed served as the input for the YOLO model, allowing the model tag colored cubes.

Upon detecting a colored cube in the camera feed, the YOLO model provided the robot with bounding box coordinates and corresponding class labels for the detected objects. These bounding box coordinates were then used by the robot's control script to calculate the position and orientation of the target cube relative to the robot's gripper.

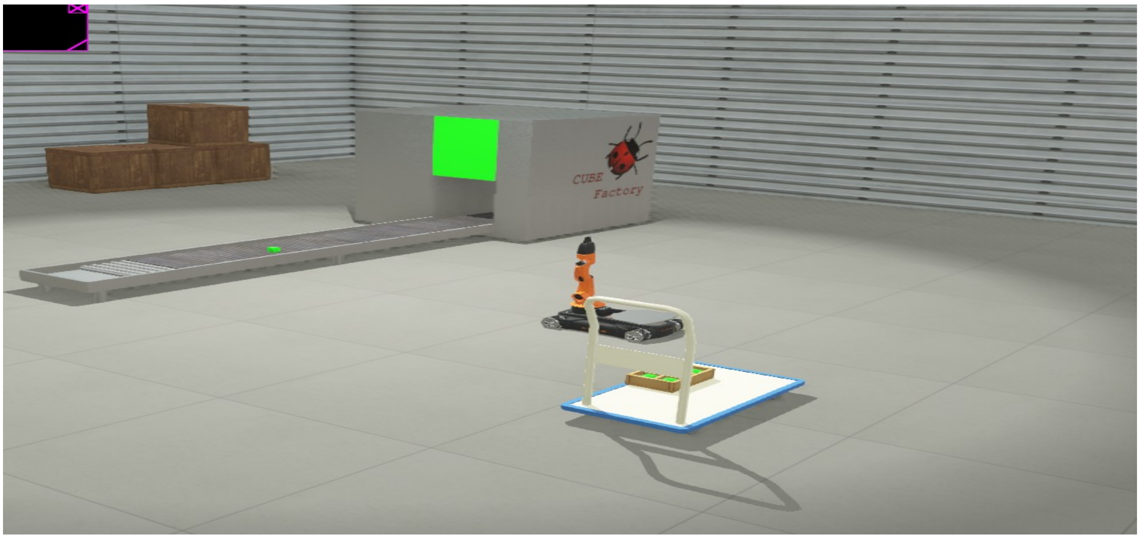

Fig 4. Simulation of robot in webots software

Chapter 5

Result & Discussion

5.1 Result

The end to end system, equipped with the customized YOLOv8 model incorporating HSV parameters, demonstrated robust performance in accurately detecting and sorting colored cubes.

The model achieved a high level of accuracy in detecting colored cubes in varied lighting conditions. The incorporation of HSV parameters significantly improved the model's ability to discriminate between different colors, resulting in precise localization and classification of cubes.

```
0: 640x640 1 bluecube, 1 green cube, 1 red cube, 78.5ms
Speed: 2.6ms preprocess, 78.5ms inference, 0.5ms postprocess per image at shape (1, 3, 640, 640)
```

Fig 5. Detection of cubes - output

The color sorting process was performed in real-time, with the YOLOv8 model processing each frame of the simulation video at a high speed. This real-time capability is crucial for practical applications requiring rapid and continuous sorting of objects in dynamic environments. The following metrics were obtained :

Table 3. Evaluation metrics of YOLOv8 model

Class	Images	Instances	Box Precision	Recall	mAP@50	mAP
all	42	98	0.938	0.993	0.989	0.929
blue cube	42	23	0.845	1	0.978	0.924
green cube	42	37	0.968	1	0.995	0.943
red cube	42	38	1	0.978	0.995	0.919

Overall, the results obtained highlight the effectiveness of integrating HSV parameters into the YOLOv8 model for color-based object detection and sorting tasks, paving the way for enhanced automation and efficiency in industrial and robotic applications.

5.2 Discussion on reinforcement learning

Application of reinforcement learning holds great potential for enhancing the performance of this color sorting robot. By leveraging reinforcement learning algorithms, such as deep Q-learning and policy gradient methods, the robot can dynamically adapt its sorting strategy, optimize trajectory planning, and continuously improve its performance through iterative learning. Reinforcement learning also enables the robot to develop robust and adaptive behaviors, handle uncertainty in its environment, and generalize its sorting skills to new tasks and environments with minimal retraining.

Considering the time constraints of the current study, further exploration and implementation of the proposed approach will be pursued as future work

Chapter 6

Conclusion

6.1 Conclusion

In this study, we have presented the development and simulation of a color sorting robot utilizing advanced computer vision techniques and machine learning algorithms. By customizing the YOLOv8 model to incorporate HSV color space information we have demonstrated the potential for enhancing the efficiency and accuracy of color-based object sorting tasks. Through experimentation and analysis, we have observed promising results, including high accuracy in cube detection, efficient sorting performance, and robustness to variations in lighting conditions.

6.2 Future Work

In future work, we aim to enhance the current system and experiment in rigorous real world settings. By leveraging reinforcement learning, we aim to further optimize the robot's performance in sorting tasks enabling a adaptive behaviour and minimize human interaction. Overall, this project expands the foundation for the development of intelligent and versatile color sorting robots with broad applications in industrial automation, logistics, and manufacturing industries.

Chapter 7

Appendix

7.1 Codebase and Dataset

1. https://github.com/rulezcasa/color-sorting-robot
2. https://universe.roboflow.com/jakub-slof/red-green-blue-cube-detection/dataset/2