ASEAN
Journal of Scientific and Technological Reports
Online ISSN:2773-8752

Research article

Design and Development of a Lime Fruit Grading and Sorting Machine Using the AIoT Approach

Nuth Otanasap[1*], Siriwich Tadsuan[1], Veerapong Kanchanawongkul[1], Sutamet Ridcham[1], and Chanintorn Chalermsuk[2]

[1] Faculty of Engineering, Southeast Asia University, Bangkok, 10160, Thailand
[5] Faculty of Arts and Sciences, Southeast Asia University, Bangkok, 10160, Thailand
* Correspondence: nuto@sau.ac.th

Citation:
Otanasap, N. ; Tadsuan, S. ; Ridcham, S. ; Chalermsuk, C. Design and development of a lime fruit grading and sorting machine using the AIoT approach. *ASEAN J. Sci. Tech. Report.* **2026** *29*(12), e264472. https://doi.org/10.55164/ajstr.v29i12.264472

Article history:
Received: March 22, 2026
Revised: August 22, 2026
Accepted: October 3, 2026
Available online: November 21, 2026

Publisher's Note:
This article is published and distributed under the terms of the Thaksin University.

Abstract: Inspired by the operational inconsistencies and human fatigue associated with manual lime grading, this research presents an automated sorting system that integrates Artificial Intelligence (AI) and the Internet of Things (IoT) into a single edge computing platform. The primary objective is to replace subjective manual assessments with a high-speed, standardized system employing computer vision technology to enhance post-harvest agricultural productivity. The methodology consists of training a lightweight YOLOv8n object detection model on a custom dataset of 7,621 images and subsequently deploying it on a Raspberry Pi 5 hardware platform. This edge device is equipped with specialized servo drives and an IoT-based Flutter application for real-time remote yield monitoring. Extensive trials conducted on an active conveyor belt system demonstrated an overall grading accuracy of 96.86%, with a 100% success rate in distinguishing between high-quality and defective limes. While minor spatial variations in the range of 1–2 mm were observed during size evaluation, the empirical results confirm the high reliability and industrial viability of the developed machine.

Keywords: Computer vision; deep learning; YOLOv8; precision agriculture; AIoT

1. Introduction

In the multinational agricultural sector, citrus fruits such as oranges, limes, and lemons form the basis for trade and food security, securing the means of living for millions of people in processing and distribution [1]. However, as the industry faces demands for higher quality and higher yields, traditional manual post-harvest treatment methods are no longer sufficient [2]. Although the traditional method of classifying and sorting fruit has been used for generations, it has native limitations due to irregularities, human fatigue, and limited processing rates, frequently resulting in lower grades and notable financial losses along the supply chain [2].

To address these problems, the need for accurate agriculture has given rise to transformative technologies, especially artificial intelligence (AI) and computer vision [3, 4]. These advances have enabled computerized execution of quality assessment, a key, labor-intensive task [4, 5]. At the core of this specialized transformation in technology is the further development of the "You Only Look Once" (YOLO) object recognition model [3]. Since its introduction, YOLO has redefined real-time sensing. The latest version, YOLOv8, features a

sophisticated, pinpoint-accuracy architecture that attains a balance of high accuracy with the processing speed required for immediate agricultural applications [3].

Although AI has become highly accessible, many farmers, especially smallholders or farmers with constrained resources, still face a consequential digital divide due to the high cost and sophistication of industrial image processing systems [4, 6]. This challenge requires the development of a cost-effective, portable, and efficient procedure that runs on end devices such as the Raspberry Pi [4, 6]. Recent studies have shown that compact models can detect citrus diseases and defects, even amidst intricate natural atmospheres with varying surroundings and lighting environments [4, 7].

This research builds on this fundamental familiarity by presenting the design and development of a lime sorting and classification system using artificial intelligence (AI). The aim of this project is to provide a solid solution for compartmentalizing limes based on size and surface quality by integrating a specially trained YOLOv8 model into a hardware system consisting of a Raspberry Pi 5, an Arduino R3, and a conveyor belt mechanism. The system is further enhanced by a mobile real-time application developed with Flutter, which enables centralized data monitoring. In brief, this work contributes to current efforts in the field of precision agribusiness by providing applicable and highly precise tools that diminish human error and add significant value to agricultural products.

2. Related Works

The global citrus industry is an important economic engine, producing approximately 158 million tons annually [1]. Although this sector is important, it faces major challenges in the area of post-harvest processing. Manual selection of fruit is still the predominant practice, but it has numerous shortcomings. Human decisions are naturally biased and tend to be inconsistent due to fatigue and surrounding factors, regularly producing lower-quality collaboration and economic losses [2, 6]. To overcome these impediments, researchers are increasingly relying on automation and using rapid advances in precision agriculture and computer vision to develop quality standards [3, 4]. The fruit sorting technology industry has evolved from customary image processing, which required time-consuming, manual feature extraction, to today's age of deep learning [8].

This development is a prime example of the progress of Convolutional Neural Networks (CNNs), which automatically comprehend intricate visual patterns such as textures, shapes, and colors [6, 8]. An important step in this development is the adoption of the YOLO principle (You Only Look Once). By treating object recognition as a one-stage regression problem, YOLO achieves significantly superior real-time performance in comparison to earlier two-stage methods such as R-CNN [9]. The architecture was significantly improved through various experiments, resulting in YOLOv8 being a versatile, anchor-free model that continues to maintain stable recognition precision while guaranteeing the computing speeds required for industrial applications [3, 9].

Specific applications in orange sorting have demonstrated high accuracy that can be achieved with this novel architecture. The “FruitVision” system uses an optimized MobileNetV3 infrastructure and achieves an impressive accuracy of 99.12% in classifying limes [8]. Furthermore, in specialized studies on the freshness of sweet limes, YOLOv8 was used to differentiate between good and rotten fruit. A mean performance (mAP) of 92% was achieved [6]. Exceeding simple classification, innovative models such as YOLOv8-Scm have been developed to support high-resolution experiments, for example, the classification of five different degrees of sun damage to citrus fruits in complex natural environments [7]. The study underlines deep learning's ability to spot subtle surface features that are often missed during conventional manual inspection [1, 7].

The transition to edge AI highlights a clear trend towards better access to these tools for smallholder farmers with limited resources [4]. By using resource-efficient devices like the Raspberry Pi, researchers have developed a portable and cost-effective system that allows real-time diagnostics without a constant cloud connection [2, 6, 11]. To further improve reliability, some systems use robot arms and rotating platforms to support multi-angle detection [2]. This method secures thorough surface coverage and considerably reduces classification mistakes compared to single-view inspection methods [2].

In spite of these successes, AI-assisted sorting of fruit tree debris remains a difficult task due to the complex background, changing light conditions, and obscuration by the fruit [3, 7]. Recent evaluations of

modern models, such as YOLO11, show that while these models are better at feature separation, the ability to perfectly distinguish targets such as green, unripe fruit from thick leaves continues as a key objective of current research [3]. Through cooperative efforts, these prototypes are further improved using processes such as transfer learning along with hyperparameter optimization, with the aim of closing the gap between experimental prototypes and robust, field-ready agricultural tools [1].

3. Materials and Methods

The evolution of artificial intelligence (AI) models enables the core calculations in the lime sorting system and converts raw image data into operable sorting commands. To achieve the project goal of a highly accurate assessment, this approach concentrates on the implementation of the You Only Look Once (YOLO) Version 8 (YOLOv8) architecture. This model was chosen because of its outstanding ability to strike a balance between high-speed processing and the required accuracy to detect subtle surface variations and calculate fruit size in real time in farming environments.

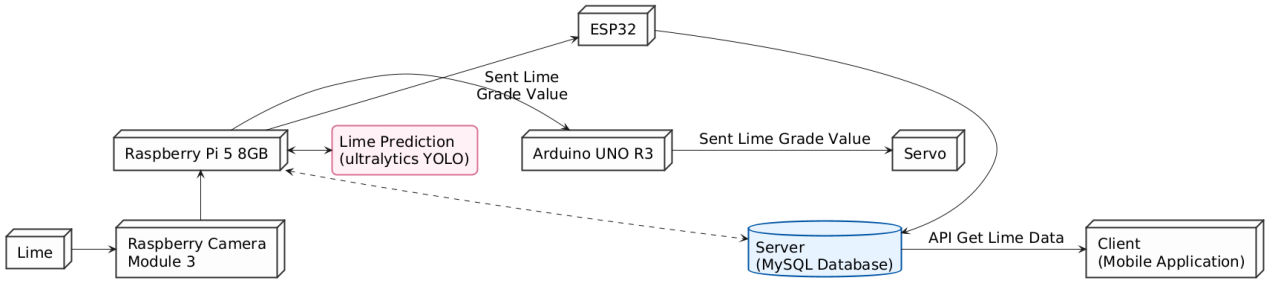

Figure 1. System framework and operational data flow of the proposed AIoT-based lime sorting machine. (Generated using PlantUML).

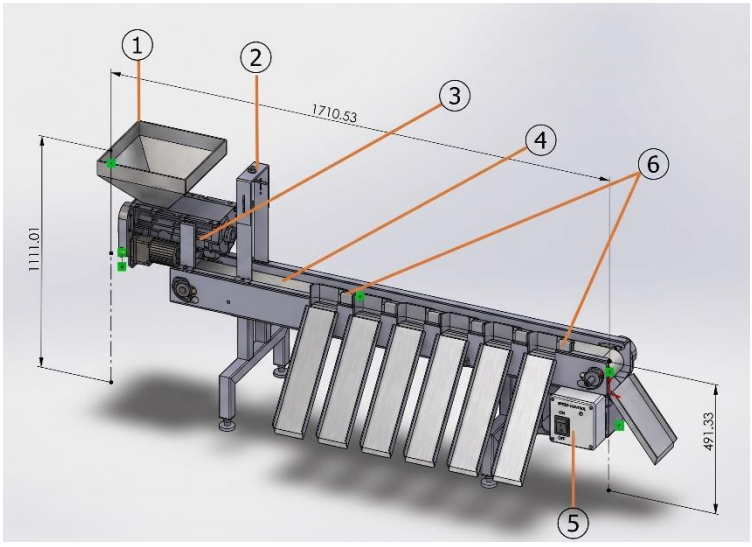

Figure 2. 3D Computer-Aided Design (CAD) physical model and spatial layout of the sorting conveyor.

3.1 System framework

The architectural design of the lime sorting system shows the intelligent interaction between hardware pieces and intelligent software systems, thus creating a unified system framework that converts visual information into physical actions. This system is based on a multi-level architecture, with the Raspberry Pi 5 acting as the central "brain" and responsible for image processing and higher-level decision-making, as shown in Figure 1. The image data is recorded using the Raspberry Pi Camera Module 3, which is mounted above the conveyor belt at a suitable location to create standard images for the AI model, as shown in Figure 2 (4). For mechanical sorting, the Raspberry Pi coordinates via I2C with the Arduino R3, which controls the timing and precise movement of the six servo motors. Finally, the system also features an ESP32 module to enable connectivity to the Internet of Things (IoT) [10] and to ensure that all sorting operations are logged in a MySQL database and immediately displayed on the mobile user interface.

The system's functional requirements define the central operational tasks necessary to automate the assessment process and create added value for users, as shown in Figure 3. Basically, the system must be able to automatically detect lime that is on the conveyor belt within the field of vision. After recognition, the system must classify the fruits into 7 different categories based on the calculated diameter and surface properties, from quality class A+ to quality class D (rejected), as shown in Figures 4 and 5. In addition to physical sorting, the system must preserve comprehensive records of these operations so that users can view real-time quantities via a mobile application and access a detailed history of the sorted output. Beyond these operational requirements, there are also requirements that are not related to the core functions and focus on the quality, reliability, and user-friendliness of the system. System stability is imperative, as the machine must function reliably under continuous workload without computational delays or communication glitches between the Raspberry Pi and the motor controller. Ease of use is another important aspect. Physical sorting machines are designed to be easily operated by agricultural workers without special technical knowledge. To support this, the mobile application was built with simplicity and clarity in mind. It uses easily understandable representations and a clear graphical user interface to ensure that complex information, such as the daily points distribution, can be grasped immediately, as shown in Figure 6. These requirements certify that the system is not only a practical prototype but also a strong and effective tool for real farming settings.

3.2 Development of an Artificial Intelligence model

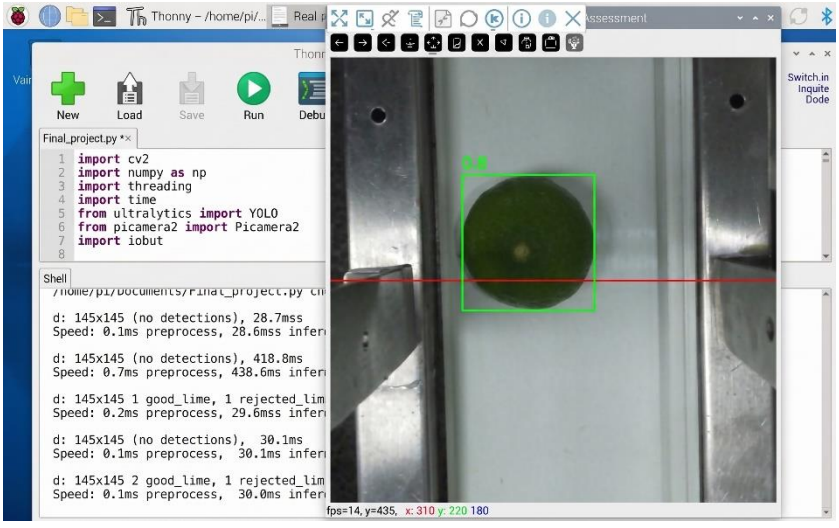

Figure 3. Real-time inference of the YOLOv8n model within the Thonny IDE development environment on the Raspberry Pi 5.

3.2.1 Data collection and investigation.

The foundation of a strong AI model depends on the quality and variety of the data used for training. For this study, a detailed dataset of 7,621 carefully selected images was used, consisting of 6,431 high-

quality limes and 1,190 limes that were rejected due to surface defects, as shown in Figure 5. These photos were taken with an "iPhone 15 Pro Max" at a resolution of 1108 x 1477 pixels. To confirm the model functions effectively when used on a real conveyor belt, the data acquisition environment was strictly controlled: Images were captured from a top-down view at a height of 34.8 centimeters. A 30-lumen flash was used to avoid uneven shadows and to simulate the lighting conditions of a hardware assembly.

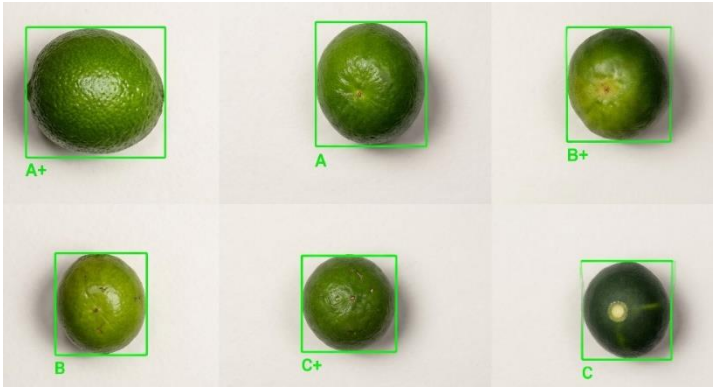

Figure 4. Visual dataset examples of the seven commercial grades predicted by the YOLOv8n model.

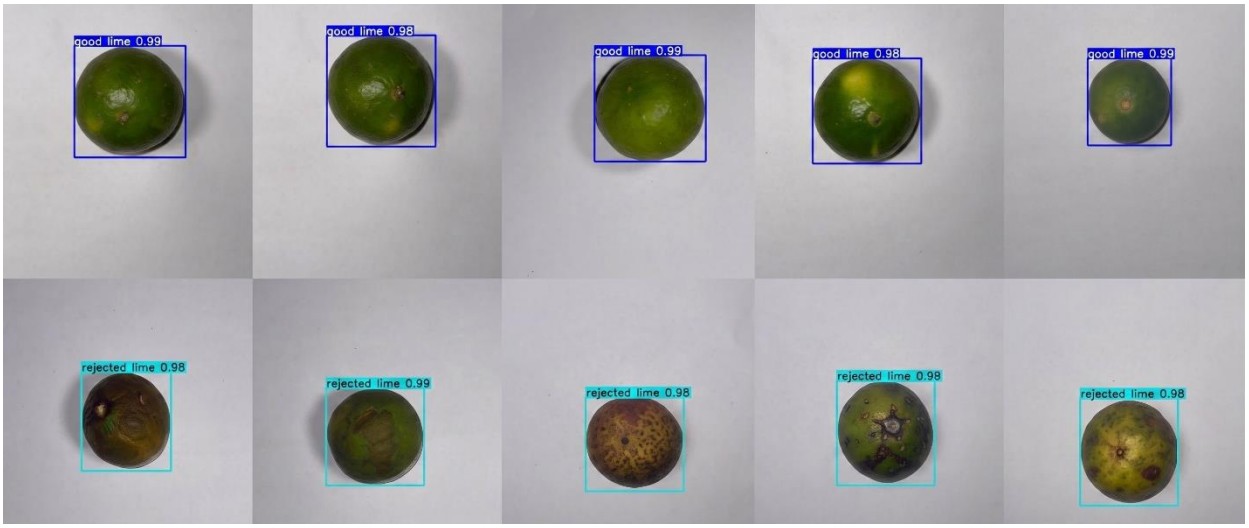

Figure 5. Representation of the custom dataset split into supervised binary categories.



Figure 6. Graphical User Interface (UI) screens of the developed Flutter IoT mobile application.

3.2.2 Accumulation observations and preprocessing.

After data acquisition, the raw images are thoroughly analyzed on the Roboflow platform. Each lime is manually marked with a directional square frame. This important step assists the system in ascertaining the fruit and preparing it for size analysis, as shown in Figures 3, 4, and 5. The data set was then split in a 70:30 ratio, with 5,358 images used for training and 2,263 images for enactment testing and confirmation. This ratio was chosen to ensure that the model can learn the characteristics of a wide variety of fruits while retaining sufficient independent validation samples to verify the accuracy of the predictions.

3.2.3 Training and model improvement.

The development of an AI model using YOLOv8n (Nano) involves a blend of architectural optimization for real-time inference and specific mathematical functions to guide the learning process through objective loss minimization and performance evaluation. This lightweight version of the YOLO architecture was chosen to meet the hardware restrictions of the Raspberry Pi 5 and to confirm that the system can maintain high frame rates during operation for the end user. To further improve performance, the model was converted to the ONNX (Open Neural Network Exchange) format.

1) Core Architectural Logic. YOLOv8n is designed to be the most lightweight variant of the YOLOv8 series, making it ideal for edge devices like the Raspberry Pi 5 used in lime grading system [9]. It utilizes an anchor-free detection head, which simplifies the network by directly predicting the center of an object instead of the offset from a predefined anchor box [9]. The model's backbone uses a C2f module (Cross-stage Partial bottleneck with two convolutions), which is mathematically designed to combine high-level features with contextual information to advance detection accuracy while preserving speed.

2) The Mathematical Foundation: Loss Functions. To train the YOLOv8n model, the system minimizes a multi-part loss function that enumerates the transformation between the predicted results and the ground truth [7]. YOLOv8 specifically employs the following [7], [9]:

- *Classification Loss:* Uses Binary Cross-Entropy (BCE) to classify the lime grade (e.g., Grade A vs. Grade D) [9].

- *Bounding Box Regression Loss*: Uses CIoU (Complete Intersection over Union) and DFL (Distribution Focal Loss) [9].

- *Enhanced Loss Option (EIoU)*: Advanced research on citrus detection suggests replacing the default loss with EIoU, which superiorly handles distinctions in fruit scale and shape [7].

The *EIoU* loss equation is defined as [9]:

$$L_{EIOU} = L_{IOU} + L_{dis} + L_{asp} = 1 - IOU + \frac{p^2(b,b^{gt})}{(w^c)^2+(h^c)^2} + \frac{p^2(w,w^{gt})}{(w^c)^2} + \frac{p^2(h,h^{gt})}{(h^c)^2}, \quad (1)$$

Where *IOU* is the overlap area, b is the center point, and *w/h* are the width and height of the boxes [7]. *gt*: This superscript stands for "*Ground Truth*". For example, wgt and hgt refer to the actual width and height of the target box as defined in the training data. c: This superscript refers to the "minimum bounding box" (also known as the smallest surrounding box) that contains both the predicted box and the target ground-truth box. Therefore, wc and hc epitomize the width and height of this enclosing boundary.

***Loss Components*:**

L_{EIOU}: This represents the "*total Enhanced Intersection over Union loss*". It is a refined loss function used to improve the model's bounding box positioning accuracy and unification speed by accumulating specific constraints on box dimensions.

L_{IOU}: This is the *"standard Intersection over Union loss",* defined mathematically as *1 - IOU*. It measures the basic correspondence between the forecast and ground-truth bounding boxes.

L_{dis}: This is the "*distance loss*" component. It obviously measures the normalized squared Euclidean distance between the center point of the predicted box *(b)* and the center point of the target box *(bgt).*

L_{asp}: This is the "*aspect ratio loss*" component. It enhances the bounding box regression by attracting the alignment of width and height between the predicted and actual boxes, which helps reduce deviations in object scale and shape.

By including these features, the L_{EIOU} loss function provides a more mathematically robust method to guide the AI model (like YOLOv8n) in precisely localizing objects such as citrus fruits on a conveyor system.

3) Grading and Sizing Equations: In the Lime Grading Machine project, the AI model does not just identify the lime; it also calculates its physical size using the "Euclidean Distance" formula derived from the predicted Bounding Box.

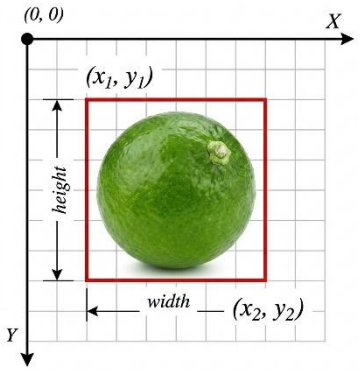

Figure 7. Geometric bounding box mapping on a Cartesian image plane.

Pixel Distance Equation:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}, \tag{2}$$

Where (x_1, y_1) and (x_2, y_2) are the coordinates of the bounding box corners.

Real-World Sizing Conversion: To convert the detected pixels into millimeters for commercial grading, the system uses a calibrated conversion factor:

$$Actual\ Distance\ (mm) = \frac{\sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2}}{36.9}, \tag{3}$$

The factor 36.9 is calibrated based on the camera height of 34.8 cm above the conveyor.

4) Model Assessment Metrics. To assess how well the YOLOv8n model performs, several statistical equations are applied to the test dataset (the 30% split of 7,621 images) [1]:

Accuracy Rate: Measures overall correctness [1].

$$Accuracy = \frac{(TP+TN)}{(TP+FP+FN+TN)}\ x\ 100\%, \tag{4}$$

Precision: Measures the exactness of positive (lime detection) predictions [1].

$$Precision = \frac{TP}{(TP+FP)}\ x\ 100\%, \tag{5}$$

Recall: Measures the ability to find all actual limes in the image [1].

$$F1 = 2\ x\ \frac{(Precision\ x\ Recall)}{(Precision+Recall)}, \tag{6}$$

F1-Score: The harmonic mean of Precision and Recall, used to ensure a balance between the two [1]. Where TP = True Positive, TN = True Negative, FP = False Positive, and FN = False Negative.

To comprehensively evaluate the localization and classification performance of the YOLOv8n model, we introduce the standard object detection metric Mean Average Precision (*mAP*). Average Precision (*AP*) is computed as the area under the Precision-Recall curve for each class. The *mAP* is defined as:

$$mAP = \frac{1}{C} \sum_{i=1}^{C} AP_i \tag{7}$$

where $C = 2$ represents our target classes ($good_{lime}$ and $rejected_{lime}$). We report *mAP@0.5*, evaluated at a fixed Intersection over Union (IoU) threshold of 0.5, and *mAP@0.5:0.95*, which represents the average *mAP* calculated across ten *IoU* thresholds from 0.5 to 0.95 with a step size of 0.05. These metrics evaluate the model's spatial alignment accuracy alongside its classification performance.

5) Decision logic for commercial grade assignment. More than simple appreciation, the AI model was also trained to accurately estimate the size. By applying the "Euclidean distance" formula to the size of the surrounding frame, the system converts the pixel measurements into real millimeters, as shown in Figure 7. This information is then processed by a logical algorithm to classify limes into six specific trade classes (A+, A, B+, B, C+, or C) or a seventh class, the "reject class" (D), for defective fruit, as shown in Figures 3, 4, and 5. By assimilating deep learning and computer-aided geometry appreciation, the system can replace subjective human inspection with a standardized and impartial quality control process.

6) Summary of Development Workflow. To ensure absolute reproducibility of the training process and the resulting model weights, the training of the YOLOv8n (Nano) network was executed under a strict, standardized set of hyperparameters. Training was conducted using the PyTorch framework on an NVIDIA GPU platform with CUDA acceleration to facilitate parallel processing and rapid convergence. Automatic Mixed Precision (AMP) was disabled to preserve numerical precision consistency throughout the training steps. The model was trained for 20 epochs. Limiting training to 20 epochs was a deliberate design choice; preliminary testing indicated that extending training beyond this threshold led to severe overfitting, which degraded the model's ability to generalize to unseen fruit samples on the moving conveyor belt. The network weights were updated using the "Adam optimizer" with an initial learning rate (lr0) of (0.001), a final learning rate fraction (lrf) of (0.01), and a weight decay coefficient of (0.0005) to apply (L2) regularization.

For data preprocessing, the original high-resolution images (1108 * 1477 pixels) were resized to a standard resolution of 640 * 640 pixels to maintain a balanced computational overhead on the target Raspberry Pi 5 edge device. Data augmentation was active during training to increase model generalization. Mosaic augmentation was enabled at full strength (1.0), dynamically combining four random training images to teach the model to identify limes of varying sizes and positions within the spatial grid. "HSV color jittering" was also applied to simulate real-world illumination changes, such as the 100-lumen conveyor light source vs. the 30-lumen initial capture conditions. "Mix-up augmentation" was disabled (0.0) as it represents an over-regularization risk for lightweight networks like YOLOv8n. The complete training configuration is detailed in Table 1.

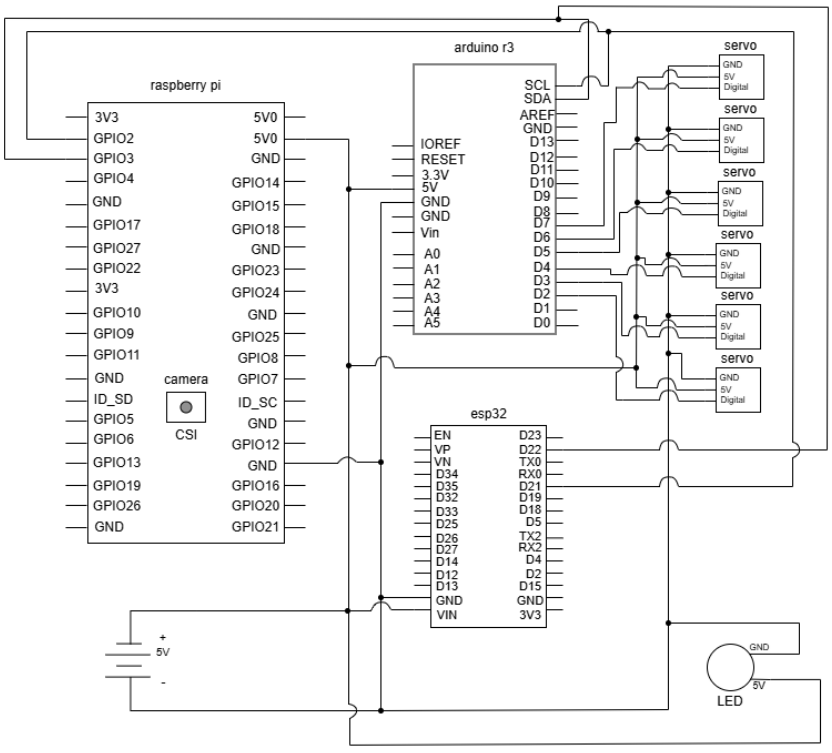

Figure 8. Electronic schematic and wiring diagram of the multi-controller edge architecture.

Table 1. YOLOv8n Training Hyperparameters and Data Augmentation Settings

Category	Hyperparameter	Configured Value/ Setting	Description
Optimization & Training	Optimizer Type	Adam	Optimization algorithm for weight updates
	Epochs	20	Total training iterations
	Batch Size	16	Number of training samples per batch
	Image Size (imgsz)	640 * 640	Input dimension resized from raw images
	Initial Learning Rate (lr0)	0.001	Initial optimizer learning rate
	Final Learning Rate Fraction (lrf)	0.01	Multiplier for final decay learning rate
	Weight Decay	0.0005	L2 regularization penalty to prevent overfitting
	Computational Platform	CUDA (GPU)	Parallel GPU computing architecture
	Mixed Precision (AMP)	Disabled	Turned off to preserve gradient calculation precision
Data Augmentation	Mosaic Augmentation	1.0 (Fully Enabled)	Combines 4 training images into a single frame
	Mix-up Augmentation	0.0 (Disabled)	Blends two images (disabled for nano model complexity)
	Hue Jitter (hsv_h)	0.015	Hue channel adjustment range
	Saturation Jitter (hsv_s)	0.7	Saturation channel adjustment range
	Value/Brightness Jitter (hsv_v)	0.4	Simulates conveyor lighting variations
	Horizontal Flip (fliplr)	0.5 (Probability)	Left-right mirroring of the rotationally symmetric fruit
	Vertical Flip (flipud)	0.0 (Disabled)	Up-down mirroring
	Translation (translate)	0.1	Horizontal/vertical shift fraction on the conveyor
	Scaling (scale)	0.5	Zoom gain fraction to adjust for camera-to-belt height

The training involves 20 epochs (iterations) using the Adam optimizer to adjust weights [7]. By processing images concluded by these equations, the YOLOv8n model in project achieved a 99.75% Accuracy Rate in classifying lime quality and a 96.86% accuracy in real-time physical grading. This performance improvement markedly cuts the computing burden of the Raspberry Pi processor.

The core of the system's sensor technology is the imaging station, on which the Raspberry Pi camera module 3 is mounted in a suitable position at a height of 34.8 cm above the belt, as shown in Figure 1. To ensure a consistent environment, a key factor for the accuracy of the "YOLOv8 model", this station is furnished with a 100-lumen light source to eliminate shadows and create a standardized background that mimics the conditions during AI training. The conveyor belt operates at a calibrated speed of 17.5 revolutions per minute. This precisely timed speed allows the image processing system to detect, process, and classify each individual piece of fruit before it reaches the sorting gate.

3.3.2 The electronic design uses a hierarchical approach and multiple controllers to cope with the high computing requirements of the AI in conjunction with the mechanical real-time control. The Raspberry Pi 5 serves as the primary processing unit and handles the YOLOv8 inference and the higher-level scoring logic. To preserve system stability and preclude processing delays, the Raspberry Pi 5 uses the I2C communication protocol to assign low-level tasks to peripheral controllers. This circuit diagram consists of an Arduino R3, which acts as a dedicated motor controller and converts grade-assignment commands into precise movements for six high-torque servo motors, as shown in Figure 8. These motors drive mechanical propellers that gradually transport limes into sorting containers according to the classification by artificial intelligence (AI).

An optional component of the control layer is the ESP32 module, which allows the system to utilize Internet of Things (IoT) features. The ESP32 receives sorting information in real time from the main controller and sends it to a MySQL database. This allows the mobile application to display the sort count and historical data in real time, as shown in Figure 6. To ensure that all components act reliably even under continuous load, the entire electronic assembly uses a 5V 6A switching power supply that provides a stable and constant voltage to fulfill the diverse requirements of digital processors and analog motors. This integrated design enables a smooth, robust, and repeatable transition from recorded images to actual lime sorting.

3.4 Software Design

The software architecture of the AI-supported lime sorting system acts as a logical bridge between high-speed image processing and real-time mechanical processing. To guarantee the robustness and expandability of the system, the software design was carefully created using the Unified Modeling Language (UML), a formalized framework for both the organizational structure and the dynamic functioning of the system, as shown in Figure 9 - 12.

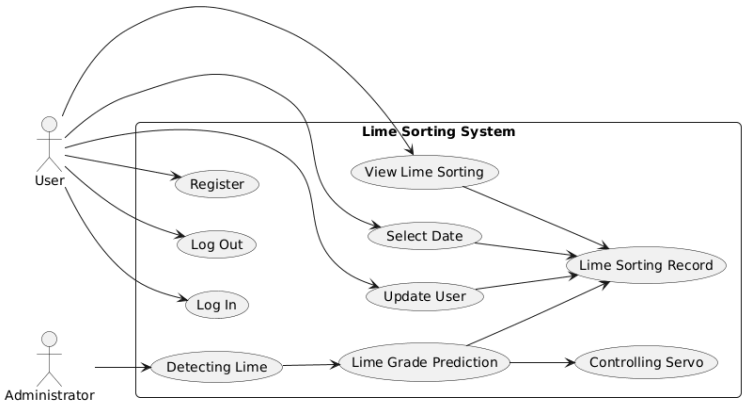

Figure 9. Unified Modeling Language (UML) Use Case Diagram. (Generated using PlantUML).

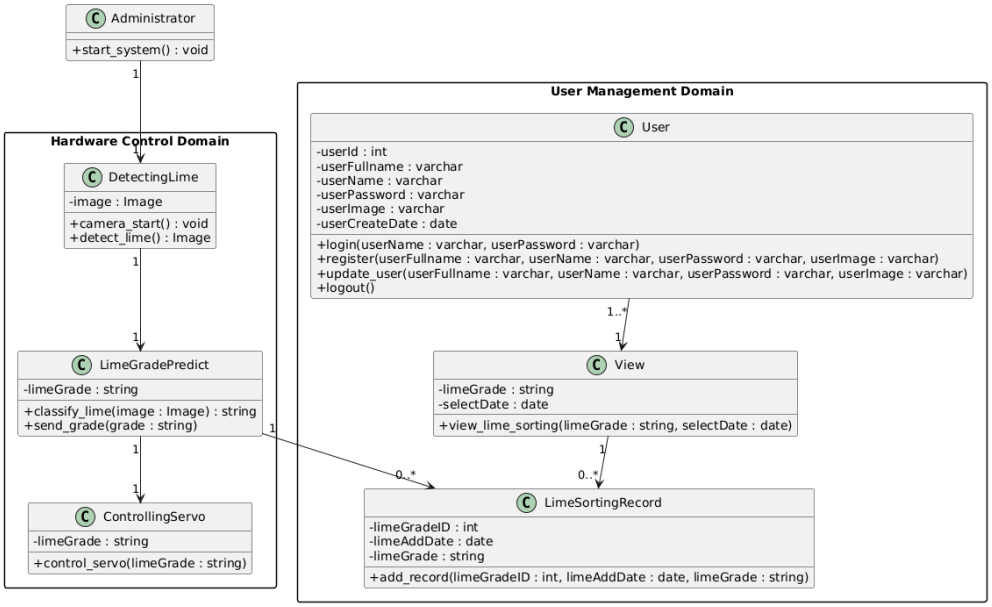

Figure 10. UML Class Diagram outlining the object-oriented software architecture (Generated using PlantUML).

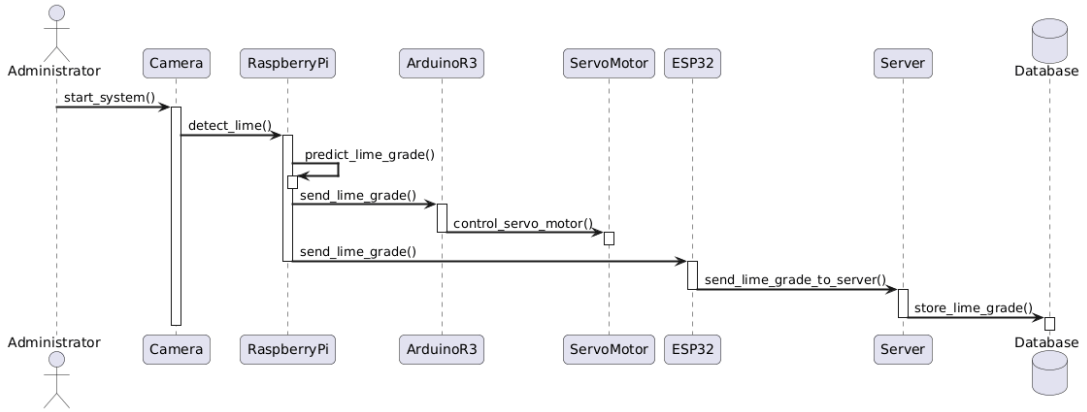

Figure 11. UML Sequence Diagram detailing the temporal message sequence of active sorting (Generated using PlantUML).

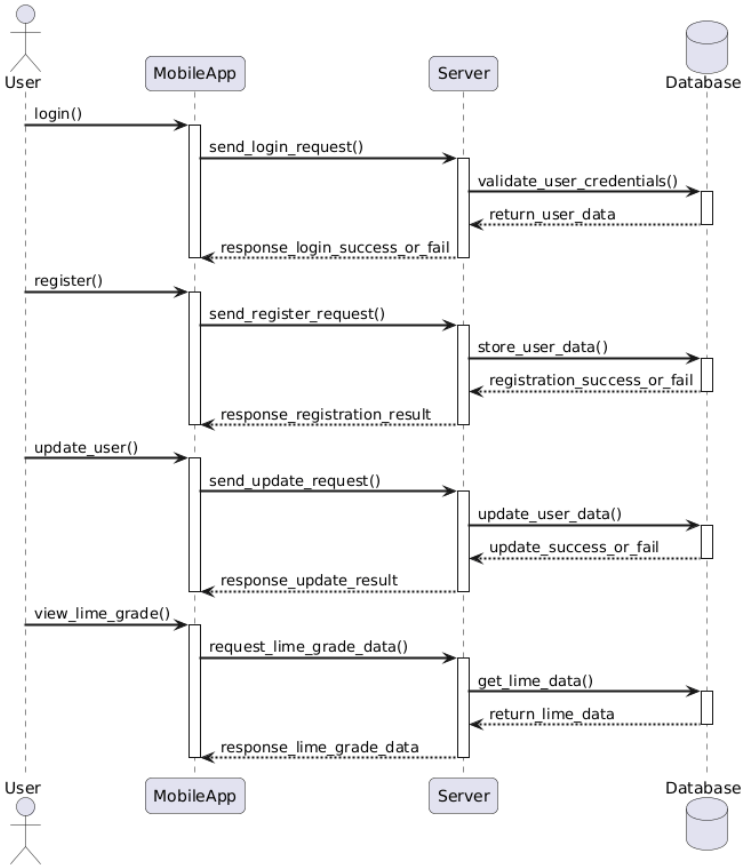

Figure 12. UML Sequence Diagram mapping mobile application interaction (Generated using PlantUML).

3.4.1 Definition of the scope of work and creation of use-case models. The system's functionality is defined by a use case diagram that divides responsibilities between two main roles: the system administrator and the users. In subsequent phases, the system administrator will monitor important automated functions, including limescale detection, level prediction, and control of the servomotor sorting gate. Conversely, users interact with the administration area to perform various actions such as registration, authentication, sorting, and data analysis. Overall, this machine has nine basic use cases, ranging from real-time prediction of lime content to previous lime sorting records.

3.4.2 Anatomy and structure: Class diagram. A class diagram illustrates the "structure" of a system and describes in detail the attributes and methods of the objects that make up the system. The architectural structure is divided into a hardware control domain (consisting of classes such as DetectingLime, LimeGradePredict, and ControllingServo) and a user management domain (User, View, and LimeSortingRecord), as shown in Figure 10. For example, the DetectingLime class handles the standard image capture, while the LimeGradePredict class processes the classification logic and converts image data into commercial quality levels (A+ to D) before the ControllingServo class is triggered to make physical deviations.

3.4.3 Behavioral logic: Modeling of activities and event arrangements. The active operation of the system is documented by an activity diagram that shows the rational sequence from beginning to end, as illustrated in Figures 11 and 12. The main cycle of automatic operation begins when the camera module detects limes on the conveyor belt. The system then enters the decision phase, in which the "YOLOv8 model" evaluates the size and surface quality of the fruits. If the limes meet the specified size criteria (e.g., larger than 50 mm for grade A+), the corresponding servo motor is activated. Otherwise, the system continues with the next higher grade until a lime is encountered that meets the criteria, or until it is sorted out as grade D. The flowchart explains this time sequence in more detail and illustrates a high-speed communication chain: The administrator initiates the system, causes the Raspberry Pi to detect activity, and then the Raspberry Pi coordinates with the Arduino R3 for motor control and the ESP32 for transmitting data to the MySQL database. Parallel execution is used in mobile apps where user data requests are validated by the server and returned for visual display.

3.4.4 Human Interaction (HI) procedure. The human-interaction interface developed with the Flutter framework was designed for ease of use in high-accuracy agriculture. The design uses a gradient of green hues to mirror the agricultural characteristics of the product.

- *Data visualization*: To furnish an immediate overview, a pie chart is used on the home screen to show the overall distribution of sorted limes across all quality levels, as shown in Figure 3.

- *Time-based analysis:* A dedicated date selection function permits users to screen the results. These are then displayed in a bar chart to resemble daily yield and quality trends. This user interface assures that complex AI-powered data is transformed into a user-friendly and actionable setup for farmers, allowing real-time monitoring of crop quality (see Figure 6).

4. System investigation

The testing procedure in this study constitutes an important turning point, as it combines theoretical artificial intelligence with the physical challenges of agricultural machinery. To ensure the reliable operation of the lime separator in a commercial environment, the evaluation process was divided into two main parts: the computational validation of the YOLOv8n model and the physical performance of the integrated conveying system, as shown in Figure 13.

4.1 Validation of the mission of the AI model.

The first test step focused exclusively on the software's ability to determine and differentiate features of static data. After intensive training, the model was tested using a dataset of 2,424 images (2,090 "good" images and 328 "rejected" images) that the system had never encountered during training. The distribution of the data in this 70:30 ratio provides a solid basis for assessing the ability to draw general conclusions. The results of this calculation method are excellent: The model obtains accuracy and an F1 score of up to 99.75%. To represent the prediction inaccuracies not only as percentages, but also, the investigation uses a confusion matrix as shown in Figures 14 and 15. This study proves that AI is able to perfectly distinguish "good" fruit from "rejected" fruit, with only six minor errors occurring in thousands of samples.

Following model training, computational validation was conducted on an independent test split of 2,263 images (1,939 "good" and 324 "rejected"). The overall performance results, including the newly introduced object detection metrics, are summarized in Table 2:

Table 2. YOLOv8n Performance Metrics on Test Dataset

Evaluation Metric	Mathematical Focus	Value (%)
Accuracy	Overall prediction correctness	99.75%
Precision	Positive predictive value (exactness)	99.75%
Recall	True positive rate (completeness)	99.75%
F1-Score	Harmonic mean of Precision and Recall	99.75%
mAP@0.5	Mean Average Precision at IoU = 0.5	98.35%
mAP@0.5:0.95	Mean Average Precision over IoU	72.87%

4.2 Integrated system and hardware testing.

In the second test phase, the use of digital images was changed to machine sorting in a natural environment. This test was conducted on a specially designed conveyor belt running at a calibrated speed of 17.5 revolutions per minute and operated with standard 100-lumen illumination to ensure consistent image quality. The test was divided into three specific experiments:

4.2.1 Randomized Market Test: To simulate real market distribution, 100 lime samples were purchased directly from a local market. In this test, the system reached an accuracy of 88%, effectively handling the "unstructured" data caused by natural harvest fluctuations.

4.2.2 Binary classification test: To test the system's ability to detect defective products, 100 items were processed (50 perfect and 50 defective). The system correctly classified all limes into this category, exhibiting its reliability as a safety filter for sorting out damaged products.

4.2.3 Accuracy testing by quality grade: Finally, 70 limes (10 per quality grade) were tested to evaluate the accuracy of the size classification. The system attained a hit rate of 93% in these subgroups. The system's strong accuracy in the extreme size classes is notably significant: quality grade A+ (> 50 mm) and quality grade D (rejected), where a hit rate of 100% was achieved.



Figure 13. Physical implementation and experimental setup of the automated AIoT-based lime fruit grading and sorting machine.

4.3 Failure analysis and specialized accuracy.

The test results showed that when defects did occur, they were typically limited to regions with a small diameter of only 1-2 millimeters. This difference is largely due to the physical orientation of the limes as they roll down the conveyor belt, and not to a flaw in the AI logic itself. Comparing the results of tests using only the software (99.75%) with the final accuracy of the integrated assessment (96.86%) shows that while the AI is very accurate, physical movements introduce minor variables that need to be taken into account in smart farming. To summarize, the multi-stage tests confirm that this machine is a foremost alternative to manual inspection and offers a standardized, repeatable, and fast solution for lime sorting.

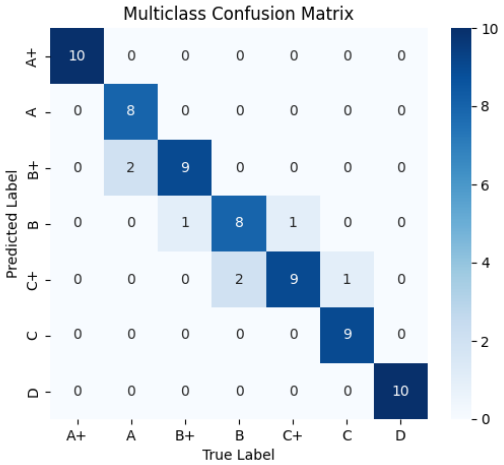

Figure 14. Binary Confusion Matrix evaluating quality classification on the static validation dataset.

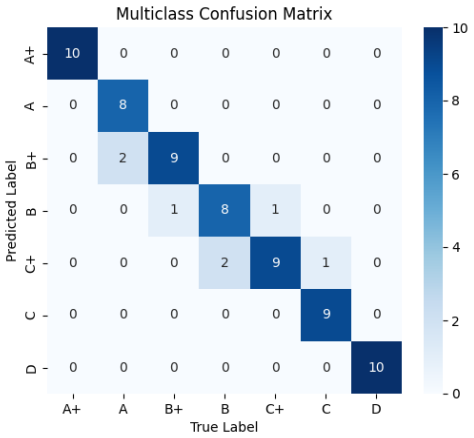

Figure 15. Multiclass Confusion Matrix evaluating physical sorting performance during active conveyor belt trials.

5. Evaluation and Conclusion

The evaluation of lime classification and sorting procedures using AI clearly demonstrates how computer-aided AI can direct physical fineness in the agricultural context. To assess the effectiveness of the system, its performance was thoroughly examined in two different environments: controlled software tests and dynamic hardware tests on a real conveyor belt.

5.1 Quantitative analysis of performance and precision.

The preferred evaluation of the standalone YOLOv8n model yielded excellent results with an accuracy of 99.75% and an F1 score of 99.75% during the training and validation phases. This step, which used a dataset of 7,621 images, demonstrated the robust capability of the model to discriminate between high-quality fruit and fruit with skin defects in a static environment. However, the real indicator of the research's effectiveness exists in the practical application of the integrated hardware. When the machine is used in a real conveyor system, it sustains high accuracy, with an overall sorting accuracy of 96.86%. A closer look at the individual evaluation categories shows that the system is most reliable at the extremes of the spectrum. Categories A+ (diameter > 50 mm) and D (rejected fruit) attain 100% accuracy. The intermediate categories (A, B+, B, C+, and C), on the other hand, reveal low variability, showing error rates between 10% and 20% per category. These deviations are typically limited to a narrow range of 1 to 2 millimeters, and this variation is mainly due to the physical orientation of the lime as it rolls down the conveyor belt, and not to a flaw in the logic of the AI system.

A critical factor for real-world viability in precision agriculture is the execution latency on resource-limited edge hardware. To assess real-time performance, the trained model was deployed on the Raspberry Pi 5 Single Board Computer (Broadcom BCM2712 Quad-Core @ 2.4 GHz). To optimize performance under hardware limitations, the model was converted from PyTorch (.pt) to the highly efficient ONNX format, utilizing ONNX Runtime for hardware acceleration on the CPU. Empirical testing on the active conveyor belt reveals that the system achieves a processing latency of ~300 ms per frame. Detailed execution profiles from Thonny IDE show a preprocessing latency of 7.2 ms and an active inference time of 307.5 ms when detecting fruit, as shown in Figure 3. This latency yields a stable execution speed of 3.33 Frames Per Second (FPS). Operating at 17.5 RPM, the conveyor transports limes sequentially at a physical rate of 8 limes per minute (approx. 7.5 seconds per lime). This gives the system a comfortable buffer of approximately 25 frames of analysis per lime before the fruit reaches the corresponding physical sorting gate, ensuring robust and highly reliable edge sorting without computation queues.

5.2 Comparison data: Binary sorting vs. multi-class sorting.

An interesting finding during the evaluation was the performance difference between binary category and multi-stage sorting. If the system's only task is to identify "good" and "bad" limes, it can do so with 100% accuracy. The slight decrease in accuracy in complete sorting (96.86%) is a consequence of the greater convolution in handling seven different outcome classes and the physical variables of the moving conveyor belt. Compared to other modern research, such as the “FruitVision” system, which achieves an accuracy of up to 99.12% in lime classification across several structural designs, the use of YOLOv8 in this project achieves a highly effective balance between velocity and accuracy for real-world applications on end-user devices such as the Raspberry Pi 5.

5.2.1 Mathematical Significance of the L_{EIOU} Loss Function: To understand the high localization precision of the developed system, it is necessary to examine the bounding box regression mechanics. The default bounding box loss in standard YOLOv8 architectures is the Complete Intersection over Union (L_{CIoU}), formulated as:

$$L_{CIoU} = 1 - IoU + \frac{\rho^2(b,b^{gt})}{c^2} + \alpha v \tag{8}$$

where α is a positive trade-off parameter, and v measures the aspect ratio consistency via the following relation:

$$v = \frac{4}{\pi^2}\left(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w}{h}\right)^2 \tag{9}$$

A mathematical limitation of L_{CIoU} lies in this aspect ratio penalty term (v). Because v only measures the relative difference in aspect ratio, if the predicted width *(w)* and height *(h)* scale proportionally such that $\frac{w}{h} = \frac{w^{gt}}{h^{gt}}$, the value of v becomes zero. Consequently, the aspect ratio loss term ceases to provide gradients, which can lead to convergence stagnation and imprecise bounding box dimensions, even if the absolute size of the predicted box is incorrect. To resolve this limitation on our conveyor line, we implemented the Enhanced Loss Option (L_{EIOU}). The L_{EIOU} loss function decomposes the aspect ratio term by directly targeting the deviations between the predicted and ground-truth dimensions independently:

$$L_{EIOU} = 1 - IoU + \frac{\rho^2(b,b^{gt})}{(w^c)^2+(h^c)^2} + \frac{\rho^2(w,w^{gt})}{(w^c)^2} + \frac{\rho^2(h,h^{gt})}{(h^c)^2} \tag{10}$$

By calculating the normalized squared differences of width and height separately relative to the minimum enclosing box dimensions (*wc* and *hc*), L_{EIOU} provides continuous, highly stable gradients for box scaling. This mathematical precision directly influences downstream physical grading accuracy. Since our system calculates the physical lime diameter (Actual Distance in *mm*) by applying the Euclidean distance formula to the coordinates of the predicted bounding box (Equations 2 and 3), a spatial offset of just 2 pixels on the raw *1108 * 1477* image translates to an error of nearly 1.5 mm in physical size estimation. Such small errors are enough to misclassify a borderline lime (e.g., misclassifying a 40.5 mm Grade C+ lime as Grade C). By leveraging L_{EIOU}

to minimize boundary localization error during training, our model maintains highly reliable spatial predictions, limiting physical boundary variations to a narrow range of *1* to *2 mm*.

5.2.1 Comparative Performance against Existing Literature: To demonstrate the practical and scientific contribution of our developed machine, Table 3 evaluates our results alongside other agricultural sorting methodologies detailed in our literature review.

5.2.3 Synthesis and Discussion: As illustrated in Table 3, traditional color-sensor setups, such as the system designed by Rodkasem et al. [13], offer higher sorting throughput (14.80 fruits/min) but struggle to achieve high grading precision, yielding an overall size accuracy of only 79.33%. This limitation is due to the lack of spatial computer vision; standard sensors cannot measure irregular boundaries or differentiate complex surface defects. Conversely, high-performance deep learning models designed for servers, such as FruitVision (99.12% accuracy) and YOLOv8-Scm (93.4% mAP), deliver exceptional classification accuracy but are computationally expensive, making them impractical for deployment on lightweight, cost-sensitive agricultural edge devices. Our system successfully bridges this gap. By utilizing a lightweight YOLOv8n model optimized for ONNX Runtime, we successfully deployed the framework on a Raspberry Pi 5 CPU without external GPU accelerators. Our system achieves a high validation accuracy of 99.75% and maintains a 96.86% sorting accuracy during dynamic conveyor belt trials. Unlike prior works that only categorize fruit into binary classes or simple maturity states, our system evaluates distinct commercial grade categories (A+ to C, and D-Reject) while simultaneously tracking crop yields in real time via a custom mobile application. This represents a highly accessible, complete, and scientifically rigorous precision agriculture solution ready for commercial operation.

Table 3. Comparative Analysis with Existing Fruit Grading and AIoT Agricultural Systems

System / Study	Classification Method	Evaluation Metric(s)	Processing Speed	Hardware Platform	Grading Depth & Features
Rodkasem et al. [13]	Traditional Color Sensors	79.33% Size Accuracy 83.33% Color Accuracy	14.80 fruits/min	Custom Logic Gates (Wooden Frame)	3 Grades (Size/Color)
Aekkarat Suksukont [12]	Custom CNN	92.0% (Unripe Lime) 98.0% (Ripe Lime)	Static processing	Server PC	2 Grades (Maturity counting only)
Bunsen [14]	ANN / SVM Classifier	95.0% Accuracy (ANN) 92.0% Accuracy (SVM)	Offline analysis	Desktop Server	Defect sorting (Sooty Mold, Canker, Thrip)
FruitVision (Hayat) [8]	MobileNetV3	99.12% Accuracy	Real-time	Server GPU / PC	Multi-class sorting
YOLOv8-Scm (Cong) [7]	Improved YOLOv8 with EIoU	mAP@0.5: 93.4%	Real-time	Server GPU	5 Sunburn Severity Grades (Citrus)
Proposed AIoT System	YOLOv8n + LEIOU + Euclidean Bounding Box	99.75% Validation Accuracy 96.86% Physical Grading Accuracy 100% Binary Defect Detection	3.33 FPS (8 fruits/min)	Raspberry Pi 5 CPU (Optimized ONNX)	7 Distinct Grades (A+ to C, and D-Reject) Flutter IoT Monitoring App

5.3 Conclusion

The design, development, and deployment of an intelligent, AIoT-equipped lime sorting machine represents a significant advancement in precision agriculture, successfully bridging the gap between advanced deep learning frameworks and robust physical automation. By integrating an optimized YOLOv8n architecture with cost-effective hardware—namely a Raspberry Pi 5, an Arduino R3, and high-torque digital servo motors—this research has delivered a highly accurate, accessible alternative to labor-intensive manual sorting. The system successfully achieved its core objectives: establishing a standardized grading workflow with high localization precision and providing farmers with an intuitive mobile dashboard for real-time crop yield analytics. Despite these successes, future iterations can further enhance throughput by incorporating dedicated edge-AI hardware accelerators (such as the Coral USB Accelerator) and replacing traditional rotational servos with industrial-grade pneumatic actuators. In conclusion, this study offers a scalable and cost-effective blueprint for automated quality control in citrus agriculture, paving the way for reduced post-harvest losses and increased market value.

6. Acknowledgements

The authors thank the Thai Pipe Industry Co., Ltd. for the financial support. Thank you, Ms. Thidarat Hirun, Mr. Nontakorn Pravatvithisuk, Mr. Pitipong Chanplaeng, and Ms. Aphasara Mahawam, who assisted in installing and maintaining both hardware and software. We also want to thank Dr. Panuwat Taerakul, Dean of Engineering Faculty at Southeast Asia University, for his significant support and guidance throughout the project.

Author Contributions: Conceptualization, N.O. and S.T.; methodology, N.O. and S.T.; software, N.O., S.R. and C.C.; validation, N.O., S.R. and C.C.; formal analysis, N.O.; investigation, N.O. ank V.K.; resources, N.O., V.K. and S.T.; data curation, N.O. and S.T.; writing—original draft preparation, N.O.; writing—review and editing, N.O.; visualization, N.O. and S.T.; supervision, N.O.; project administration, N.O.; funding acquisition, N.O.

Funding: This research was funded by Thai Pipe Industry Co., Ltd.

Conflicts of Interest: The authors declare no conflict of interest.

References

[1] Goyal, A.; Lakhwani, K. Integrating Advanced Deep Learning Techniques for Enhanced Detection and Classification of Citrus Leaf and Fruit Diseases. *Sci. Rep.* **2025**, *15*(1), 12659. https://doi.org/10.1038/s41598-025-97159-0

[2] Xiang, T. W.; Ting, L. S.; Chet, K. V.; Soong, L. W. Design and Development of an Automated Fruit Quality Classifier and Sorter Using a Robotic Arm. *Edelweiss Appl. Sci. Technol.* **2025**, *9*(10), 32–49. https://doi.org/10.55214/2576-8484.v9i10.10340

[3] Sapkota, R.; Karkee, M. Comparing YOLOv11 and YOLOv8 for Instance Segmentation of Occluded and Non-occluded Immature Green Fruits in Complex Orchard Environment. *arXiv* **2024**, arXiv:2410.19869.

[4] Joshi, H. Edge-AI for Agriculture: Lightweight Vision Models for Disease Detection in Resource-Limited Settings. *arXiv* **2024**, arXiv:2412.18635.

[5] Otanasap, N.; Chalermsuk, C.; Bungkomkhun, P. A Survey of IoT: Advances in Smart and Dynamic Environmental Monitoring. In *Proceedings of the 8th International Conference on Environmental Engineering, Science and Management*; 2019.

[6] Anusha, K.; Uma, K.; Jayasri, K.; Kambham, S.; Dandamudi, S. D. IoT Based Food Spoilage Detection Using Machine Learning Techniques. In *Proceedings of the 2024 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI)*; IEEE, 2024; pp 1–7. https://doi.org/10.1109/ACCAI61061.2024.10602355

[7] Cong, G.; Chen, X.; Bing, Z.; Liu, W.; Chen, X.; Wu, Q.; Guo, Z.; Zheng, Y. YOLOv8-Scm: An Improved Model for Citrus Fruit Sunburn Identification and Classification in Complex Natural Scenes. *Front. Plant Sci.* **2025**, *16*, 1591989. https://doi.org/10.3389/fpls.2025.1591989

[8] Hayat, A.; Morgado-Dias, F.; Choudhury, T.; Singh, T. P.; Kotecha, K. FruitVision: A Deep Learning Based Automatic Fruit Grading System. *Open Agric.* **2024**, *9*(1), 20220276. https://doi.org/10.1515/opag-2022-0276

[9] Terven, J.; Córdova-Esparza, D. M.; Romero-González, J. A. A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS. *Mach. Learn. Knowl. Extr.* **2023**, *5*(4), 1680–1716. https://doi.org/10.3390/make5040083

[10] Somsupharprungyos, S.; Tanveenukul, R.; Nokkruth, B. The Development of a Real-Time Data Collection System from Sensor Networks to Support Smart Farming Application Development in Communities. *SAU J. Sci. Technol.* **2024**, *10*, 74–84.

[11] Otanasap, N.; Tadsuan, S.; Chalermsuk, C. An AIoT-Based Air Quality Monitoring System for Real-Time PM2.5 Prediction in Urban Environments. *ASEAN J. Sci. Technol. Rep.* **2025**, *28*(1), e255168. https://doi.org/10.55164/ajstr.v28i1.255168

[12] Suksukont, A. Development of Counting and Analysis Lemons System with Computer Vision Technology. *J. Inf. Sci. Technol.* **2023**, *13*(1), 10–16.

[13] Rodkasem, S.; Wongpotha, S.; Yoddamnern, T. Lemon Color and Size Sorting Machine. In *Proceedings of the 4th National Conference on Learning Innovation and Invention 2020*; Pathum Thani, Thailand, December 17, 2020; Rajamangala University of Technology Thanyaburi: Pathum Thani, Thailand, 2020.

[14] Boonsem, T. *Surface Defect Detection in Lime by Using Computer Vision*; Master's Independent Study, Sirindhorn International Institute of Technology, Thammasat University: Thailand, 2022.