Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 10, 32-49
2025
Publisher: Learning Gate
DOI: 10.55214/2576-8484.v9i10.10340

Design and development of an automated fruit quality classifier and sorter using a robotic arm

Tan Wei Xiang[1], Lim Sin Ting[1*], Koo Voon Chet[1], Lim Way Soong[1]
[1]Faculty of Engineering and Technology, Multimedia University, Jalan Ayer Keroh Lama,75450 Melaka. Malaysia; stlim@mmu.edu.my (L.S.T.).

Abstract: This study presents the design and development of an automated apple quality classification and sorting system to overcome key limitations of manual grading, such as inconsistency, inaccuracy, and labor dependency. The primary objectives are to detect and classify fruit quality based on visual parameters such as color, shape, and defects, and to employ a robotic arm that autonomously sorts the apples into predefined quality categories based on the classification results. A YOLOv8 segmentation model was trained and deployed on a Raspberry Pi 4 Model B to perform real-time classification of apples as good or bad. Further analysis using OpenCV was applied to good apples by evaluating red color ratio and fruit size (height and width) to determine premium or standard grade. The hardware setup includes a PiCamera for image capture and servo motors for sorting and rotating apples to support both single-side and multiple-side detection modes. The YOLOv8 system achieved a mean average precision (mAP) of 96.56%. Hardware testing demonstrated improved classification accuracy from 97% using single-side detection to 100% using multiple-side detection across 60 apple samples. A Tkinter-based graphical user interface (GUI) was developed to allow users to control detection modes, view classification outcomes, and monitor sorting actions in real time. This system offers a low-cost and scalable solution for automated fruit sorting. Its adaptability to multiple detection angles and integration with physical sorting mechanisms make it a practical option for small-scale agricultural automation and research environments.

Keywords: *Apple quality classification, Multi-angle detection, Robotic arm, YOLOv8 segmentation.*

1. Introduction

As the agricultural industry faces increasing demand for productivity and consistent quality, automation has emerged as a key solution to address the limitations associated with traditional manual methods. Manual fruit grading and sorting processes typically suffer from significant drawbacks such as inconsistency, inaccuracy, and high dependency on human labor. These shortcomings result in compromised fruit quality and inefficiencies in supply chains, ultimately affecting the profitability and reliability of agricultural operations. Therefore, to overcome these challenges and improve operational efficiency, the adoption of advanced technologies such as artificial intelligence (AI) and robotics is increasingly explored [1].

Traditionally, fruit quality assessment involves human evaluators visually inspecting fruits based on features such as color, size, and the presence of defects. However, human evaluation is inherently subjective and influenced by fatigue, environmental conditions, and varying skill levels among workers, leading to variations in grading outcomes [1]. Recent advances in AI-powered computer vision techniques offer promising solutions to overcome these challenges by automating the visual inspection process. Among these, object detection and segmentation models, particularly the YOLO (You Only Look Once) series, have demonstrated remarkable capabilities in real-time and accurate classification

Received: 12 August 2025 / Revised: 15 September 2025 / Accepted: 17 September 2025 / Published: 3 October 2025
* Correspondence: stlim@mmu.edu.my

tasks. YOLOv8, the latest iteration of this technology, efficiently detects and classifies visual features, making it highly suitable for real-time fruit quality analysis [2].

Various automated systems have been proposed for fruit quality assessment, leveraging technologies ranging from simple mechanical sorting to sophisticated AI-integrated robotic systems. While mechanical sorting systems provide basic categorization based on simple physical attributes, they lack adaptability and accuracy in assessing detailed fruit conditions such as subtle surface defects or precise color gradation [3]. On the other hand, AI-integrated systems demonstrate significant improvements in accuracy and consistency but often require complex setups and multi-camera configurations, resulting in increased system cost and complexity [4]. AI-based fruit sorting systems, such as the YOLOv8-powered model by Hou et al. [5], demonstrate high classification accuracy for tomatoes based on visual features, though limited adaptability to other crops reduces their overall versatility. Abbas et al. [6] proposed an automated sorting and grading system for agricultural products that utilizes a single camera for image acquisition. While this approach is cost-effective, it limits the system's ability to capture images of fruit from multiple angles, which can reduce the accuracy of sorting and grading.

Filho et al. [7] developed a real-time fruit classification system using the TensorFlow Object Detection API and Raspberry Pi 5, achieving a mean average precision of 67.54% across four classes. However, its reliance on predefined classifications limits adaptability to other crop types.. Chakraborty et al. [8] developed an automated citrus fruit grading system combining a lightweight CNN model (SortNet) for real-time defect detection with weight-based categorization, achieving 97.6% accuracy in visual sorting and 91.3% in weight grading, demonstrating high precision and suitability for real-time applications. Tanvashi et al. [9] developed an AI-driven tomato sorting system using YOLOv5 and a Jetson Nano platform, achieving 97% mAP and 22.31 ms inference time for real-time classification into ripeness categories, though its single-view limitation due to the absence of a roller-type conveyor restricts comprehensive fruit assessment. Adamu et al. [10] developed an automated fruit sorting system using a TCS3200 color sensor and Arduino Mega to classify fruits based on ripeness, achieving 90% accuracy; while effective in reducing manual labor, its reliance on color limits assessment of other quality parameters, though functionality is enhanced via an Android-based remote-control application.

Motivated by these challenges and the limitations of existing solutions, this study introduces a novel automated fruit quality classification and sorting system integrating a YOLOv8 segmentation model, a robotic arm mechanism, and image processing techniques using OpenCV. The proposed system operates by capturing real-time images of apples using a PiCamera, analyzing them through the YOLOv8 model to classify fruits as good or bad, and further categorizing good apples into premium or standard grades based on additional analysis of red color ratio and size measurements. The robotic arm, controlled by servo motors and integrated with a rotating platform, supports both single-side and multiple-side detection modes, thereby enhancing the accuracy and robustness of the classification process.

Several studies have investigated automated sorting systems. However, few have combined the flexibility of multi-angle detection with a practical robotic arm sorting mechanism in a cost-effective, compact platform. Previous research has shown the effectiveness of AI-powered detection systems in single-view applications, yet these methods often fail to capture comprehensive fruit conditions that require multiple perspectives [5, 11]. Therefore, the integration of a rotating platform that enables multi-side detection in this study addresses this critical gap, providing a more reliable and accurate evaluation of fruit quality.

1.1. Experimental Setup

The block diagram, as shown in Figure 1, provides an overview of the input, process, and output stages of the apples classification and sorting system. The input section begins with the Raspberry Pi Camera Module Rev 1.3, which captures real-time images of the apples placed on the rotating platform. These images are sent directly to the Raspberry Pi 4 Model B, which serves as the central processing

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 10: 32-49, 2025
DOI: 10.55214/2576-8484.v9i10.10340

unit of the system. In the processing stage, the Raspberry Pi runs a YOLOv8 segmentation model to detect and segment whether the apple is of good or bad quality. If the fruit is classified as good, it is further analyzed using OpenCV to calculate its red colour intensity and size. These parameters are then used to determine if the apple should be categorized as premium or standard. The classification results of reject, standard, or premium are generated based on this detection and analysis.

The output section of the system consists of three MG996R servo motors and one TBS-K20 servo motor, which are powered by a 5V external power supply and controlled via the Raspberry Pi's GPIO pins. Two MG996R motors are responsible for controlling the movement of the 3D-printed robotic arm, including the base rotation and arm lifting functions. The TBS-K20 servo is used to control the gripper mechanism, enabling the robotic arm to grasp and release apples based on classification decisions. One of the MG996R servo motors is responsible for the rotating platform, executing physical sorting actions according to the classification output. The robotic arm places the apples into the corresponding bins based on their detected category. The rotating platform is designed and used in multiple-side detection modes, with the latter allowing the platform to rotate the apple to different angles for more comprehensive image capture. A Tkinter-based GUI is integrated into the system to display real-time classification results, video streaming, and system status. The GUI also provides control buttons, such as start and stop, to enhance user interaction and monitoring. This block diagram effectively demonstrates the integration of machine vision, AI-based classification, and mechanical actuation to achieve a real-time automated apple sorting system.

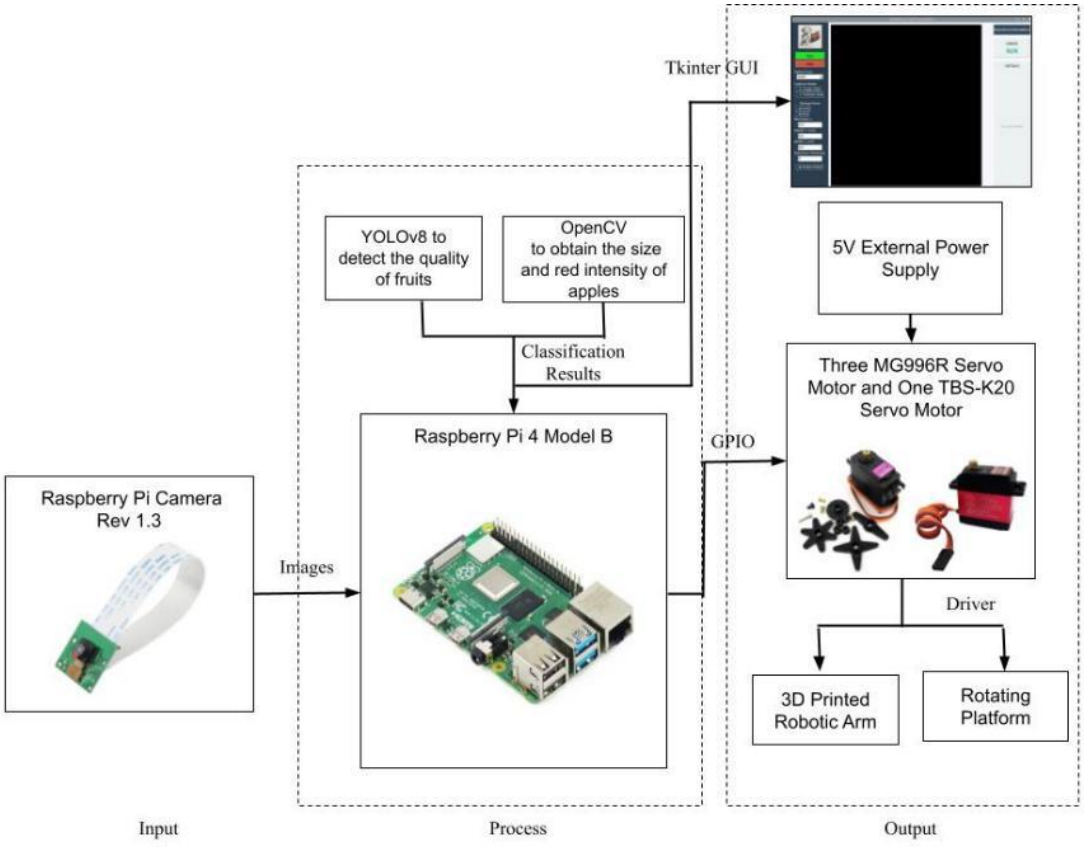

Figure 1.
Block Diagram of the system.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 10: 32-49, 2025
DOI: 10.55214/2576-8484.v9i10.10340

The flowchart depicted in Figure 2 illustrates the complete operational sequence of the automated apples quality classifier and sorter. The process begins with the initialization of key components, including the PiCamera, Raspberry Pi, MG996R servo motors, and TBS-K20 servo motor, which control the apple rotation and robotic arm. Once initialized, the camera feed is activated to enable real-time image capture. An apple is then placed on the rotating platform by the user. Before detection begins, the user selects the capture mode, either single side or multiple sides (7 sides at 0°, 30°, 60°, 90°, 120°, 150°, and 180°), as well as the sorting criteria, which may be based on colour, size, or both. These user-defined settings influence how the apple will be evaluated. The system captures an image of the apple and processes it using YOLOv8 and OpenCV on the Raspberry Pi. First, it determines whether the apple is of good quality. If not, it is immediately classified as rejected, and the robotic arm places it in the rejection bin, but the system still calculates colour intensity (red ratio) and size. If the apple is considered good, the system performs a second-level classification to determine whether it is premium or standard, based on threshold values for colour intensity (red ratio) and size. The robotic arm then picks and places the fruit into the corresponding bin. After sorting, the system checks for the presence of another apple. If another apple is detected, the system repeats the classification and sorting process. If no new apple is detected, the camera feed is turned off manually, and the system ceases operation.

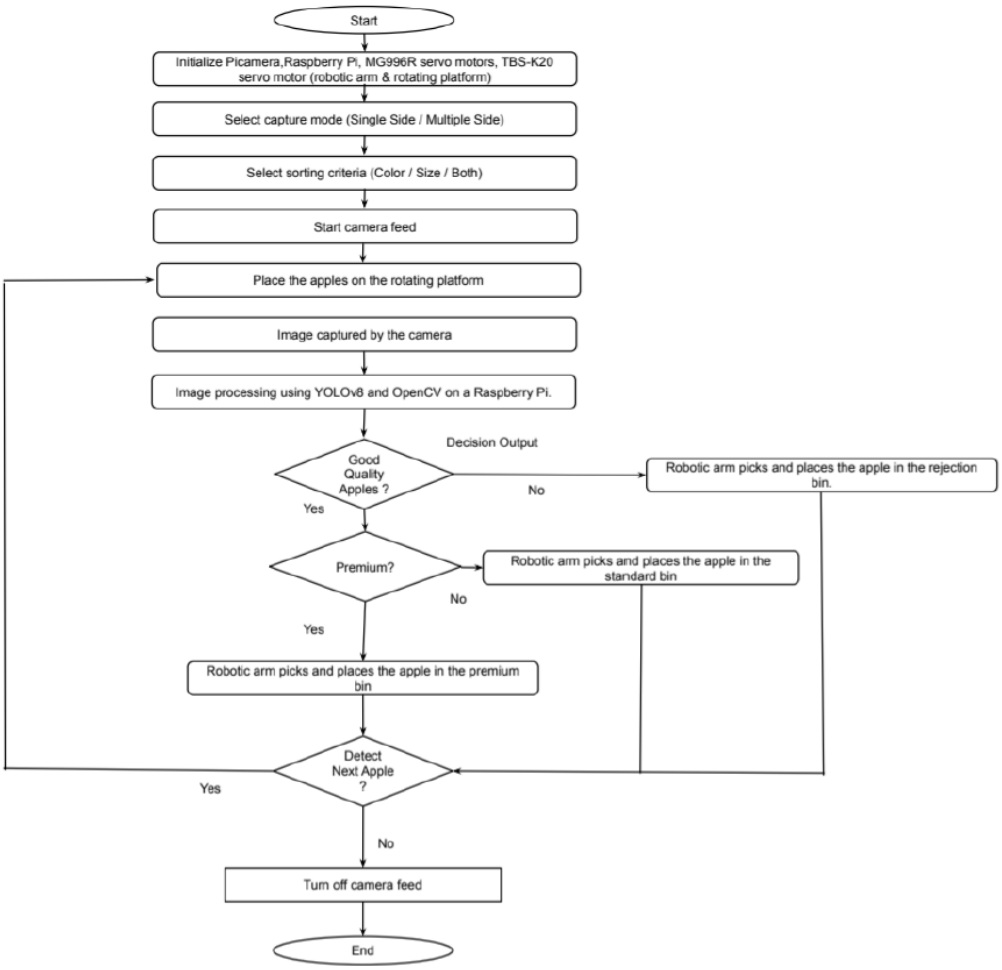

Figure 2.
Flowchart of the system.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 10: 32-49, 2025
DOI: 10.55214/2576-8484.v9i10.10340

1.2. Hardware Implementation

The schematic diagram shown in Figure 3 illustrates the complete electrical wiring of the apple classification and sorting system. At the core of the setup is the Raspberry Pi 4 Model B, which functions as the main controller for image processing and motor control. A Raspberry Pi Camera Module Rev 1.3 is connected to the Pi via the Camera Serial Interface (CSI) port to capture real-time images of the apples.

The system uses a total of three servo motors for the 3DOF robotic arm, comprising two MG996R servo motors and one TBS-K20 servo motor, as well as one MG996R servo motor dedicated to the rotating platform. Each servo motor is connected to a specific GPIO pin on the Raspberry Pi for PWM control. Specifically, GPIO 17 is assigned to control the base rotation of the robotic arm, GPIO 27 is used for the arm lifting movement, and GPIO 22 controls the TBS-K20 servo motor integrated within the gripper mechanism. The rotating platform, which spins the apple for multiple-angle detection, is driven by an MG996R servo motor connected to GPIO 18. These GPIO pins deliver PWM signals from the Raspberry Pi to each servo motor, allowing for precise and coordinated angular motion across the entire system.

Power for all four servo motors is supplied by an external 5V power adapter, which is connected to a breadboard power rail. The positive and ground lines from the adapter are distributed to the servo motors to prevent overloading the Raspberry Pi.

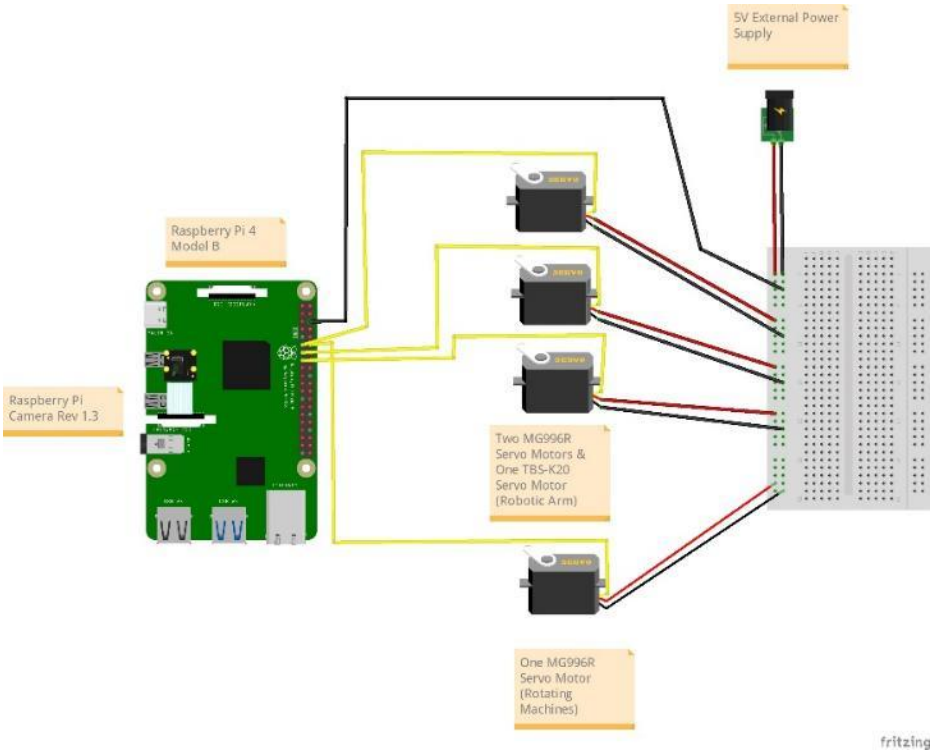

Figure 3.
Flowchart of the system.

A critical component of the system is the 3D printed 3-degree-of-freedom (3DOF) robotic arm depicted in Figure 4, specifically designed for precise fruit manipulation. This robotic arm integrates robust servo motors, including MG996R for primary actuation and TBS-K20 for finer control, enabling smooth and accurate pick-and-place operations. To enhance the system's ability to perform detailed fruit analysis, a rotating platform is integrated into the setup, driven by an MG996R servo motor. The

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 10: 32-49, 2025
DOI: 10.55214/2576-8484.v9i10.10340

rotating platform, detailed in Figure 5, provides controlled rotation, allowing apples to be viewed from multiple angles. This multi-angle assessment significantly improves the system's capability to identify subtle defects or variations in fruit quality.

Figure 4.
3D printed robotic arm.

Figure 5.
3D printed rotating platform.

The robotic gripper, showcased in Figure 6, incorporates soft sponge padding to minimize the risk of fruit damage during handling. The PiCamera Rev 1.3, strategically mounted using a custom-designed and 3D-printed camera stand, shown clearly in Figure 7, ensures stable, consistent, and repeatable positioning of the camera, which is essential for capturing high-quality images for accurate fruit classification and size analysis. This design guarantees gentle yet firm grasping, crucial for preserving the integrity and marketability of the apples being sorted. All the aforementioned hardware elements, such as the Raspberry Pi, robotic arm, rotating platform, and camera module, are securely assembled and mounted on a stable wooden platform, as depicted in Figure 8. This robust platform ensures the stability of the system during operation and facilitates straightforward transportation and testing in diverse environments.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 10: 32-49, 2025
DOI: 10.55214/2576-8484.v9i10.10340

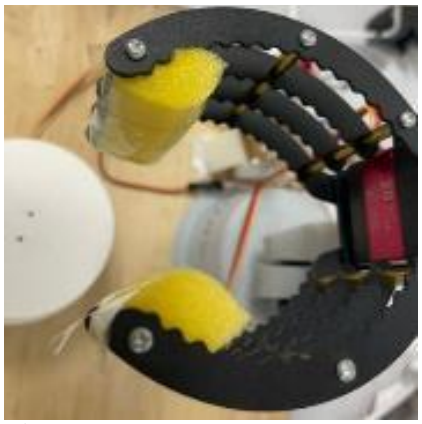

Figure 6.
Gripper modified with soft sponge padding.

Figure 7.
Gripper modified with soft sponge padding.

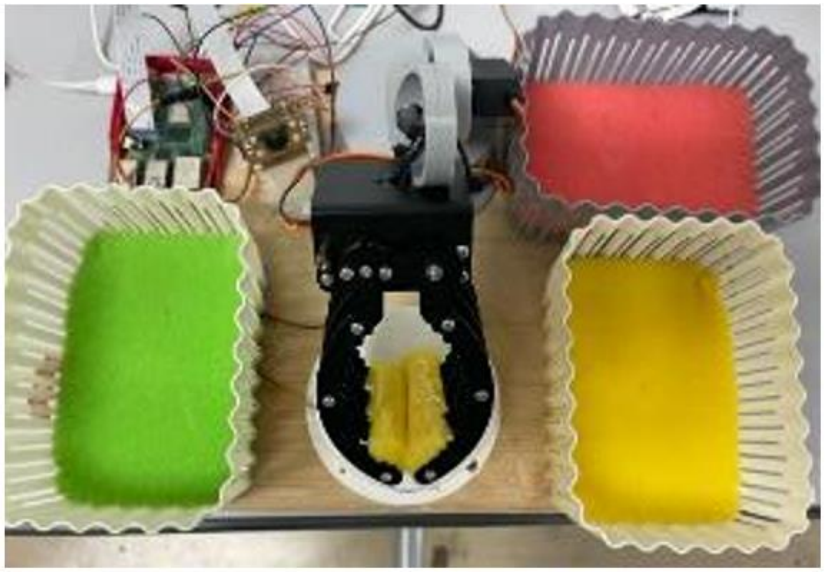

Figure 8.
Overall system.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 10: 32-49, 2025
DOI: 10.55214/2576-8484.v9i10.10340

1.3. Software Implementation

The dataset used for YOLOv8 model development was created by combining publicly available apple datasets from Roboflow with self-collected images captured as shown in Figure 9. The resulting dataset consists of a total of 1,191 images, categorized into two classes: good apple, as shown in Figure 10, and bad apple, as shown in Figure 11. Initially, the available dataset from Roboflow contained images annotated with bounding boxes, which were not sufficient for the precise object shape representation needed for accurate red intensity and size analysis. Therefore, all annotations were converted to segmentation masks using the polygon annotation tool in Roboflow, as shown in Figure 12, allowing for pixel-level object detection.

Figure 9.
Self-collected images of apples.

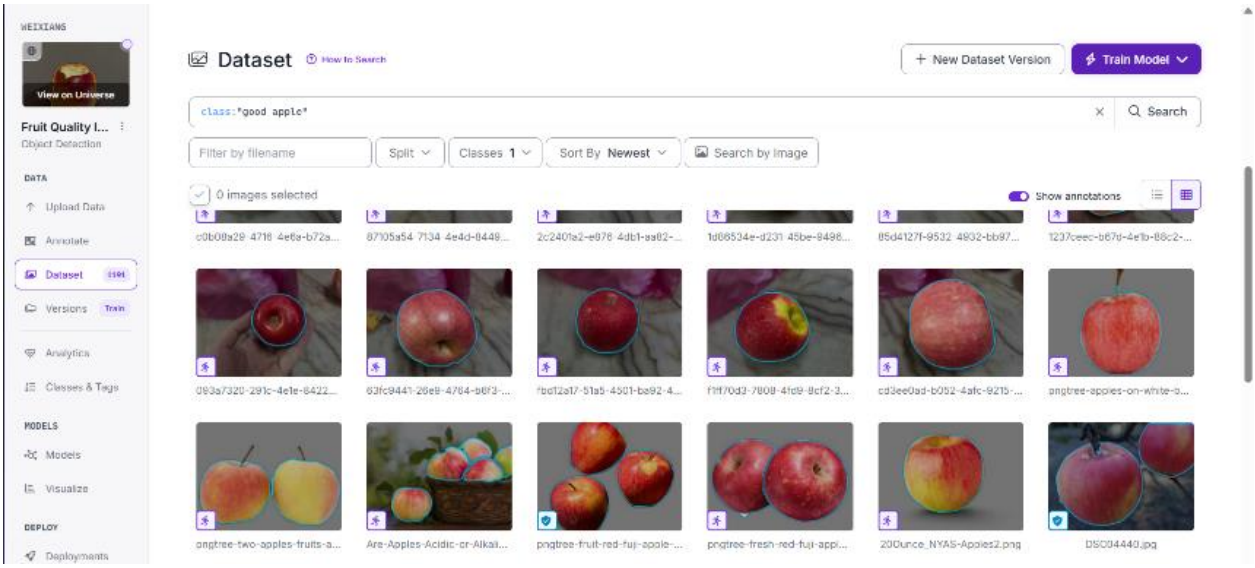

Figure 10.
Good apples class.

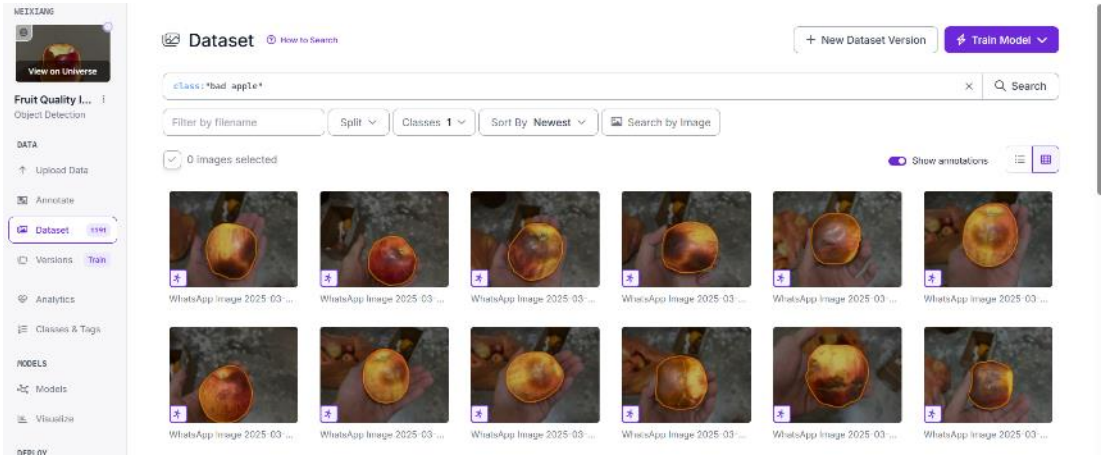

Figure 11.
Bad apples class.

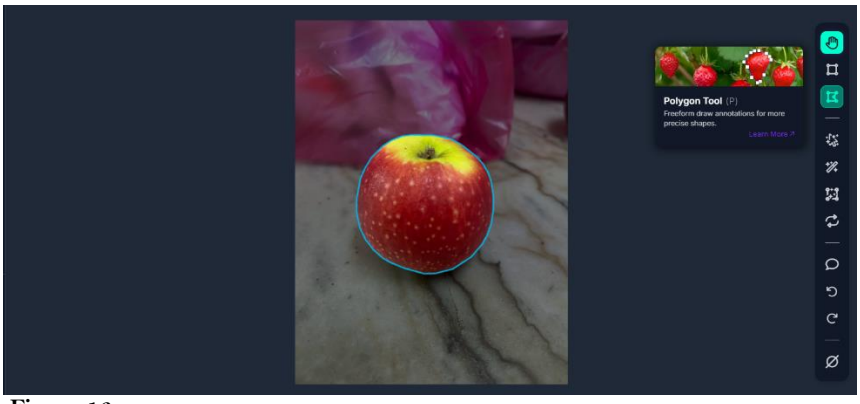

Figure 12.
Apple image annotated using the polygon tool in Roboflow for segmentation.

Figure 13 illustrates the rule-based decision logic used for classifying apples into three categories: premium, standard, and reject. This logic is implemented in both single-sided and multiple-sided detection modes of the system. The process begins with the detection of the apple using the YOLOv8 segmentation model, which determines whether the apple is classified as good or bad. If the apple is detected as bad, it proceeds directly to the reject category after evaluating its red intensity and size, which are used for display and documentation purposes only. On the other hand, if the apple is classified as good, the system evaluates the apple's red intensity and size using OpenCV. Based on predefined threshold values, if both the red intensity and size exceed the thresholds, the apple is sorted as premium. If either of the two features falls below the required threshold, the apple is considered standard. This structured logic ensures that only apples with high red colour intensity and sufficient size are placed in the premium category, while others are categorized accordingly to maintain consistency and grading accuracy across both detection modes.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 10: 32-49, 2025
DOI: 10.55214/2576-8484.v9i10.10340

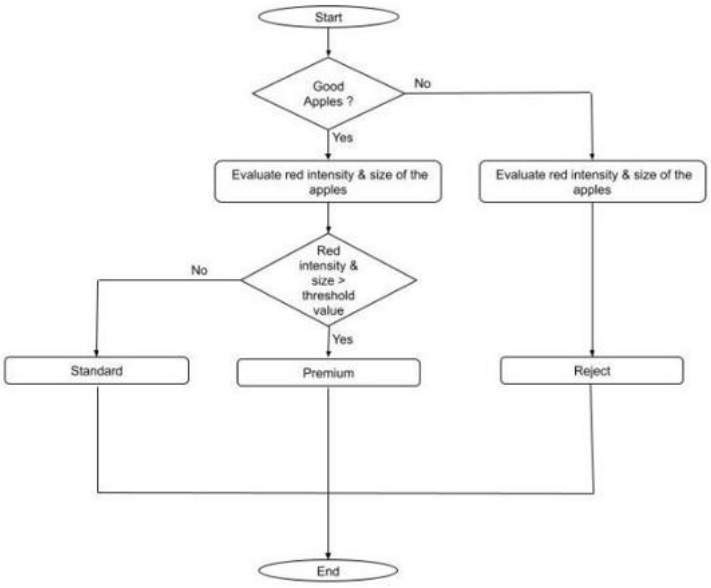

Figure 13.
Flowchart for sorting logic and classification rules.

Figure 14 illustrates the operational flow of the single-side detection mode used in the apple sorting system. The process begins with initializing a counter, $i = 0$, and setting a user-defined detection threshold (default is 5). The system continuously captures frames from the camera at regular intervals. Each captured frame is then passed to the YOLOv8 segmentation model for apple detection and analysis. When an apple is successfully detected, the system proceeds to classify its quality by analyzing red intensity and size using OpenCV. The result of each detection is stored, and the counter, i, is incremented, which means only successful apple detections are counted toward the threshold. This loop continues until the total number of valid detections reaches the threshold. At that point, the system applies majority voting to the recorded classifications to decide the final grade. Finally, the robotic arm executes the sorting operation, placing the apple into its appropriate category: premium, standard, or reject. This logic ensures accuracy by relying on multiple confirmed detections before making a sorting decision.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 10: 32-49, 2025
DOI: 10.55214/2576-8484.v9i10.10340

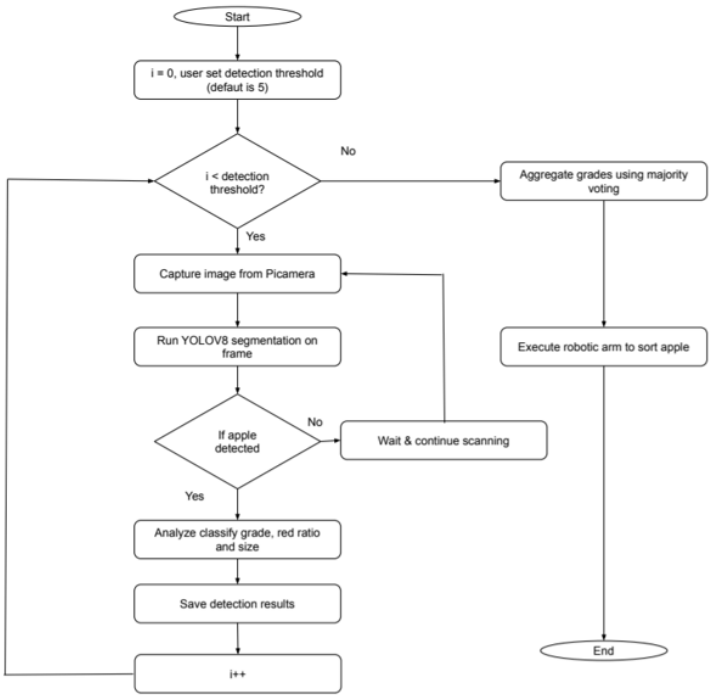

Figure 14.
Flowchart of single-side detection logic.

Figure 15 presents the detection logic for the multiple-side detection mode to enhance apple classification accuracy by capturing images from several predefined angles: 0°, 30°, 60°, 90°, 120°, 150°, and 180°. The process begins with initializing the rotation sequence and setting the initial angle to 0°. For each angle, the system captures an image using the PiCamera and runs YOLOv8 segmentation to detect the presence of an apple. When an apple is detected, its grade, red ratio, and size are analyzed. If the apple is classified as a reject, the system pauses at that angle and requires three consecutive rejection results from that same viewpoint. If this condition is met, a confirmation check follows. Once confirmed as a reject, the robotic arm immediately executes the sorting action. If the rejection condition is not confirmed, or if the apple is classified as standard or premium, the detection result is saved, and the system rotates to the next angle. If no apple is detected, the system triggers an auto-sweep rotation, rotating the platform in 10° steps between 0° and 180° and back. If an apple is detected during this sweep, the detection process resumes; otherwise, the system resets and returns to the starting angle, waiting for a new apple.

The processes of detection repeat until all rotation angles have been evaluated. At the end of the sequence, the system applies majority voting to all detection results to determine the final classification, except for the grade of rejection. Then, the robotic arm sorts the apples into the corresponding bins. This refined logic minimizes misclassification risks and improves the system's robustness, especially in identifying partially defective apples.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 10: 32-49, 2025
DOI: 10.55214/2576-8484.v9i10.10340

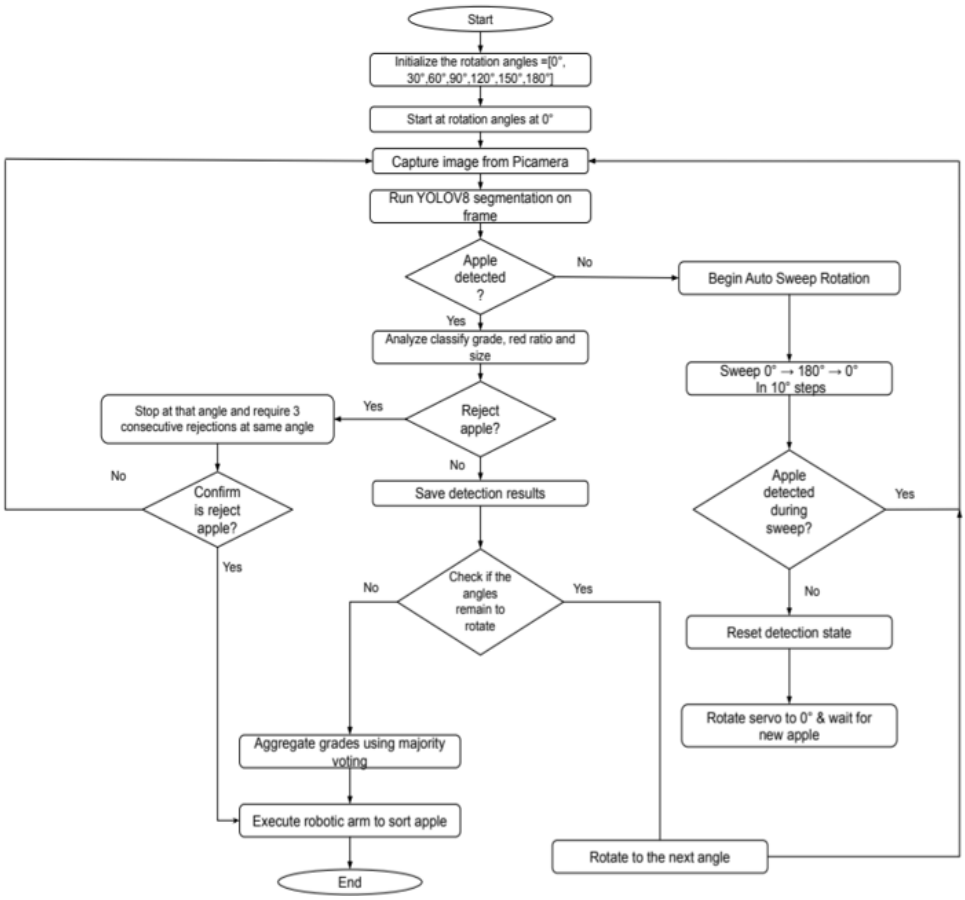

Figure 15.
Flowchart of multiple side detection logic.

The graphical user interface (GUI) of the developed sorting system, as shown in Figure 16, is designed to offer flexibility and user control during the fruit classification and sorting process. On the left sidebar, it features a logo at the top, followed by "Start" and "Stop" buttons that allow users to begin or stop the sorting operation. Below that, a fruit selection dropdown menu is provided, currently defaulted to "Apple." This list is customizable; users can add other fruits, such as oranges, by training and integrating new datasets into the system. The next section allows the user to select the capture mode: either single side or multiple sides, which determines whether the apples will be evaluated from one angle or rotated for multi-angle inspection.

Further down, the user can configure sorting criteria by enabling or disabling the checkboxes for colour and size. By default, both options are selected, meaning apples will be sorted based on both red colour intensity and size. Below this, input fields are available to adjust classification thresholds: red ratio, height (cm), and width (cm). The detection threshold setting is specifically available only in single-side mode and defines how many valid detections must be confirmed before the sorting decision is executed.

In the center of the interface, a large live video display area presents real-time footage from the camera, with detection overlays shown during operation. The live video stream shown in the GUI is meant purely for visual reference, allowing the user to monitor the camera feed in real time. However, the YOLOv8 detection does not run on every frame of the live stream. Instead, it processes the video feed periodically by capturing and analyzing individual frames at specific intervals. This method is used to reduce system load and latency, which is especially important when running on a resource-limited

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 10: 32-49, 2025
DOI: 10.55214/2576-8484.v9i10.10340

platform like the Raspberry Pi. Once a frame is captured, the YOLOv8 model is applied to that frame, and if a valid apple is detected, further analysis (colour ratio and size measurement) and classification logic are triggered. This approach ensures efficient detection while maintaining a responsive and smooth live display.

On the right side of the GUI, the detection information panel provides a summary of the apple's current classification result under GRADE, along with detailed metrics such as red ratio, size, and detection angle under DETAILS. When no apple is detected, this panel displays "N/A" and "No data available." Overall, the GUI is intuitive and adaptable, allowing users to easily configure parameters, extend the system to other fruits, and manage the sorting process interactively.

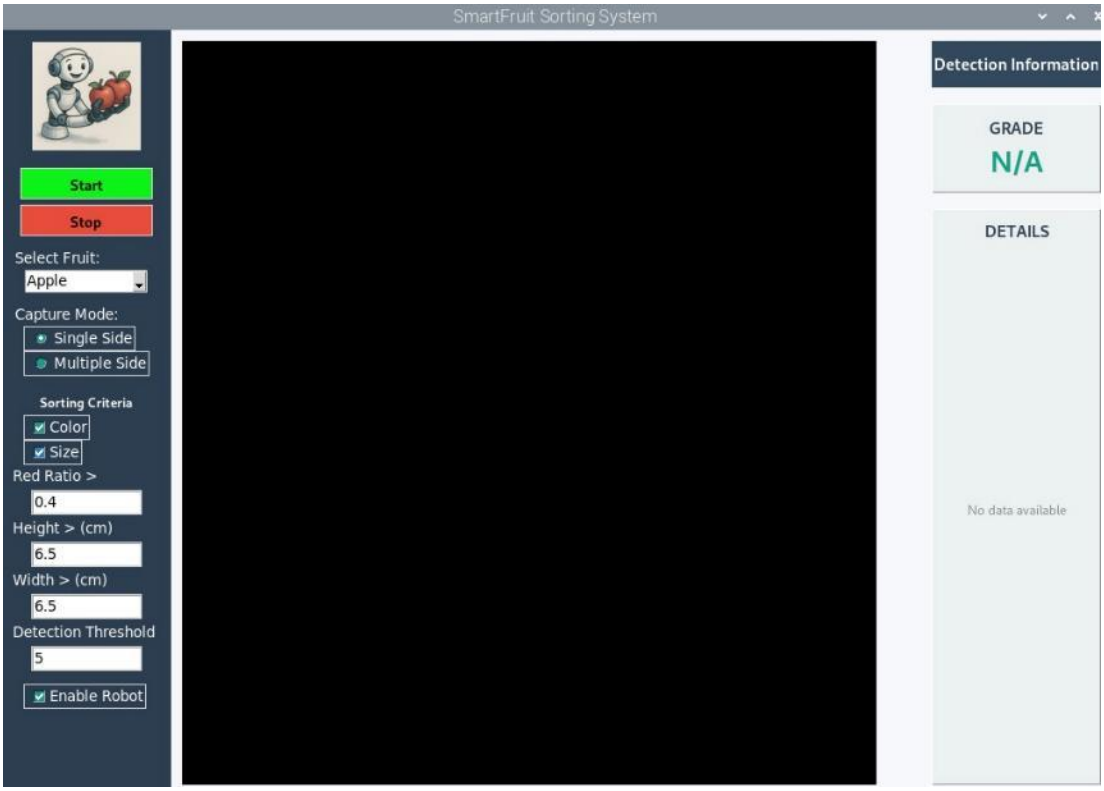

Figure 16.
GUI of the developed fruit sorting system.

2. Result and Discussion

The performance of the YOLOv8 segmentation model was quantitatively evaluated using precision, recall, and mean Average Precision (mAP) for both bad apples and good apples, as summarized in Table 1. The model achieved a precision of 94.01% for both classes, indicating that most apples labeled as "good" or "bad" were correctly identified with minimal false positives. The recall rate was 93.31%, reflecting the model's effectiveness in detecting nearly all true instances of good and bad apples, with only a small number of false negatives. Furthermore, the model achieved a high mean Average Precision (mAP) of 96.56%, confirming consistent performance across various confidence and IoU thresholds. These strong results are supported visually by the training curves in Figure 17, which show stable increases in precision and recall values over the 200 training epochs. Overall, these metrics confirm that the YOLOv8 model is highly effective and reliable for real-time apple quality classification in the proposed fruit sorting system.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 10: 32-49, 2025
DOI: 10.55214/2576-8484.v9i10.10340

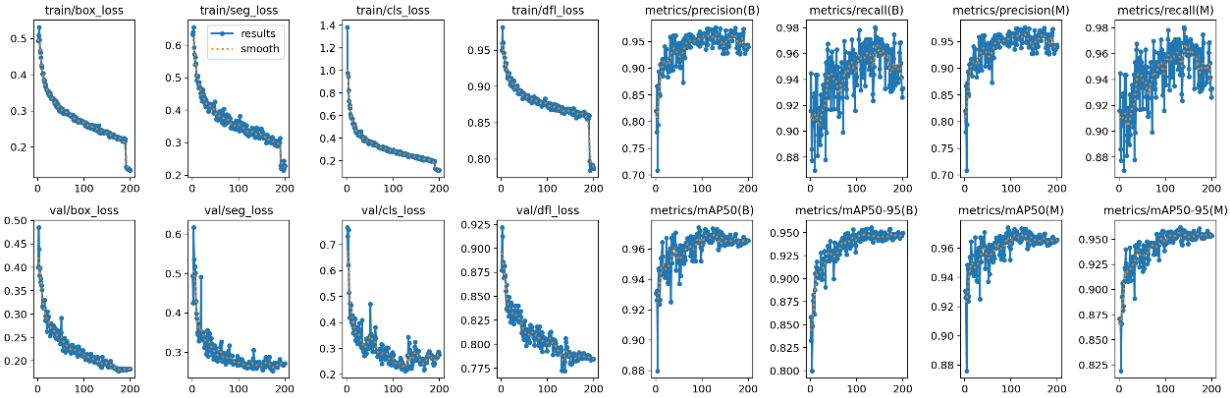

Figure 17.
YOLOv8 at 100 epoch statistic.

Table 1.
Summary of precision, recall, and mAP for apple classification.

Metric	Bad Apple (B)	Good Apple (M)
Precision (%)	94.01	94.01
Recall (%)	93.31	93.31
mAP (%)	96.56	96.56

To evaluate the classification performance of the YOLOv8 segmentation model in software testing, a total of 100 apple images were used, comprising 50 good apples and 50 bad apples. These images were captured under varying conditions, including different backgrounds, lighting intensities, and viewing angles, to test the robustness and generalization capability of the model. Despite the diversity in testing conditions, all apples were accurately classified, achieving 50 out of 50 correct classifications for both good and bad categories. The results indicate that the YOLOv8 model is highly effective and reliable in detecting apple quality at the software level. The results demonstrate the model's strong ability to generalize across different environmental factors, making it a promising solution for real-world fruit quality assessment.

Based on the classification results, the YOLOv8 segmentation model achieved perfect performance in software testing as shown in Table 1. The sensitivity value reached 100%, indicating that all actual positive samples (good apples) were correctly identified as true positives, with no false negatives observed. Similarly, the specificity was also calculated to be 100%, meaning all actual negative samples (bad apples) were correctly classified as true negatives, with no false positives recorded. Furthermore, the accuracy of the model, calculated using the combined true positive and true negative values, also resulted in a perfect score of 100%, confirming that the model correctly predicted all test cases. These outcomes clearly demonstrate the high effectiveness and reliability of the YOLOv8 model in software-based classification scenarios, particularly under the controlled conditions of this test set.

Table 2.
Sensitivity, specificity, and accuracy evaluation for single and multiple-sided hardware testing.

Operation Mode	Sensitivity	Specificity	Accuracy
Single-sided	100%	90%	97%
Multiple- side	100%	100%	100%

To further evaluate the classification performance of the system during single-side hardware testing, three key metrics were calculated: sensitivity, specificity, and accuracy. These metrics provide insight into the system's ability to correctly identify apples of different quality categories. The evaluation results are presented in Table 2. The system achieved a sensitivity of 100%, indicating that

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 10: 32-49, 2025
DOI: 10.55214/2576-8484.v9i10.10340

all 39 apples labeled as positive cases, such as premium and standard apples, were correctly identified, with no false negatives. The specificity was calculated at 90%, meaning that 19 out of 21 actual negative cases, which are reject apples, were correctly classified, while 2 samples were misclassified. The overall accuracy of the classification system was 97%, derived from correctly predicting 58 out of 60 total samples. These results confirm that the rule-based classification logic applied in the second prototype iteration is highly reliable for real-time hardware implementation using single-side detection. However, the lower specificity compared to sensitivity suggests the need for refinement in distinguishing rejected apples more effectively. This issue is addressed further in the multi-side detection phase.

To quantitatively evaluate the performance of the final prototype with multi-angle detection, standard performance metrics such as sensitivity, specificity, and accuracy were calculated, and the evaluation results are summarized in Table 2. These metrics provide insight into the system's ability to correctly classify apples across all quality categories during real-time hardware testing. The results demonstrate perfect classification performance. The system achieved a sensitivity of 100%, correctly identifying all 40 true positive cases, which are the premium and standard apples, without any false negatives. The specificity also reached 100%, as all 20 rejected apples were correctly classified, with no false positives recorded. Consequently, the overall accuracy of the system was 100%, derived from a total of 60 correct classifications out of 60 apples tested. These results confirm that the implementation of a rotating platform and multi-angle evaluation significantly enhances the reliability and robustness of the system. The high sensitivity and specificity reflect the system's strong generalization across varying fruit conditions, establishing the effectiveness of the final prototype for practical deployment in automated fruit sorting applications.

The evaluation of sorting speed under single-side detection involved recording the total time from apple detection to the completion of sorting by the robotic arm and returning to the home position across three quality categories: premium, standard, and reject. As shown in Table 3, ten tests were conducted for each category, and the results demonstrated consistent performance with only slight variations between categories. The average sorting speed for premium apples was 16.72 seconds, while standard apples averaged 17.18 seconds, and reject apples recorded an average of 17.93 seconds. The marginally longer duration for rejecting apples can be attributed to the physical placement of the reject bin, which required the robotic arm to travel a slightly longer path for placement compared to the other two bins. The overall average sorting speed across all categories was calculated to be 17.28 seconds, indicating that the system can handle apples efficiently under the constraints of single-side detection. The measurement incorporated robotic arm movement, image processing, and decision-making time.

Table 3.
Sorting speed of single-side detection.

Test No	Sorting speed (seconds)		
	Premium	**Standard**	**Reject**
1	16.11	16.28	20.24
2	16.16	16.86	18.46
3	15.84	16.88	16.29
4	15.72	16.01	17.39
5	16.43	19.06	19.09
6	16.13	16.11	15.76
7	18.31	16.13	18.11
8	16.97	19.68	17.92
9	18.04	18.44	18.51
10	17.49	16.39	17.53
Average	16.72	17.18	17.93
Total average of sorting speed	17.28		

The sorting speed can be improved by reducing the detection threshold, which in this prototype was set to 5. Lowering this threshold would allow the system to make decisions with fewer detection

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 10: 32-49, 2025
DOI: 10.55214/2576-8484.v9i10.10340

confirmations, thus shortening the overall processing time, albeit at a potential cost to classification accuracy.

The evaluation of sorting speed under multi-side detection involved rotating apples to capture multiple angles before classification. As shown in Table 4, ten tests were conducted for each apple quality category, which are premium, standard, and reject. The system demonstrated a noticeable increase in sorting time compared to single-side detection, primarily due to the additional time required for platform rotation and multi-angle evaluation. The average sorting time for premium apples was 27.65 seconds, while standard apples took slightly longer at 28.59 seconds. Interestingly, rejected apples were sorted more quickly, with an average time of 23.87 seconds. This efficiency was achieved because the system was configured with a shortcut logic: if the model detected an apple as a "Bad Apple" in three consecutive frames, it would immediately finalize the rejection decision, skip the remaining rotation steps, and trigger the robotic arm to sort the apple into the reject bin without delay. The overall average sorting speed for the multi-side detection system was 26.7 seconds, which is higher than the single-side system (17.28 seconds). However, this trade-off is justified by the improved classification accuracy obtained through better surface coverage. Further improvements in sorting speed could be achieved by reducing the rotation angle range. For example, narrowing the rotation span from 0° to 180° to a smaller sweep would minimize rotational delay while still ensuring sufficient coverage for detection.

Table 4.
Sorting speed of multiple-side detection.

Test No	Sorting speed (seconds)		
	Premium	Standard	Reject
1	26.01	27.49	28.29
2	24.92	26.81	22.57
3	27.19	33.98	14.16
4	25.98	27.01	28.07
5	26.31	27.59	28.31
6	32.99	27.52	13.58
7	26.52	32.29	23.74
8	26.15	28.31	23.77
9	32.32	27.37	28.59
10	28.10	27.57	27.61
Average	27.65	28.59	23.87
Total average of sorting speed	26.7		

Table 5 provides a comprehensive comparison between the developed system and several existing studies related to fruit classification and sorting. The proposed system utilizes YOLOv8 segmentation combined with rule-based colour and size analysis, enabling it to adapt to different fruit types through retraining, unlike the compared systems, which are generally limited to specific fruits such as tomatoes. This adaptability is a significant strength of the proposed design. Notably, the system integrates a rotating platform for multiple-angle detection, allowing for more comprehensive surface analysis compared to the single-angle detection methods employed by other systems, Hou et al. [5], Tanvashi et al. [9] and Abbas et al. [6]. This advancement contributed to the 100% classification accuracy observed in the final prototype, surpassing the accuracy rates reported by other studies, which are 99.1% and 97%. While the proposed system has slightly slower sorting times, 17.28 seconds for single-side detection and 26.7 seconds for multiple-side detection, compared to high-speed performance, it demonstrates significant improvements in the accuracy of Hou et al. [5], 2.67 seconds and Tanvashi et al. [9] 1.2 seconds. This trade-off is justified by the higher classification precision and detailed surface analysis. Additionally, both Hou et al. [5] and Tanvashi et al. [9] relied on conveyor-based systems designed for high-throughput environments, while the proposed system targets accuracy, adaptability, and cost-effectiveness for smaller-scale operations. Moreover, while Abbas et al. [6] used traditional

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 10: 32-49, 2025
DOI: 10.55214/2576-8484.v9i10.10340

MATLAB-based image processing and did not support real-time sorting. The proposed system successfully integrates real-time classification and mechanical sorting, reinforcing its strength in practical deployment.

Table 5.
Comparison with recent work

Criteria	Proposed system	Hou, et al. [5]	Tanvashi, et al. [9]	Abbas, et al. [6]
Detection Method	YOLOv8 Segmentation	YOLOv8 Detection	YOLOv5 Detection	MATLAB-based Image Processing
Classification Method	Colour & Size Analysis + AI	Colour, Ripeness, Defect Analysis	Ripeness Analysis	Colour, Texture, Defect Detection
Fruit Type Adaptability	Adaptable (Retrainable)	Limited to Tomatoes	Limited to Tomatoes	Limited (Angle Dependent)
Hardware Complexity	Moderate (Pi, Servo Motors, Camera)	Moderate (Robotic Arm, Conveyor)	Moderate (SCARA Robot)	Simple (Conveyor, Arduino)
Detection Angles	Multiple Angles (Rotating Platform)	Single Angle	Single Angle	Single Angle
Sorting Accuracy (%)	100% (Final Prototype)	99.10%	97%	Not Mentioned
Sorting time (s)	17.28 (Single-side), 26.7 (Multiple-side)	2.67 (1350 fruits/hour)	1.2	Not Mentioned
Real-time Operation	Yes	Yes	Yes	No

3. Conclusion

In this work, an automated fruit quality classifier and sorter using a robotic arm was successfully developed to address the limitations of manual fruit grading, such as human error, subjectivity, and labor dependence. The system integrates a YOLOv8 segmentation model trained to classify apples into good and bad categories, followed by further grading of good apples into premium and standard classes using OpenCV-based color intensity and size analysis. The proposed system consists of a 3DOF robotic arm and a PiCamera mounted on a rotating platform, enabling single-side and multiple-side detection. With real-time image capture and intelligent sorting capabilities, the system achieved a high classification accuracy of 97% using single-side detection and further improved to 100% accuracy under multiple-side detection. The addition of a rotating platform played a key role in enhancing robust detection by exposing all fruit surfaces for inspection, thus reducing false classifications. Furthermore, the developed GUI allowed users to select detection modes, view real-time classification results, and control sorting operations efficiently. The system demonstrated not only accurate classification but also practical sorting capabilities using cost-effective components and a compact design. In summary, the developed automated fruit sorting system offers a scalable and efficient solution for small-scale agriculture and research settings, contributing to the advancement of intelligent automation in post-harvest processes. It shows great potential for future adaptation in sorting other fruit types and integration into larger smart farming systems.

Funding:

This work is supported by the Faculty of Engineering and Technology, Multimedia University, under the FYP fund.

Transparency:

The authors confirm that the manuscript is an honest, accurate, and transparent account of the study; that no vital features of the study have been omitted; and that any discrepancies from the study as planned have been explained. This study followed all ethical practices during writing.

References
[1] A. H. U. Tumanggor, C. E. F. Tjomiadi, and M. S. M. O. S. Selvija, "Human reliability analysis on fresh fruit bunches sorting workers," *Jurnal Ilmiah Teknik Industri*, vol. 21, no. 2, pp. 289-298, 2022. https://doi.org/10.23917/jiti.v21i2.19713

[2] M. Sohan, T. Sai Ram, and C. V. Rami Reddy, "A review on YOLOv8 and its advancements. In: Jacob, I.J., Piramuthu, S., Falkowski-Gilski, P. (eds)," in *Data Intelligence and Cognitive Informatics. ICDICI 2023. Algorithms for Intelligent Systems. Springer, Singapore*, 2024, in Data Intelligence and Cognitive Informatics, pp. 529-545.

[3] G. M. Abro and K. Kumar, "Implementation of fruit grading & sorting station using digital image processing techniques," *Sir Syed University Research Journal of Engineering & Technology*, vol. 7, no. 1, pp. 19-24, 2017.

[4] M. H. Dairath *et al.*, "Computer vision-based prototype robotic picking cum grading system for fruits," *Smart Agricultural Technology*, vol. 4, p. 100210, 2023. https://doi.org/10.1016/j.atech.2023.100210

[5] L. Hou *et al.*, "Tomato sorting system based on machine vision," *Electronics*, vol. 13, no. 11, p. 2114, 2024. https://doi.org/10.3390/electronics13112114

[6] A. Abbas *et al.*, "Drones in plant disease assessment, efficient monitoring, and detection: A way forward to smart agriculture," *Agronomy*, vol. 13, no. 6, p. 1524, 2023. https://doi.org/10.3390/agronomy13061524

[7] E. V. Filho, L. Lang, M. L. Aguiar, R. Antunes, N. Pereira, and P. D. Gaspar, "Computer vision as a tool to support quality control and robotic handling of fruit: A case study," *Applied Sciences*, vol. 14, no. 21, p. 9727, 2024. https://doi.org/10.3390/app14219727

[8] S. K. Chakraborty *et al.*, "Development of an optimally designed real-time automatic citrus fruit grading–sorting machine leveraging computer vision-based adaptive deep learning model," *Engineering Applications of Artificial Intelligence*, vol. 120, p. 105826, 2023. https://doi.org/10.1016/j.engappai.2023.105826

[9] S. Tanvashi *et al.*, "Integration of computer vision and 4DOF SCARA robot arm for tomato sorting," in *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT) (pp. 1-6). IEEE*, 2024.

[10] Y. Adamu, A. Adamu, S. Kolo, and A. Nnanna, "Development of an automated fruit sorting machine using an embedded system (Arduino Mega Based)," *International Journal of Scientific & Engineering Research*, vol. 10, no. 6, p. 1099, 2019.

[11] O. O. Arjenaki, P. A. Moghaddam, and A. M. Motlagh, "Online tomato sorting based on shape, maturity, size, and surface defects using machine vision," *Turkish Journal of Agriculture and Forestry*, vol. 37, no. 1, pp. 62-68, 2013. https://doi.org/10.3906/tar-1201-10