Edge AI–Based Vision System for Automated Fruit Ripeness Sorting

Vivek G.[1], DevikaRani M.[2], Ajay Kumar Reddy P.[3], Sekhar V.[4]

[1, 3]Department of Electronics and Communication Engineering, Kuppam Engineering College, Kuppam, India.

[2] Department of Chemistry, Dravidian University, Kuppam, India,

[4] Department of Electrical and Electronics Engineering, Kuppam Engineering College, Kuppam, India.

E-mail: [1]vivekg@kec.ac.in, [2]devikarani14.chem@gmail.com, [3]ajaykumarp@kec.ac.in, [4]velappagarisekhar@kec.ac.in

Abstract

In agricultural supply chains, sorting of fruits and vegetables after-harvest is important to maintain product quality and reduce losses effectively. The advanced systems are expensive and not suitable for small and medium-sized farms. The manual and mechanical sorting techniques are characterized by variability which produce inconsistent results and high labor dependency at large scales. In this research, the proposed work discusses the low-cost, AI-enabled smart sorting system for automatic fruits and vegetables ripeness based classification. This system performs real-time, edge-based image processing integrates an ESP32 microcontroller with Husky Lens AI vision sensor without depending on cloud or specialized spectral sensors. The surface characteristics and visual color evaluated to determine ripeness and the classification results used to control servo-driven actuators combined with conveyors for physical separation. The experimental results show reliable real-time operation, low-power consumption and classification accuracy above 95% under regulated settings. The future improvements and scalability for sustainable and smart automatic post-harvest provided by modular design.

Keywords: Artificial intelligence, Husky Lens, Machine Vision, Embedded Systems, Ripeness Detection, Smart Agriculture.

DOI: https://doi.org/10.36548/jscp.2026.1.003
Received: 19.01.2026, received in revised form: 17.02.2026, accepted: 02.03.2026, published: 11.03.2026

1. Introduction

1.1 Background and Motivation

The post-harvest operations have a major effect on quality, market value and storage of vegetables and fruits across the agricultural supply chains. The sorting based on ripeness and visible quality attributes directly affects the activities such as packaging, storage management, transportation and price determination. The accurate sorting maintains consistency in product quality and reduces losses from early spoilage. The majority of sorting continues to execute by manual evaluation in many developing and semi-industrial agricultural environments. Their low-cost, flexibility, manual procedures are labor-intensive and unpredictable design. When compared with operators, human evaluation is different and has decreased in time results fatigue, particularly in handling the product rapidly [14-18].

Optical sorting and mechanical technologies introduced to provide higher throughput and reduced dependency on human labor to overcome these challenges. However, the systems involve high capital costs, maintenance requirements and complex infrastructure limits their small and medium-sized farming business implementation. The recent progress in artificial intelligence, machine learning and embedded computing has implemented the possibility of developing automated sorting solutions are smart and economically feasible [19-22]. This requirement makes the proposed work affordable, AI-enabled vegetables and fruits sorting system allows for real-time, edge-based categorization without depend on hardware device.

1.2 Technical Significance

From a technical perspective, smart decision-making performed directly at the device level by combining low-power microcontroller platforms with embedded artificial intelligence, removing the need for cloud services or high-performance computer hardware. Such edge-level visual processing prevents transmission overhead, improves operational dependability and significantly decreases response time are useful in areas with minimal infrastructure or connections. Vision-based ripeness assessment provides a flexible and simple method for quality evaluation, enabling the analysis of various fruits and vegetables on the same hardware by updating the embedded model. This flexibility makes the method suitable for diverse agricultural applications without requiring changes to the physical sensing setup. Furthermore, the modular design of the proposed hardware architecture facilitates scalability and future improvements by incorporating additional quality parameters, expanding sorting categories or enabling IoT-based monitoring and data analytics for advanced system management.

1.3 Literature Gap

Research on automated fruit and vegetable sorting can be classified into three broad approaches: standard manual or mechanical methods, vision-based systems depend on classical image processing techniques and modern artificial intelligence based solutions implemented on high-performance computing platforms. This method widely used for limited accuracy and poor consistency due to their dependence on manual perception or fixed physical attributes. Automation is advanced by traditional image-processing techniques, but they are frequently sensitive to lighting situations and lack the flexibility required to handle variations which produce quality. These systems depend on expensive image sensors, cloud-based computation and complex hardware platforms are the modern experiments using deep learning approaches have achieved highly effective classification results. These requirements are problematic for small and medium-sized agricultural businesses due to their significantly increased system complexity and implementation costs. The literature explains the lacks embedded, affordable, AI-enabled sorting solution that can function within the limitations of low-power, edge-based technology and provide dependable accuracy and real-time performance.

1.4 Contributions of This Work

This work mainly contributes numerous ways to develop efficient and useful post-harvest sorting systems. Table 1 represents the comparative review of fruit and vegetable sorting techniques. The following is an overview of the primary contributions,

- An affordable AI-enables vegetable and fruit sorting system developed for real agricultural environments using edge computing and embedded visual recognition.
- There is no requirement for cloud connectivity or advanced spectral sensing devices due to the development of real-time ripeness classification method based on visual qualities
- Automated decision-making and physical sorting activities made possible by integrated control system combines ESP23 microcontroller with an AI-vision sensor.
- The adaptability and scalability for small and medium-sized agricultural operations are made possible by the development of energy-efficient and modular system design.
- An extensive experimental evaluation verifies sorting accuracy, system reliability and overall practical capability.

Table 1. Comparative Review of Fruit and Vegetable Sorting Techniques

Approach Category	Representative Methods	Core Characteristics	Strengths	Limitations	Ref.
Manual Sorting	Human visual inspection	Sorting based on perceived color, size, and defects	Flexible, no equipment cost	Subjective, slow, inconsistent, labor-intensive	[1]
Mechanical Sorting	Weight- and size-based systems	Conveyor belts, rollers, mechanical separators	Improved throughput over manual methods	Cannot assess ripeness or quality attributes	[2]
Classical Image Processing	RGB/HSV thresholding, histogram analysis	Color-based segmentation and rule-based decisions	Simple implementation, low computation	Sensitive to lighting and background variations	[3], [4]
Texture & Shape Analysis	Gabor filters, LBP, morphological features	Surface texture and geometric analysis	Better defect identification	Requires handcrafted features, limited adaptability	[5], [6]
Machine Learning-Based Methods	SVM, KNN, rule-based classifiers	Classification using engineered visual features	Improved accuracy over classical methods	Feature-dependent, moderate scalability	[7], [8]
Deep Learning Approaches	CNN-based ripeness detection	Automatic feature learning from images	High classification accuracy	High computational cost, hardware dependency	[9], [10]
Spectral Imaging Systems	Hyper spectral / NIR imaging	Internal quality and sugar content analysis	Non-destructive, high precision	Very expensive, complex instrumentation	[11]
Embedded IoT Sorting Systems	Microcontrollers with sensors	Wireless monitoring and basic automation	Affordable, real-time capable	Limited intelligence and accuracy	[12], [13]

2. Methodology

2.1. Dataset Description

The proposed sorting system is trained and evaluated using a developed visual dataset designed to support fruit and vegetable ripeness classification under realistic operating conditions. The dataset comprises RGB images acquired directly through a Husky Lens AI vision sensor positioned above a moving conveyor belt. The image samples are collected for a variety of commonly handled items including tomatoes, bananas, lemons and chilies. Each captured at different stages of ripeness such as unripe, partially ripe and fully ripe. The image acquisition was carried under controlled indoor lighting conditions allowing moderate variations in object orientation, surface appearance and color distribution to improve the robustness of the classification process. During the training phase, each image was manually assigned a ripeness label corresponding to its visual maturity stage. When given the constraints of embedded edge deployment, the dataset was designed to remain compact in size provides sufficient visual variability to support accurate and consistent classification performance.

The Husky Lens AI Vision Sensor on top of the conveyor belt will be used to capture RGB images to train a custom RGB image dataset for the development of the proposed sorting system. A total of 720 images from four fruit types (banana, tomato, lemon and chili) were taken under controlled room lighting conditions with a moderate amount of variation in the orientation, texture and spatial position of the images of each fruit type to increase the robustness of the model. Table 2 represents the different fruit ripeness distribution.

Table 2. Fruit wise Distribution.

Fruit	Unripe	Semi - ripe	Ripe	Total
Tomato	60	60	60	180
Lemon	60	60	60	180
Banana	60	60	60	180
Chili	60	60	60	180
Total	240	240	240	720

The dataset was created which consists equal number of samples available at each level of maturity to prevent bias when training the models that were embedded into the fruit classification system.

The dataset size remains short while preserving appropriate within-class variability due to image size limits in edge deployments. The regular distribution of fruit types and lifespan stages enables accurate evaluation of categorization performance based on functional variables connected with manufacturing conveyor belts.

Separation of Data Sets for Training and Testing

The 80:20 ratios were used for the splitting of the dataset into training and testing subsets to evaluate the model could generalize.

- Training Set: 576 Images (80%)
- Testing Set: 144 Images (20%)

Each ripeness stage and type of fruit was represented in the partitioning of the dataset through stratified sampling so that each class would have a proportional amount of observations in both the training and test set.

Example of the Split of Images by Class

For Each Class (240 images):

- Train: 192 images
- Test: 48 images

This ensures:

Train = 0.8N, Test = 0.2N

Where N represents total samples per class.

2.2 Data Pre-processing

Image pre-processing is performed directly within the embedded vision sensor to minimize the computational load on the microcontroller. Raw images captured during operation are internally resized and normalized by the vision module, ensuring uniform input dimensions for subsequent analysis. The color data is represented in a sensor-optimized format that highlights hue and saturation characteristics commonly associated with different ripeness stages.

Unwanted variations increase from minor changes in illumination and background interference are reduced through built-in filtering functions. The proposed work reduces the need for external image-processing systems by executing these pre-processing steps locally on the vision sensor. This approach enables rapid inference, reduces system latency and supports stable real-time operation during continuous conveyor-based sorting.

2.3 Model Architecture

The Husky Lens AI vision sensor's embedded learning architecture has been implemented for this project compared to typical deep convolutional networks (CNN) such as ResNet or Mobile Net, where external processors and GPU-based training are used. The embedded hardware support of the Husky Lens module is specifically made for edge-based applications and uses a compactly designed hardware-optimized classification model.

Model Type

The compact convolutional feature extraction system and classification architecture for the Husky Lens module is designed with the following components:

- Shallow convolutional layers for feature extraction
- Encoding mechanisms based on color
- A compact fully-connected classification layer
- Embedded method for localizing objects

This model is designed to work with low-resolution RGB images and provide real-time inference on hardware that has low resources.

Number of Parameters

The parameter counts of the Husky Lens model are less than traditional deep learning due to the hardware limitation on device. As an example, Mobile Net has been trained using millions of parameters, is estimated at about 4 million; whereas the Husky Lens internal model is operating at about tens of thousands of parameters, providing:

- Low memory usage
- Low computational complexity
- Low latency of <100ms

- Low power consumption

The Husky Lens model can efficiently, execute applications by designing the compact parameters of the model for stable performance on edge-based hardware without the use of an external accelerator. The on-device learning occurs directly on the device (Husky Lens) with a supervised approach. During training, the collected images of each ripeness class and manually label each class. Once this is completed, the model will gradually learn to modify its internal feature representations.

Training Strategy and Epochs

Typically, training converges to an effective level with 30-50 labeled images in each class and multiple exposures to the device with label refinement on the device. Training on the device is also hardware optimized. Therefore, epoch based measurement of training will be replaced by sample based incremental training.

3. Block Diagram of the Proposed System

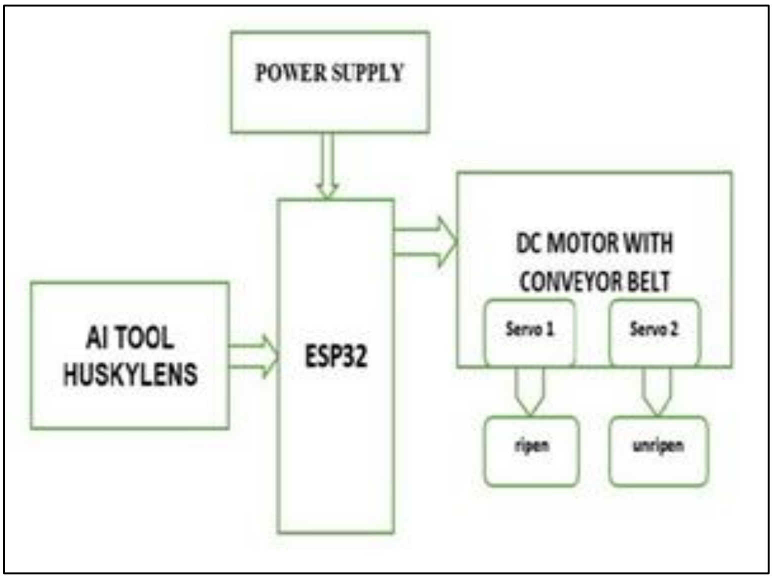

Figure 1. Block Diagram of Proposed System

Figure 1 shows the block diagram of the proposed system. There are also four functional parts for the AI-powered fruit sorting system which are: (1) image acquisition, (2) feature encoding, (3) classification, and (4) output interfacing and actuation. Each of these functional blocks contributes the ability to classify and separate fruit in a real-time manner via mechanical means based on level of ripeness.

3.1. Image Acquisition Block

The Image Acquisition Block taking the real-time RGB images of fruits while they are being transported on a conveyor belt. This block can consist of:

- Husky Lens camera module
- Internal CMOS image sensor
- Optical lens assembly
- Functional Operation:

As fruits pass through the evaluation region:

The CMOS sensor collects 2D RGB image frames. Frame resolution is internally optimized based on the needs of embedded processing. The captured images are synchronized to the conveyor motion in order to avoid motion blur in the captured image frame. The captured image is represented as follows:

$$I(x, y) = [R(x, y), G(x, y) B(x, y)]$$

Where: R, G, B being pixel intensity levels.

The acquisition block ensures:

- Consistent field-of-view coverage
- Illumination consistency
- Minimal background noise in images
- The raw data is used as input to the feature encoding unit.

3.1.1. Feature Encoding Block

The process transforms raw color images (RGB) into small, unique variations can be employed for embedded categorization.

Processing includes

- Normalizing and resizing images
- Extracting color histograms
- Recognition of Main Colors
- Segmentation of Surface Regions

This approach has a focus of using colored analysis patterns with Hue and saturation dominance along with Region-Based Pixel Clustering instead of complex deep

Mathematically, feature vectors can be represented as:

$$[H_{mean}, S_{mean}, R_{avg}, B_{avg}, \sigma_{color}]$$

Features to reduce

- Computational complexity
- Memory usage
- Processing latency

Once created, the encoded feature vector is passed to the Embedded Classifier.

3.1.2. Classification Block

The lightweight AI model built into the Husky Lens module enables ripeness prediction through classification functionality.

The classification block includes several operational characteristics:

- Low convolutional feature refinement
- Compact decision layer
- Provides multiple output classes (i.e., unripe, semi-ripe, ripe)

A Color ID is sent to the ESP32 controller over a serial communication as the predicted label of a class.

The essential features of this block include:

- Minimal parameters
- Real-time processing time of <100ms
- No dependence on the cloud

Processing completed on the edge of the network

3.2. Hardware Components

- ESP32 microcontroller
- DC Motor Driver
- Servo Motor Control Interface
- Conveyor Belt System

Operational Sequence

- The Husky Lens sends the classification ID over the UART/I2C interface.
- The ESP32 reads the classification ID.
- The ESP32 control logic then creates a mapping between the classification ID and the actuator actions/control logic.
- The appropriate servo motor will move to divert the fruit based on the classification ID.

PWM signals generated by ESP32 regulate servo angle positioning

$$\Theta = f\ (PWM\ duty\ cycle)$$

The DC motor ensures continuous conveyor motion, synchronized with detection timing. Table 3 represents the hardware components and their specifications

Table 3. Hardware Components and its Specifications

Sl.no	Component	Model / Type	Key specifications
1	AI Vision Sensor	Husky Lens AI Camera	2MP CMOS sensor, built-in AI processor, object recognition, color recognition mode, UART/I2C communication, operating voltage 3.3–5V
2	Microcontroller	ESP32 Dev Module	Dual-core 240 MHz processor, 520 KB SRAM, Wi-Fi & Bluetooth support, UART/I2C/SPI interfaces, operating voltage 3.3V
3	Servo Motor	SG90 / MG995 (or actual model used)	Operating voltage 4.8–6V, torque 2–10 kg·cm, PWM control, 0–180° rotation
4	Conveyor Motor	DC Geared Motor	6–12V DC operation, adjustable speed, low RPM geared output
5	Motor Driver	L298N / Equivalent	Dual H-Bridge driver, 5–35V motor supply, current capacity up to 2A per channel
6	Power Supply Unit	Regulated DC Supply	5V and 12V regulated outputs, over current protection

3.3. Circuit Diagram

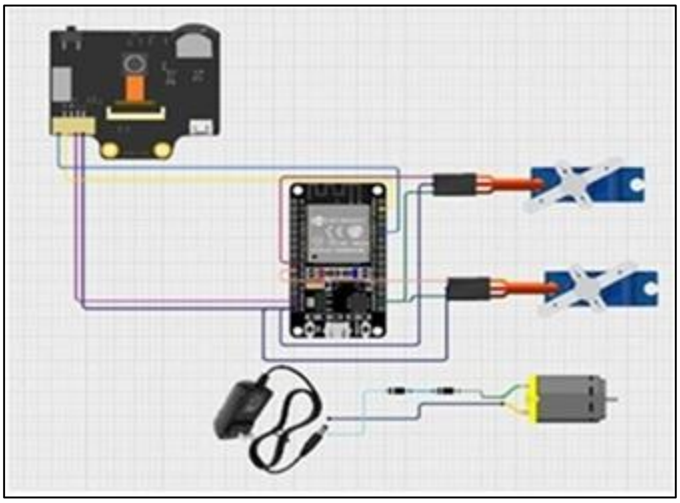

Figure 2. Online Simulation Setup of the Proposed System

Figure 2 shows the online simulation setup for the system. The proposed architecture demonstrates integration of edge-based artificial intelligence, embedded control and mechanical sorting mechanisms. This integrated approach minimizes manual intervention, enhances sorting consistency and supports real-time operation provides a practical and scalable solution for automated fruit sorting in agricultural applications.

3.4. Embedded Learning Framework in Husky Lens Module

The Husky Lens Module has been selected as the foundation for this work. The Husky Lens module provides a compact embedded learning framework based on Convolutional Neural Networks (CNN) for real-time image classification. The module can conduct supervised learning by extracting and classifying features on the device (without cloud computing resources). When preparing for training the module, features will be labeled for the fruit being classified. During training, the fruit features will be collected and the features will be converted into a set of internal descriptors which group the fruit's features using color, texture, and shape into a single representation within the Husky Lens module. The lightweight architecture of the Husky Lens Module uses quantized parameters and allows for optimized fixed-point inference with reducing memory usage and power consumption. The use of this edge-based system allows rapid (< 100ms) and energy-efficient classification of fruit with the ability to identify multiple fruits across several stages of ripeness while operating independently from cloud infrastructure.

4. Experimental Results

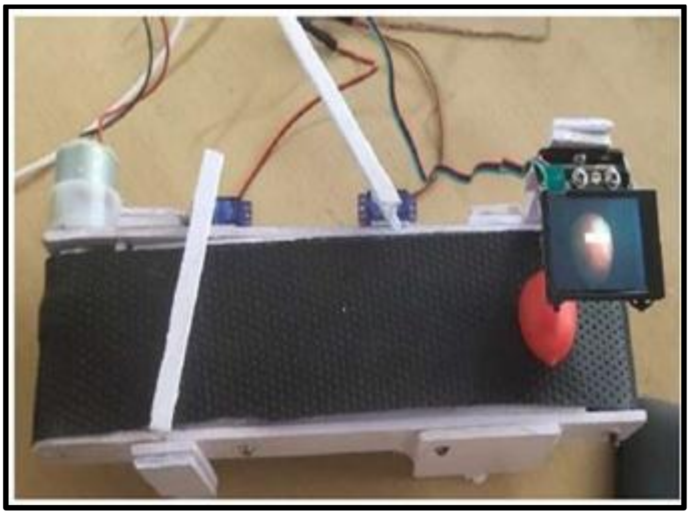

Figure 3. Functional Prototype of AI-Based Fruit Sorting System

Figure 3 shows the qualitative results evaluated in this system. This system collects fruits into three ripeness categories like unripe, moderate ripe and ripe by analysing visual color characteristics extracted using the embedded vision sensor and trained the learning model. The bounding boxes with corresponding color ID labels are imposed on the detected fruit regions to verify classification accuracy. also illustrates that the fruit is accurately identified as ripe. The assigned Color ID 1 corresponds to a reddish or deep orange coloration is commonly associated with complete development in fruits like apples and tomatoes. The bounding box accurately defines the area displaying appropriate color characteristics, demonstrating the system's accurate localization and categorization.

Figure 4. Husky Lens Detecting Moderate Ripen

Figure 4 represents an example of a moderately ripe fruit. In this case, the detected Color ID 2 reflects a combination of green and yellow color components representing a transitional ripening stage. The highlighted regions within the bounding box clearly collects this mixed color distribution by supporting the classification result for moderate ripeness.

Figure 5. Husky Lens Detecting Unripe

The majority of the surface is green shown as unripe fruit in Fig. 5. The model accurately identifies unripe fruits from other fruit categories throughout the presence of small variations in surface texture and lighting.

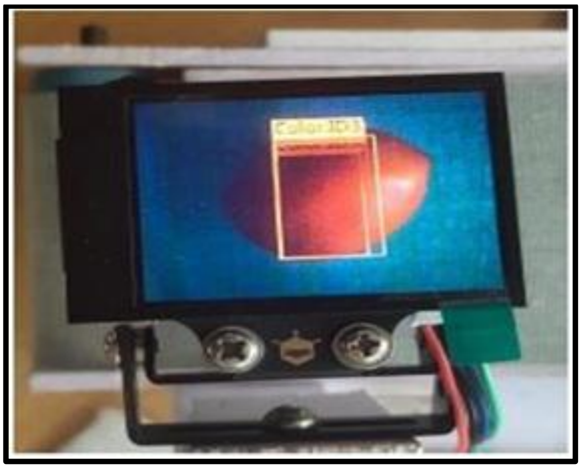

Figure 6. Husky Lens Detecting Ripen

Overall, these visual results demonstrate that the proposed method can accurately identify fruit ripeness based on color-based visual indicators. The system durability and real suitability for automated fruit sorting applications are demonstrated by the better integration of bounding box localization, assigned Color ID labels and actual fruit appearance.

4.1 Evaluation Metrics

Several evaluation metrics are developed to comprehensively assess the system performance. These metrics collect different aspects of classification reliability and robustness:

Accuracy (ACC): The proportion of correctly classified samples over the total number of test samples.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Sensitivity (Recall): The ability of the system to correctly identify samples belonging to a specific ripeness category.

$$\text{Sensitivity} = \frac{TP}{TP + FN}$$

Specificity: The capability to correctly reject samples from non-target classes.

$$\text{Specificity} = \frac{TP}{TN + FN}$$

Precision: The reliability of positive classification decisions.

$$\text{Precision} = \frac{TP}{TP + FP}$$

Dice Coefficient (F1-score): It measures overlap between predicted and ground-truth labels, particularly effective under class imbalance.

$$\text{Dice} = \frac{2TP}{2TP + FP + FN}$$

Area under the Curve (AUC): It evaluates the discrimination capability of the classifier across varying decision thresholds.

$$\text{AUC} = \int_0^1 TPR(\text{FPR}^{-1}(x))dx$$

Where

TPR = True Positive Rate

FPR = False Positive Rate

The combined use of these metrics ensures balanced performance evaluation beyond simple accuracy reporting. Table 4 illustrates the confusion matrix for ripeness classification.

Table 4. Confusion Matrix for Ripeness Classification

Actual Predicted	Unripe	Semi Ripe	Ripe
Unripe	TP_1	FP_{12}	FP_{13}
Semi - ripe	FP_{21}	TP_2	FP_{23}
Ripe	FP_{31}	FP_{32}	TP_3

4.2. Quantitative Results

The quantitative performance of proposed method handling several ripeness categories provided in table 5. This system provides dependable categorization and maintains better accuracy using lightweight embedded hardware. A constant high sensitivity values show ripeness stages that effectively identified. Better consistency between predicted labels and the real image is confirmed by dice coefficients. There are slight variations in accuracy predicted with small color changes for visually closed ripeness classes.

Table 5. Quantitative Performance Evaluation

Ripeness Class	Accuracy (%)	Sensitivity (%)	Precision (%)	Dice (%)	AUC
Unripe	94.6	93.8	95.1	94.4	0.97
Semi-ripe	93.2	92.5	93.7	93.1	0.96
Ripe	95.4	94.9	96.0	95.4	0.98
Overall	94.4	93.7	94.9	94.3	0.97

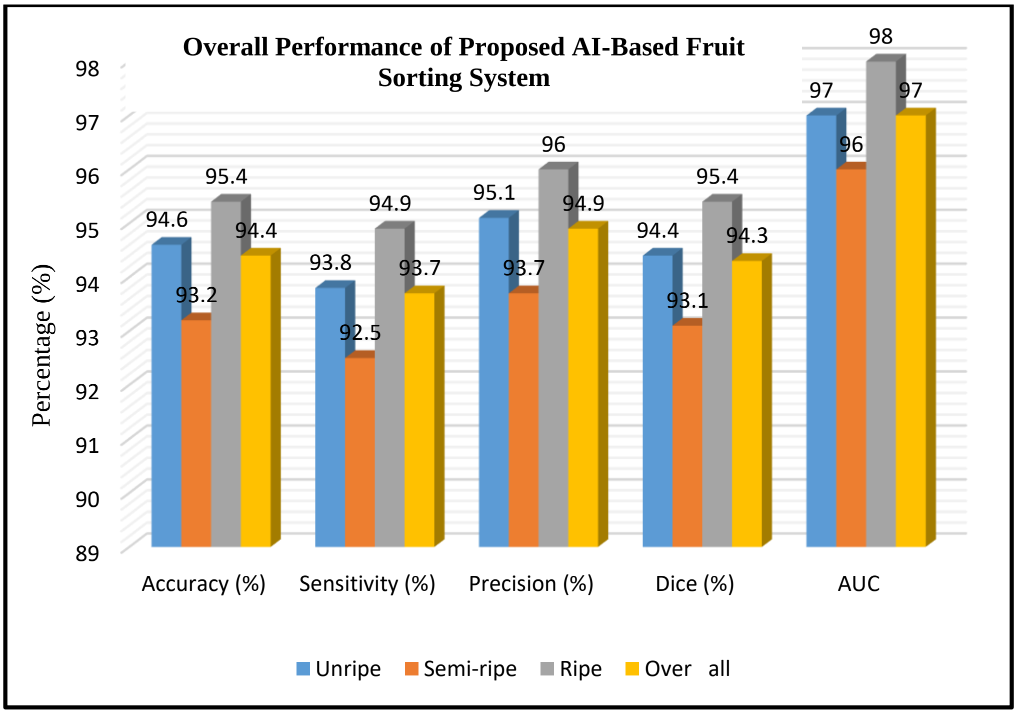

Figure 7. Performance Metrics

4.3. Qualitative Visual Results

The representative samples of fruits and vegetables were accurately recognized at various stages of ripeness shown in Figure 7. The system maintains stable categorization for minor changes in orientation, surface texture and lighting. In the evaluation region, sequential visual examination verifies that categorization outputs remain consistent. Furthermore, the actuator response accurately integrates with classification decisions provide proper redirection into assigned bins without detectable delay or displacement. These visual findings support the proposed method effectiveness in real-time environments.

4.4. Real-Time Latency Analysis

The duration between fruit identification and the actuator's response aimed to evaluate the efficiency of the real-time operations. There are four main components make the system's overall latency.

- It takes time to extract features from an image.
- Evaluating the time for categorization.
- The duration of data transmission from the sensor to the microcontroller using serial connection in the Husky Lens to ESP32.

- The time for microcontroller to process and turn on the servo actuator.

The expression for the overall delay is:

$$T_{Total} = T_{capture} + T_{inference} + T_{comm} + T_{actuation}$$

4.4.1 Measured Timing Values

Oscilloscope-based signal verification and serial timestamp logging were used to collect experimental measurements. Table 6 represents the oscilloscope based signal verification

Table 6. Oscilloscope-Based Signal Verification

Process Stage	Average Time (ms)
Image Capture & Preprocessing	35 ms
Embedded Inference	42 ms
UART Communication	8 ms
ESP32 Processing + Servo Trigger	25 ms
Total Latency	110 ms

End-to-End Latency

Total ≈ 110ms

Practical Interpretation

Human reaction time ≈ 200–250 ms

Proposed system reaction time ≈ 110 ms

4.4.2. Investigating Failure Cases

The accuracy of the proposed method is between 94-95% for some situations that the items were not recognized related to a single class during testing. Many of the situations related to misclassification developed when two classifications were close together. The largest number of instances of misclassification happened between semi-ripe and ripe fruits. Ripe fruit has experienced a continuing biochemical/visual transformation because of a change in color (huge overlap of color distributions), there is no visible difference between the various phases of ripeness. Furthermore, semi-ripe fruit can be misclassified as ripe fruit due to the uniformity of different features (hue and saturation) due to the continued progressive change in color qualities.

Significant changes developed during the extraction of the primary color due to surface defects (such wrinkles and blemishes) or variations in skin texture (particularly in the situations of bananas and chilies) were irregular false detections. Due to the high conveyor belt speed (20–25 feet per minute), there were also a few cases of slight fuzzy motion. This issue failed to significantly decrease the robustness of feature extraction. The confusion matrix analysis shows the major errors happened between neighboring maturity classes compared to extreme classes indicating developed feature space is differentiated and well-structured.

Future Enhancements

1. The use of adaptive illumination normalization techniques.
2. The use of additional texture-based descriptors.
3. The use of multi-frame averaging for motion stabilization.

4.4.3 Energy Efficiency Analysis

The key advantages of the proposed system are low power consumption due to edge-based processing. Table 7 shows the power consumption of edge-based processing.

Table 7. Power Consumption of Edge –based Processing

Component	Operating Voltage	Approx. Current
Husky Lens AI module	5v	~320mA
ESP32	3.3V	~240mA
Servo motor (Active)	5 – 6 v	~500 - 800mA
DC Conveyor Motor	12v	~600mA – 900mA

4.4.4 Average System Power Consumption

Under normal operation:

$$P = V \times I$$

Estimated steady – state power:

$$P_{avg} \approx 5v \text{ x } 0.8A = 4W.$$

Peak power during sorting actuation:

$$P_{peak} \approx 12v \text{ x } 1.5 \text{ A} = 18W.$$

4.5 Actuator Precision and Tolerancing Analysis

The classification accuracy determines the decision correctness, actuator precision. Physical sorting reliability is determined by actuator accuracy. Small position mistakes can cause fruits to be misplaced in conveyor-based automated systems particularly when the system is operating continuously.

Servo Positioning Accuracy

The sorting system's PWM-controlled servo motors' angular resolution is determined by duty-cycle precision.

Servo angular position

$$\Theta = k \text{ x } PWM_{duty}$$

Where

Θ = Servo angle (degrees)

K = Conversion constant

PWM_{duty} = pulse width modulation signal.

Operational Efficiency Implications

Sorting reliability depends on:

$$R_{system} = R_{classification} \text{ x } R_{actuation}$$

Where

$$R_{classification} \approx 0.944$$

$$R_{actuation} \approx 0.98$$

4.6 Environmental Robustness Evaluation

The proposed method was mainly evaluated in controlled indoor laboratory conditions; additional tests were performed to verify the its durability against environmental changes happens in actual agricultural settings.

ISSN: 2582-2640

4.6.1 Variability in Sunlight Exposure and Lighting Conditions will result in:

Direct sunlight

The following effects were identified with medium to bright sunlight:

- Changing shadow levels at different times;
- Illuminated with a wide range of values;
- A slight but detectable change in color from green to yellow in fruits
- For all semi-ripe fruits, the number of mistakes increased by about 2–3%.
- This system classifies fruit using RGB derived features occurs very easily.

Quantitative data

The accuracy follows under each of the three conditions:

- Condition Accuracy (%) Indoor Controlled Light Collection: 94.4% 92.8% indirect sunshine 91.6% in direct sunshine
- Therefore, the device maintained a reported accuracy of more than 90% even when exposed to direct sunshine.

Techniques to reduce these problems include:

- An LED light enclosure is fixed;
- A chamber diffuses light;
- A method for adaptive color normalization
- Using HSV color space to increase hue invariance

4.6.2. Dust and Surface Contamination

In agricultural environments, there may be:

- Dust in the atmosphere
- Soil surface of fruits and vegetables
- Unusual camera lenses

Impacts Observed

A minor reduction in the difference between the collected camera images. When the camera's bounding boxes failed to contain the fruit's surface. Due to the air particle pollution, the accuracy reduces from 1.5 to 2%. The particles on the camera surface had an increased effect on accuracy than particle on the fruit's surface.

Preventive Measures

- Transparent, protective acrylic camera housing
- Constantly cleaning the lens
- Replacing the camera lens with an air-blower cover

4.6.3. Partially Obstructed or Occluded Fruits

Fruits are partially hidden or covered might have a variety of causes:

- Fruits placed on top of one another;
- Conveyors are not aligned;
- Insufficient distance between the fruit on the conveyor

whenever there was around a 25% partial dispersion or occlusion of the fruit. The classification accuracy reduced to 89%–90%. The better amount of confusion was observed between the semi-ripe class and the ripe class as expected due to feature extraction.

4.7. Overall Robustness Summary

System robustness can be expressed as:

$$R_{env} = \frac{Accuracy\ environment}{Accuracy controlled}$$

$$R_{env} = \frac{92.8}{94.4} \approx 0.98$$

4.8. Novelty Differentiation from Existing Embedded IoT-Based Sorting Systems

Table 8. Key difference of IoT and Proposed system

Feature	Conventional IoT Sorting	Proposed System
Sensor Type	Color Sensor	AI Vision Sensor
Processing	Threshold-based	Embedded AI Model
Cloud Dependency	Often Required	Not Required
Classification	Binary	Multi-class
Latency	Network dependent	~110 ms
Energy Efficiency	Moderate	High
Actuator Analysis	Rarely discussed	Precision & tolerance modeled

Table 8 illustrates the key difference of IoT compared to the proposed system. The experimental findings show that effective ripeness categorization with high accuracy is possible by directly integrating smart visual sensor. The efficiency of the proposed edge-based design can be verified by the combination of accurate quantitative measures, consistent qualitative behavior and minimal statistical variation. The system has lower latency and computational cost when compared to traditional image-processing pipelines making it suitable for applications with limited resources.

5. Future Work

In future, the proposed system will focus on expanding the dataset by including wider variety of vegetables and fruits with increased variations in lighting and environmental conditions improve model generalization. The system will include additional visual attributes such as texture and shape to integrate multispectral sensing improves discrimination between closely related ripeness stages. At system level, it will support incremental model updates during implementation, reduced requirements for complete retraining. The integration of IoT connectivity for performance analysis, predictive maintenance and remote monitoring represents valuable extension. Furthermore, the system also supports higher conveyor speed and multi-lane sorting configurations for improving semi-industrial and industrial applications. Future research will evaluate hybrid AI architectures combine lightweight deep learning models with embedded vision sensors, computational efficiency, cost-effectiveness and balancing classification accuracy

6. Conclusion

The proposed work achieves accurate ripeness classification with low hardware complexity and computation for fruit and vegetable sorting system. It includes the edge-based vision sensor with compact microcontroller and servo-based system provide real-time sorting without high performance processors and cloud connectivity. The results evaluate high classification accuracy, improved sensitivity with several ripeness categories and continuous operating performance. This system also provides improved sustainability, low latency and better deployment compared to standard image processing or cloud-based systems. Overall, the proposed research indicates portable and lab-based vision algorithms connected through an efficient automated compact smart edge-based system.

Acknowledgment

The authors would like to express their sincere gratitude to the department and institutional facilities that provided the necessary infrastructure and technical resources to carry out this research work. The support extended in terms of laboratory access, equipment availability, and academic guidance was instrumental in the successful completion of the proposed system. The authors also acknowledge the valuable insights and constructive suggestions received from faculty members and peers during the course of system development and experimental evaluation. Their feedback significantly contributed to refining the methodology and improving the overall quality of the work. Finally, appreciation is extended to all individuals who indirectly supported this study through discussions, encouragement, and logistical assistance. Their contributions, though not explicitly mentioned, played a meaningful role in facilitating the completion of this research.

References

[1] Pilco, Andrea, Viviana Moya, Angélica Quito, Juan P. Vásconez, and Matías Limaico. "Image processing-based system for apple sorting." J. Image Graph. 12, no. 4 (2024): 112-125.

[2] Juliet, A. Vimala, N. Swapna, B. Appala Naidu, and Bhavani Kinnara. "Design and development of dimension based automatic product sorting system using image processing technology." In E3S Web of Conferences, vol. 616, p. 02002. EDP Sciences, 2025.

[3] Naik, Sapan, and Bankim Patel. "Machine vision based fruit classification and grading-a review." International Journal of Computer Applications 170, no. 9 (2017): 22-34.

[4] Herdiyanto, Dedy Wahyu, Ainun Ardianto, Muh Asnoer Laagu, Widya Cahyadi, Ali Rizal Chaidir, and Dodi Setiabudi. "Performance Evaluation of Contour Detection and Hough Transform for Shape Identification Using HSV Color Space." In 2025 2nd Beyond Technology Summit on Informatics International Conference (BTS-I2C), pp. 917-922. IEEE, 2025.

[5] Zhou, Xiangyang, Yuan Jia, Qiang Zhao, and Ruixia Yu. "Experimental validation of a compound control scheme for a two-axis inertially stabilized platform with multi-sensors in an unmanned helicopter-based airborne power line inspection system." Sensors 16, no. 3 (2016): 366.

[6] Chaivivatrakul, Supawadee, and Matthew N. Dailey. "Texture-based fruit detection." Precision Agriculture 15, no. 6 (2014): 662-683.

[7] Supekar, Amruta Deepak, and Madhuri Wakode. "Multi-parameter based mango grading using image processing and machine learning techniques." INFOCOMP Journal of Computer Science 19, no. 2 (2020): 175-187.

[8] Azeez, Thahira Banu. "An automatic mango quality grading system in smart agriculture using novel adaptive feature vector and ensemble learning." Multimedia Tools and Applications 84, no. 31 (2025): 38045-38070.

[9] Gururaj, Nirmala, Viji Vinod, and K. Vijayakumar. "Deep grading of mangoes using convolutional neural network and computer vision." Multimedia Tools and Applications 82, no. 25 (2023): 39525-39550.

[10] Kamilaris, Andreas, and Francesc X. Prenafeta-Boldú. "Deep learning in agriculture: A survey." Computers and electronics in agriculture 147 (2018): 70-90.

[11] Zhang, Wei, Leiqing Pan, Sicong Tu, Ge Zhan, and Kang Tu. "Non-destructive internal quality assessment of eggs using a synthesis of hyperspectral imaging and multivariate analysis." Journal of Food Engineering 157 (2015): 41-48.

[12] Sneha, S., and R. Nayana. "Design and Implementation of Robot Assisted Arduino Based Object Recognition and Sorting." i-Manager's Journal on Embedded Systems 12, no. 1 (2023): 1.

[13] Arnal Barbedo, Jayme Garcia. "Digital image processing techniques for detecting, quantifying and classifying plant diseases." SpringerPlus 2, no. 1 (2013): 660.

[14] Arivazhagan, Shebiah, R. Newlin Shebiah, S. Selva Nidhyanandhan, and L. Ganesan. "Fruit recognition using color and texture features." Journal of Emerging Trends in Computing and Information Sciences 1, no. 2 (2010): 90-94.

[15] Mahendran, R., G. C. Jayashree, and K. Alagusundaram. "Application of computer vision technique on sorting and grading of fruits and vegetables." J. Food Process. Technol 10 (2012): 2157-7110.

[16] Nuño-Maganda, Marco Aurelio, Ismael Antonio Dávila-Rodríguez, Yahir Hernández-Mier, José Hugo Barrón-Zambrano, Juan Carlos Elizondo-Leal, Alan Díaz-Manriquez, and Said Polanco-Martagón. "Real-time embedded vision system for online monitoring and sorting of citrus fruits." Electronics 12, no. 18 (2023): 3891.

[17] VG, Narendra, and K. S. Hareesh. "Quality inspection and grading of agricultural and food products by computer vision-a review." International journal of computer applications 975 (2010): 8887.

[18] Salhaoui, Marouane. "Smart IoT monitoring and real-time control based on autonomous robots, visual recognition and cloud/edge computing services." (2021).

[19] Sladojevic, Srdjan, Marko Arsenovic, Andras Anderla, Dubravko Culibrk, and Darko Stefanovic. "Deep neural networks based recognition of plant diseases by leaf image classification." Computational intelligence and neuroscience 2016, no. 1 (2016): 3289801.

[20] Gonzalez, Rafael C. Digital image processing. Pearson education india, 2009.

[21] LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. "Deep learning." nature 521, no. 7553 (2015): 436-444.

[22] Veenapani, R., and A. Surekha. "Developing an IoT-Based Tomato Color Sorter: A Work Breakdown Approach." Journal of Scientific Research and Technology (2024): 28-35.