Comprehensive Journal of Science
Volume (11), Issue (41), (JUNE, 2026)
ISSN: 3014-6266

مجلة العلوم الشاملة
المجلد (11) العدد 41 يونيو 2026
ردمد: 3014-6266

Fruit Ripeness Detection Using a YOLO-Based Machine Learning Framework

Hazem Ashor Amran Abolholl[1], Esam Alzmazem[2], Moamer Ehtash[3], and Taihiret Alhashan[4]
[1]College of Engineering Technologies - Yefren, abolholl.hazem@gmail.com [2]Higher Institute Of Engineering Techniques - Tripoli, jane.doe@example.com [3]College of Engineering Technologies - Yefren, moamerehtash@gmail.com [4]College of Engineering Technologies - Yefren, tgrada@yahoo.com

تاريخ الاستلام:2026/05/06 –تاريخ المراجعة:2026/05/28– تاريخ القبول: 2026/06/08– تاريخ للنشر: 2026 /06/28

Abstract

Accurately and efficiently assessing fruit ripeness is crucial for optimising harvesting, reducing post-harvest losses, and ensuring consistent product quality. This study introduces a robust fruit ripeness detection framework based on the You Only Look Once (YOLO) architecture, combining real-time object detection with ripeness classification. A dataset covering multiple fruit types, including apples, bananas, and tomatoes, was created and annotated across four ripeness stages: unripe, partially ripe, ripe, and rotten. To enhance model robustness under varying lighting, occlu- sion, and background conditions, extensive data augmentation and preprocessing techniques were applied. The YOLO-based model achieved strong performance, with a mean Average Precision (mAP) of 96.4%, precision of 95.7%, recall of 94.8%, and an overall ripeness classification accuracy of 96.1%. Comparisons with YOLOv3 and YOLOv5 variants demonstrated that YOLOv8 offered the highest detection and classification accuracy while maintaining low inference latency, making it suitable for real-time deployment in both field and greenhouse environments. Qualitative tests also confirmed reliable detection under complex lighting and background scenarios. This framework holds particular promise for application in Libya, where agriculture is a key sector of the economy. By enabling fast, objective, and consistent assessment of fruit quality, the system supports the modernisation of Libyan agriculture and contributes to improved productivity, sustainability, and post-harvest management. Integrating detection and classification into a unified, real-time system offers a scalable solution for smart farming, yield optimisation, and quality control.

1 Introduction

Accurately assessing fruit ripeness is a critical requirement in modern agriculture, influencing harvest timing, post–harvest handling, supply chain efficiency, and overall product quality. Traditional ripeness evaluation methods rely heavily on manual inspection, which is inherently subjective, labor-intensive, and prone to inconsistency, particularly in large-scale production environments [1]. These limitations contribute significantly to post-harvest losses, which can exceed 30% in many agricultural systems, underscoring the need for objective and automated solutions.

Recent advances in computer vision and deep learning offer promising alternatives to manual quality assessment. Convolutional neural networks (CNNs) have demonstrated strong capabilities in extract- ing high-level visual features relevant to agricultural tasks, achieving performance comparable to or exceeding human evaluation in classification and detection problems [2, 3]. Among deep learning-based approaches, the You Only Look Once (YOLO) family of object detection models has emerged as a leading candidate due to its balance of

accuracy, speed, and ability to operate in real time [4, 5]. YOLO's single-stage detection paradigm enables simultaneous object localisation and classification, making it well suited for field applications where rapid decision-making is essential.

A growing number of studies have adopted YOLO variants for fruit detection and ripeness clas- sification across a range of crops, including strawberries, avocados, tomatoes, and citrus [6, 7, 8].

Improved YOLO-based models have been developed to overcome domain-specific challenges such as small fruit sizes, occlusion, dense foliage, and variable lighting, as seen in ORD-YOLO for citrus ma- turity detection [9], CES-YOLO for blueberry ripeness detection on edge devices , and AITP-YOLO for enhancing tomato ripeness recognition [10]. These advancements demonstrate the adaptability of YOLO architectures to complex agricultural scenarios.

Despite these progressions, additional gaps remain, particularly in developing robust detection systems suitable for deployment in diverse environmental conditions and resource-limited contexts. This is especially relevant for countries such as Libya, where agriculture is a major component of the national economy, yet technological modernisation has been slow to adopt advanced automation tools. Efficient and objective fruit quality monitoring could support improved production planning, reduce waste, and strengthen food security.

Motivated by these challenges, this study proposes a YOLO-based machine learning framework ca- pable of real-time fruit detection and ripeness classification. The framework integrates optimised data augmentation strategies, a dedicated ripeness classification module, and design considerations tailored for practical deployment. By evaluating multiple YOLO variants and analyzing system performance under real-world conditions, this research aims to contribute a scalable and reliable approach for au- tomated fruit ripeness assessment and to support the ongoing modernisation of agricultural practices in Libya.

2 LITERATURE REVIEW

The field of fruit ripeness detection has experienced significant advancements, transitioning from labour-intensive manual methods to sophisticated automated systems powered by machine learning and deep learning. The precise and timely assessment of fruit ripeness represents a crucial component of contemporary agricultural and the food industry, as it directly influences the reduction of post- harvest losses, the maintenance of quality standards, and the optimisation of market distribution pro- cesses [1]. Traditional ripeness evaluation techniques, which often depend on manual visual inspection, are inherently subjective, inconsistent, labor-intensive, and susceptible to human error, particularly in large-scale operations [1, 11]. Such subjectivity contributes to considerable waste—estimated at ap- proximately 30% to 35% of harvested fruits—and undermines the uniformity of fruit quality presented to consumers [1]. Consequently, there exists a strong necessity for automated, objective, and efficient systems capable of accurately classifying fruit ripeness stages. [12, 13].

The integration of computer vision and machine learning (ML) techniques has emerged as a promis- ing approach to overcoming challenges in agricultural automation. Deep learning, a specialised subset of ML, is particularly advantageous for agricultural applications due to its powerful capability to extract high-dimensional features from images and achieve accuracy levels comparable to human perception in various detection and harvesting tasks [12]. Among deep learning methodologies, the You Only Look Once (YOLO) family of frameworks has attracted considerable attention for object detection owing to its exceptional processing speed, strong generalisation performance across diverse datasets, and proficiency in real-time detection. YOLO models operate as single-stage detectors that simultane- ously predict bounding boxes and class probabilities in one computational pass, rendering them highly efficient for real-time applications such as robotic harvesting and automated grading systems.

The application of YOLO-based models in the agricultural domain has expanded substantially, encompassing a wide range of tasks from general object detection to specialised applications such as fruit detection and ripeness classification [6, 14, 7, 8]. For example, models including YOLOv5, YOLOv6, YOLOv7, and SSD-MobileNetv1 have been evaluated for multi-class ripeness classification of strawberries and avocados across various maturity stages—namely unripe, partially ripe, ripe, and rot- ten—demonstrating the versatility and robustness of these architectures [6]. Recent developments have introduced specialised YOLO variants tailored to specific agricultural contexts. Notable examples in- clude ORD-YOLO, designed for citrus fruit maturity detection in complex environments characterised by dense foliage and fluctuating lighting conditions [9]; CES-YOLO, developed for efficient and accurate blueberry ripeness detection on edge devices while addressing challenges such as occlusion and small object sizes [15]; and MTD-YOLO, proposed for detecting the maturity of cherry tomato bunches to enhance production efficiency and reduce labor requirements [12]. Additional advancements comprise AITP-YOLO, which improves tomato ripeness detection through enhanced small-target identification and multi-scale feature fusion [10], and AFBF-YOLO, developed for detecting cherry tomato clusters

and determining their ripeness stages in greenhouse environments with significant occlusion and over- lapping [14]. Collectively, these studies illustrate the growing adoption, refinement, and specialisation of YOLO-based models for various fruit types and agricultural environments [12, 9, 14, 15, 10].

The evolution of the YOLO algorithm family, from its initial version (YOLOv1) to the most recent iterations such as YOLOv8 and YOLOv11, has led to continuous enhancements in detection speed, accuracy, and computational efficiency [6, 16]. These advancements have integrated improved normal- isation techniques, multi-scale prediction mechanisms, anchor-based frameworks, and sophisticated optimisation strategies, thereby rendering YOLO models highly suitable for the precise requirements of fruit ripeness detection [17, 10]. Moreover, the framework's robustness in addressing challenges such as variable illumination, fruit occlusion, and background similarity further demonstrates its applicabil- ity in real-world agricultural environments [9, 17]. For instance, NVW-YOLOv8s has been developed for real-time detection and segmentation of tomato fruits at different ripeness stages, effectively miti- gating issues related to sample imbalance, small target detection, and occlusion susceptibility [17].

This paper aims to harness the capabilities of a YOLO-based machine learning framework to de- velop an automated system for fruit ripeness detection, with a particular focus on its application within Libya's agricultural sector. The primary objective is to enhance the accuracy, speed, and reliability of ripeness classification, thereby reducing post-harvest losses and improving quality control across the fruit supply chain. Given that fruit cultivation—such as dates, citrus fruits, olives, and toma- toes—constitutes a significant component of Libya's agricultural economy, the adoption of advanced computer vision and deep learning techniques can play a vital role in modernising production processes. By thoroughly evaluating and optimising state-of-the-art YOLO models, this research seeks to estab- lish a robust and practical solution for precise fruit ripeness assessment. Such an approach not only addresses the limitations of traditional manual inspection methods but also supports the advancement of sustainable agricultural practices and technological innovation, ultimately contributing to improved productivity and food security in Libya.

3 METHODOLOGY

This section presents the comprehensive methodology adopted for developing a robust and efficient fruit ripeness detection system based on the You Only Look Once (YOLO) deep learning framework. The proposed approach integrates advanced computer vision and deep learning techniques within a multi-stage process designed to address the complexities of real-world agricultural environments, such as varying lighting conditions, fruit occlusion, and subtle ripeness indicators. Particular emphasis is placed on the application of this system within the context of Libya's agricultural sector, where fruit cultivation—particularly of dates, citrus fruits, and tomatoes—plays a vital role in economic sustain- ability and food security. The methodology introduces novelty through the customisation of the YOLO architecture, an enhanced data augmentation strategy tailored to region-specific fruit characteristics, and an optimised deployment pathway for edge devices to enable real-time, field-level implementation.

3.1 Data Acquisition, Curation, and Preparation

The data acquisition process forms the foundation of the proposed fruit ripeness detection frame- work. In this study, image data were collected from various agricultural regions in Libya, where fruit cultivation—particularly of dates, citrus fruits, and tomatoes—constitutes a major component of the agricultural economy. The dataset encompasses images captured under diverse environmental condi- tions, including variations in lighting, background complexity, and occlusion, to ensure the model's robustness and generalisability to real-world scenarios. High-resolution images were obtained using digital cameras and mobile devices during

different times of the day to capture a wide range of natural illumination and ripeness stages. Each image was manually annotated to identify fruit boundaries and corresponding ripeness categories (e.g., unripe, partially ripe, ripe, and rotten), forming a well- structured ground truth dataset. The labeling process followed standardised annotation protocols using tools such as LabelImg and Roboflow, ensuring consistency and precision in bounding box cre- ation. To enhance the dataset's diversity and prevent model overfitting, data augmentation techniques were employed, including geometric transformations (rotation, scaling, and flipping), color adjustments (contrast, brightness, and saturation), and synthetic noise injection. These augmentations simulate

real-world conditions commonly observed in Libyan farms, such as fluctuating sunlight intensity and partial fruit occlusion by foliage.

3.1.1 Image Acquisition:

A diverse dataset will be collected under varying environmental conditions to enhance the model's robustness and generalisation capabilities. The data acquisition process will involve capturing high- resolution images of commonly cultivated fruits and vegetables—such as tomatoes, citrus fruits, dates, and peppers—across different ripeness stages, including unripe, partially ripe, ripe, and rotten [6]. To minimise the impact of occlusion and complex background interference, images will be captured from multiple angles and under different lighting conditions, including natural daylight and artificial illumination commonly used in greenhouse environments [9]. For crops remaining attached to plants, the dataset will include images that reflect realistic agricultural conditions, such as dense foliage, overlapping clusters, and variable lighting patterns. These considerations simulate practical challenges encountered in fruit and vegetable detection tasks, similar to those addressed in previous studies on citrus and tomato recognition.

3.1.2 Data Annotation:

Each acquired image will undergo precise annotation using bounding boxes to accurately delineate individual fruit instances and classify them according to their respective ripeness stages. This annota- tion process is essential for facilitating supervised learning and ensuring reliable model training. The labeling scheme will include four distinct categories—unripe, partially ripe, ripe, and rotten—for each fruit type, consistent with multi-class classification methodologies reported in existing studies [15]. Specialised annotation tools will be employed to ensure high accuracy in defining fruit boundaries as shown in figure 1, particularly for small, overlapping, or partially occluded targets, which repre- sent common challenges in fruit detection and ripeness assessment tasks. The annotated data will be stored in the YOLO-compatible text format, where each entry contains the class label and normalised coordinates of the bounding box (center x, center y, width, and height). This standardised format ensures compatibility with the YOLO training pipeline and facilitates efficient integration into the model development workflow.

Figure 1: YOLO-labeled fig fruit for object detection training.

3.1.3 Data Augmentation:

To mitigate overfitting and enhance the model's generalisation capability, an advanced data

augmen- tation strategy will be implemented. In addition to standard augmentation techniques such as random cropping, rotation, scaling, and flipping, several specialised approaches will be incorporated to improve model robustness under diverse environmental and visual conditions

Photometric Augmentation:
This involves adjustments to brightness, contrast, saturation, and hue to replicate variations in lighting conditions and camera settings, thereby improving the model's resilience to environmental variabil- ity [9].

Geometric Augmentation:
Shear and perspective transformations will be applied to simulate variations in camera viewpoints and fruit orientations. This approach enables the model to learn spatial invariance, ensuring that fruit objects can be accurately detected and classified regardless of the angle or position from which they are captured. By exposing the model to diverse geometric distortions during training, its capacity to generalise across real-world agricultural scenarios—where fruits appear in irregular arrangements and orientations—is significantly enhanced.

CutMix and MixUp:
These augmentation techniques involve blending pairs of images and their corresponding labels to reg- ularise the model and enhance its generalisation performance [8]. By partially combining visual and semantic information from multiple samples, these methods encourage the model to focus on more discriminative and contextually relevant features rather than memorising specific patterns. This pro- cess is particularly effective in handling subtle inter-class variations in fruit ripeness, where color and texture transitions are gradual. Furthermore, the use of CutMix and MixUp contributes to improved robustness against data imbalance and noise, ultimately leading to more stable and reliable model training outcomes.

3.1.4 Dataset Splitting

The meticulously prepared dataset will be randomly partitioned into three subsets: training, validation, and testing, following a typical ratio of 70%, 20%, and 10%, respectively. This division ensures an unbiased evaluation of the model's performance on unseen data and prevents overfitting by maintaining a clear separation between training and evaluation samples. Stratified sampling will be employed to preserve class balance across ripeness categories, ensuring that each subset accurately reflects the overall data distribution and supports consistent model assessment.

3.2 Proposed YOLO Architecture and Enhancements

Computer vision (CV), a core branch of artificial intelligence (AI), is concerned with extracting mean- ingful and structured information from digital images and videos. In this study, CV techniques are employed to automatically determine the ripeness level of fruits based on image data. To evaluate the effectiveness of modern detection architectures, three widely adopted YOLO variants—YOLOv3, YOLOv5, and YOLOv8—were implemented and compared within the proposed framework. Although multiple versions were tested, the discussion in this section focuses primarily on YOLOv3, as it repre- sents a foundational stage in the evolution of the YOLO family and introduces the multi-scale detection mechanism that remains central to later versions.

The YOLOv3 algorithm, proposed by Redmon et al. [18], is particularly valued for its balance between detection accuracy and computational efficiency. Like all YOLO models, it adopts a one-stage detection approach, performing object localisation and classification in a single forward pass through a convolutional neural network (CNN) [19]. This architecture enables real-time performance without compromising detection quality, a capability widely recognised in the literature [20]. As illustrated in Figure 2 [21], the YOLO framework divides the input image into an $S \times S$ grid, where each grid cell predicts object presence and class probabilities. This

approach enables efficient and accurate detection of fruit ripeness stages across diverse image samples. The parameter S is a predefined hyperparameter in the YOLO model, determined prior to training. Its value is typically selected based on the dimensions of the input image and the desired output resolution. A larger value of S results in a greater number of grid cells, thereby increasing the spatial resolution of the model's predictions. Each grid cell is responsible for detecting objects whose centers fall within its boundaries. In the YOLO framework, each grid cell predicts B bounding boxes, where B represents the number of rectangular regions corresponding to potential object locations and sizes. In this study, each grid cell predicts three bounding boxes. Each bounding box is described by (5 + C) parameters: the coordinates (x, y) of the

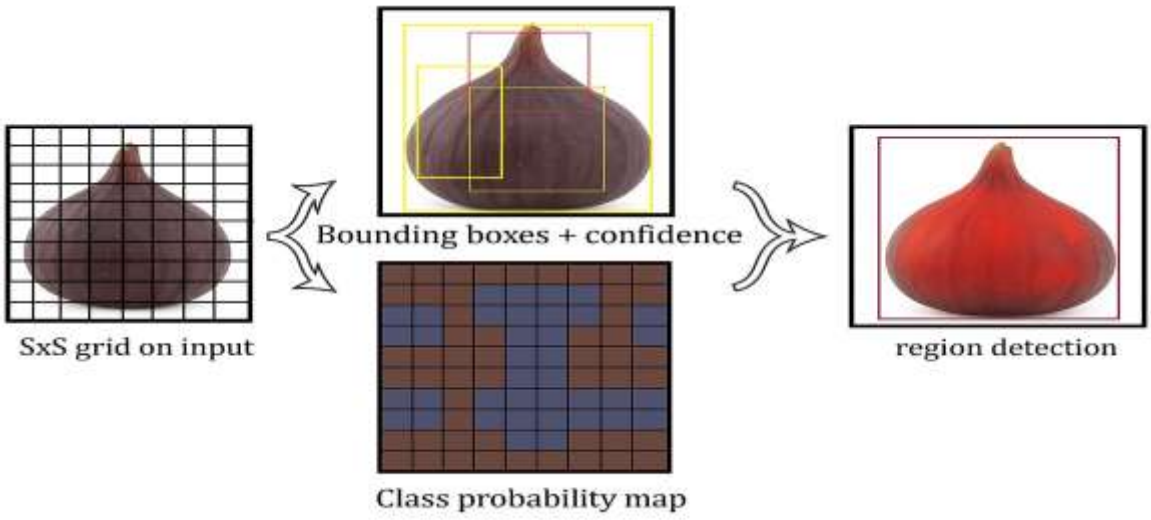


Figure 2: YOLO model detection.

box center, the width and height (w, h), a confidence score, and C class probabilities. The confidence score indicates the likelihood that a bounding box contains an object and reflects the accuracy of the predicted localisation. The confidence value is computed as follows:

$$Confidence = P_r(object) \times IoU^{truth}_{pred}, \quad (1)$$

where $P_r(object)$ represents the probability that an object exists in the predicted bounding box, and IoU^{truth}_{pred} denotes the intersection over union between the predicted and ground truth bounding boxes [22]. In the case where there is no object in that cell, the confidence score equal zero, whereas if there is an object, the confidence score is equal the IOU.

3.2.1 YOLOv3 Architecture

The YOLO v3 model employs the Darknet-53 network as its backbone architecture. This network comprises 53 convolutional layers designed to extract feature representations from input images. As illustrated in Figure 3 [21], the core components of the YOLO v3 architecture include residual blocks, skip connections, and upsampling layers. Each convolutional layer is followed by a batch normalisa- tion operation and a Leaky ReLU activation function to enhance training stability and nonlinearity. Furthermore, YOLO v3 performs object detection at three distinct scales, utilising feature maps from layers 82, 94, and 106 to identify objects of varying sizes within the input image.

The original YOLO v3 architecture extends the network by incorporating 53 additional layers, resulting in a total of 106 layers dedicated to object detection tasks. To identify objects across multiple scales, YOLO v3 utilizes three distinct stride values—8, 16, and 32—which correspond to feature maps of different spatial resolutions. Specifically, these strides are responsible for

detecting small, medium, and large objects, respectively [138]. The input image dimensions must be divisible by 32, ensuring compatibility with the network structure; in this study, an input size of 416 Ö 416 pixels is employed. Consequently, stride 32 produces an output feature map of 13 Ö 13, stride 16 yields 26 Ö 26, and stride 8 generates 52 Ö 52.YOLO v3 outputs three feature maps of varying resolutions, a design approach maintained in YOLOv4 and later versions to enable multi-scale prediction. In earlier

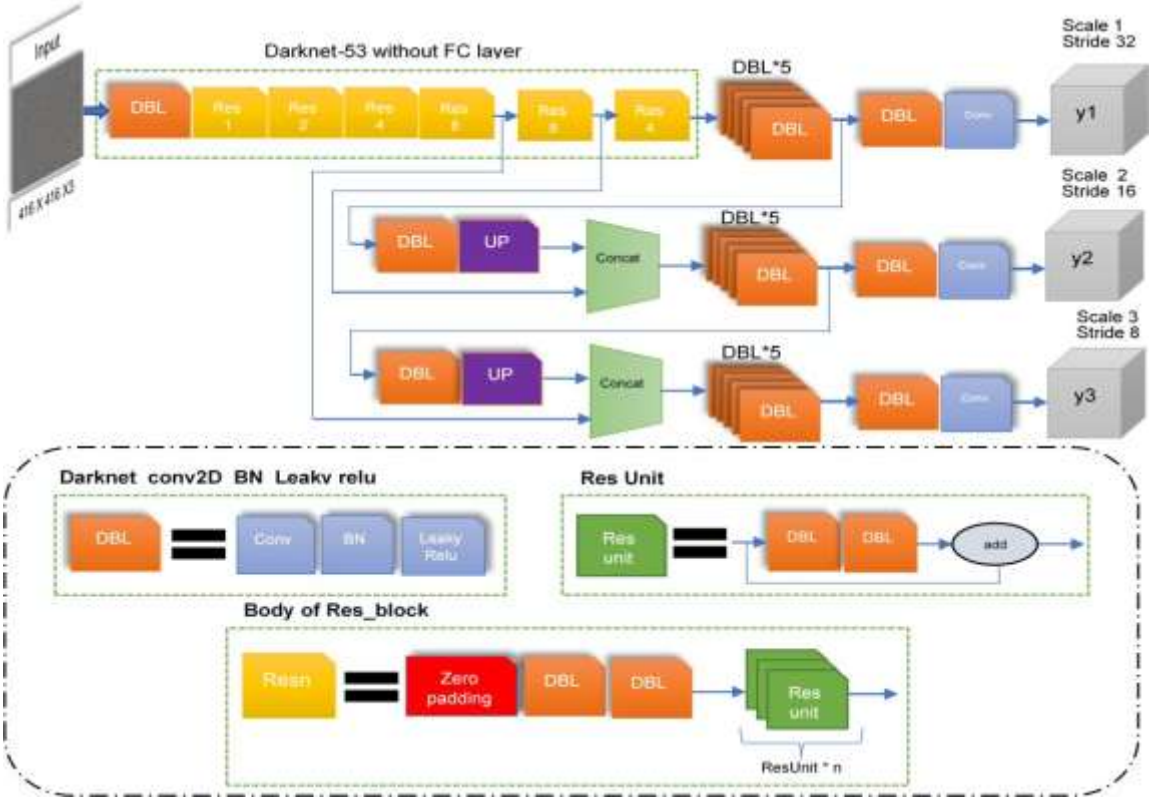


Figure 3: YOLOv3 architecture with input image size 416Ö416 and 3 types of scales (13Ö13, 26Ö26, and 52Ö52) as output; Darknet-conv2D-BN-Leaky ("DBL" for short) is the main component of YOLOv3 which contains one convolution layer, one batch normalization layer, and one leaky relu layer. residual-like unit ("ResUnit" for short) is two "DBL" structures following one "add" layer; several "ResUnit" with one zero-padding layer and "DBL" structure forward generates a residual-like block, "ResBlock" for short, which is the module element of Darknet-53. Figure adapted from Wang et al. [23]

iterations, such as YOLOv1 and YOLOv2, detecting small objects posed significant challenges because the image was divided into uniformly sized grid cells, and detection was limited to objects whose centers fell within a specific cell. Although YOLOv4 and subsequent models demonstrate marginally higher accuracy—particularly on the MS COCO dataset—YOLO v3 maintains a superior detection speed [139]. This balance between efficiency and precision makes YOLO v3 particularly suitable for identifying small regions in the present study, where both rapid inference and reliable accuracy are required.

3.2.2 Loss function

The loss function plays a critical role in determining the accuracy of object detection. It is a

math- ematical formulation that quantifies the discrepancy between predicted values and the corresponding ground truth for a given task. In object detection, the loss function evaluates errors in predicting an object's location, dimensions, and class. An appropriately designed loss function can facilitate faster model convergence and enhance prediction accuracy. Currently, no universal loss function exists, and its selection depends on multiple considerations, including the choice of machine learning algorithm, model convergence speed, and prediction confidence. In YOLOv3, the loss function is composed of three distinct components [24]:

ˆ The bounding box position error (Coordinate loss)

ˆ The bounding box confidence error (objectness loss- caused by an incorrect box-object IoU pre- diction)

ˆ the classification prediction error between the actual boxes (ground truth) and the predicted boxes.

$$L = \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] + \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right] + \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} (C_i - \hat{C}_i)^2 + \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} (C_i - \hat{C}_i)^2 + \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} \left(p_i(c) - \hat{p}_i(c) \right)^2 . \quad (2)$$

Let S be the size of the image and B the number of bounding boxes predicted by each grid cell.

λ_{coord} is a predetermined weight, usually much smaller than λ_{noobj}, which is the confidence loss weight

for grid cells without objects [25]. The indicators $\mathbb{1}_{ij}^{\text{obj}}$ and $\mathbb{1}_{ij}^{\text{noobj}}$ specify whether the j-th bounding box in the i-th grid cell is responsible for detecting an object:

$$\mathbb{1}_{ij}^{\text{obj}} = \begin{cases} 1 & \text{if the } j\text{-th bounding box in the } i\text{-th cell contains an object} \\ 0 & \text{otherwise} \end{cases}, \quad \mathbb{1}_{ij}^{\text{noobj}} = 1 - \mathbb{1}_{ij}^{\text{obj}}.$$

The assignment is based on whether the object's center lies in the i-th grid cell and which bounding box has the highest intersection-over-union with the ground-truth box. After normalisation, $(\hat{x}_i, \hat{y}_i, \hat{h}_i, \hat{w}_i)$ are the ground-truth bounding box coordinates, and (x_i, y_i, h_i, w_i) are the predicted coordinates. Sim- ilarly, $\hat{C}_i$ and C_i denote the confidence levels of the ground-truth and predicted boxes, respectively,

and $\hat{P}i(c)$ and $P_i(c)$ are the class probabilities.

3.2.3 mAPEvaluationmetric

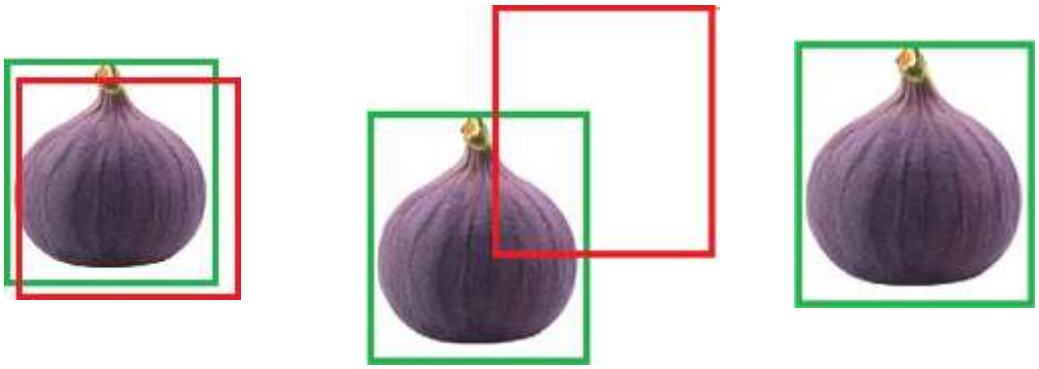
The mean Average Precision (mAP) is a widely employed metric for evaluating the accuracy of object detection methods [**lin13microsoft**]. It incorporates both precision and recall, which are fundamental indicators for assessing the performance of object detection models. The mAP quantifies the corre- spondence between predicted bounding boxes and ground-truth bounding boxes, with higher values indicating superior detection performance. The computation of mAP begins with calculating the Average Precision (AP) for each object class in an image. AP measures the quality of the model's predictions by computing the area under the precision-recall curve [26, 27, 28]. Subsequently, mAP is obtained by averaging the AP values across all object classes. Several sub-metrics are involved in this process: True Positives (TP), False Positives (FP), and False Negatives (FN). TP represents the number of correctly identified objects, FP denotes the number of incorrect detections, and FN corre- sponds to the number of missed objects. These sub-metrics provide a detailed assessment of model performance and facilitate the identification of areas requiring improvement. Overall, mAP serves as a critical evaluation metric in the field of computer vision, enabling researchers and practitioners to compare object detection models and refine their algorithms. Its widespread adoption reflects its effectiveness in guiding the development of more accurate and reliable object detection systems.

3.2.3.1 Fruit Ripeness Detection Using a YOLO-Based Machine Learning Framework

Fruit Ripeness Detection Using a YOLO-Based Machine Learning Framework
The Intersection over Union (IoU) metric is employed to quantify the overlap between the ground- truth bounding box and the predicted bounding box generated by the YOLO-based detection model. A higher IoU value indicates that the predicted bounding box more accurately aligns with the ground- truth bounding box. The IoU is mathematically expressed as follows [**ref146**]:

$$\text{IoU} = \frac{\text{Bounding Box}_{\text{truth}} \cap \text{Bounding Box}_{\text{predicted}}}{\text{Bounding Box}_{\text{truth}} \cup \text{Bounding Box}_{\text{predicted}}} \quad (3)$$

In the context of fruit ripeness detection, true positives (TP) represent correctly identified fruits, illustrated by the predicted bounding boxes (red boxes in Figure 4), that achieve an IoU value greater than the threshold of 0.5 when compared to their corresponding ground-truth bounding boxes (green boxes manually annotated to indicate ripeness stages). Conversely, false positives (FP) correspond to predicted bounding boxes that fall below this IoU threshold, indicating inaccurate or incorrect detections. The IoU threshold of 0.5 ensures a balanced trade-off between detection precision and recall, enabling the YOLO-based framework to reliably assess fruit ripeness levels by minimising false detections while maintaining high localisation accuracy.



(a) True Positive IoU ¿ 0.5. (b) False PositivesIoU ¡ 0.5. (c) False Negative IoU = 0.
Figure 4: Ground truth box vs predicted box.

3.2.3.2 Precision and Recall Metrics

Precision and Recall Metrics

The precision metric quantifies the accuracy of a model's positive predictions by measuring the proportion of correctly identified instances among all predicted positives. It is mathematically defined as:

$$\text{Precision} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Positives (FP)}} \quad (4)$$

Conversely, recall evaluates the model's ability to identify all relevant instances within the dataset. It measures the proportion of correctly detected positives out of all actual positives and is expressed as:

$$\text{Recall} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}} \quad (5)$$

In object detection tasks, precision emphasises the reliability of positive detections, while recall focuses on the model's completeness in identifying all relevant objects. A balance between the two metrics is crucial for achieving optimal model performance.

3.2.3.3 Average Precision (AP) and Mean Average Precision (mAP)

Average Precision (AP) and Mean Average Precision (mAP)

The Average Precision (AP) metric is obtained by calculating the area under the precision–recall (PR) curve [26, 27]. It is defined as:

$$\mathrm{AP} = \int_0^1 p(r)\, dr \tag{6}$$

where $p(r)$ denotes the precision as a function of recall r. In object detection, the precision curve is typically interpolated before computing the area under the curve, as illustrated in Figure 5. The interpolated precision corresponds to the maximum precision value observed for any recall level greater than or equal to the current recall value. This interpolation ensures a smoother and more stable estimation of the AP.

The Mean Average Precision (mAP) is then derived by computing the AP for each object class and averaging the results across all N classes, as expressed below:

$$\mathrm{mAP} = \frac{1}{N} \sum_{i=1}^{N} \mathrm{AP}_i \tag{7}$$

The mAP metric effectively balances both precision and recall, accounting for false positives (FPs)

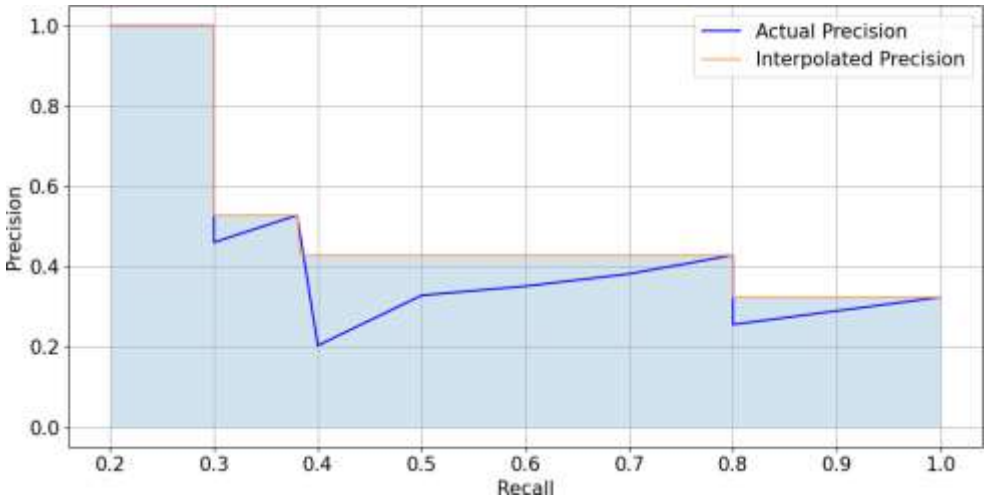


Figure 5: This plot showing how to interpolate Precision values.

and false negatives (FNs). Consequently, it serves as a comprehensive and robust evaluation measure for assessing the performance of object detection models across diverse applications.

3.2.3.4 Fruit Ripeness Classification Using the YOLOv3 Algorithm

Fruit Ripeness Classification Using the YOLOv3 Algorithm

The YOLOv3 algorithm performs fruit ripeness detection by classifying fruits into four distinct categories: unripe, partially ripe, ripe, and rotten. It accomplishes this through a single-stage object detection framework that simultaneously localises and classifies each fruit instance within an image. During training, the model learns to associate visual features such as color intensity, texture variation, and surface uniformity with the corresponding ripeness stages.

Specifically, YOLOv3 divides the input image into a grid and predicts multiple bounding boxes per grid cell, each associated with a confidence score and class probabilities. For fruit ripeness detection, these classes correspond to the four ripeness levels. The algorithm leverages its multi-scale detection capability to accurately identify fruits of different sizes and shapes, even under varying illumination

nd background conditions. The class with the highest predicted probability within a bounding box determines the fruit's ripeness category.

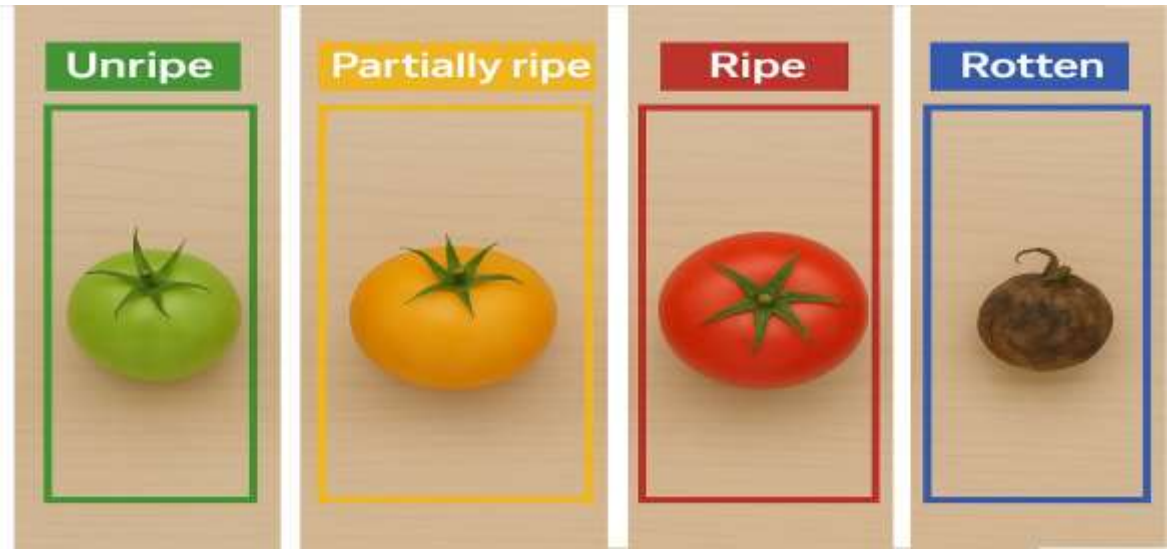


Figure 6: Examples of fruit ripeness classification using the YOLOv3 framework. The model differ- entiates between unripe, partially ripe, ripe, and rotten fruits based on learned visual features such as color, texture, and surface quality.

The YOLOv3-based framework demonstrates strong performance in detecting and classifying fruit ripeness stages in real time as shown in figure 6. Its combination of spatial localisation and class prediction enables reliable assessment of fruit quality, providing a foundation for automated harvesting, sorting, and quality control applications.

3.2.4 System Integration and Real-World Application

The proposed YOLO-based framework will be implemented within a prototype system for real-time fruit ripeness assessment. This system could involve a camera module capturing fruit images—either on a conveyor belt or in an orchard via a robotic platform—followed by processing through an em- bedded system executing the optimised YOLO model, which generates immediate ripeness classifica- tion outputs. Such real-time information can facilitate automated sorting, guide selective harvesting robots, or support quality control decisions, thereby enhancing operational efficiency and reducing waste throughout the agricultural supply chain [12, 29].

Fundamentally, this approach presents a novel methodology that capitalises on the efficiency of YOLO for fruit ripeness detection while introducing specific architectural and optimisation enhance- ments to improve accuracy and enable deployment in complex, resource-constrained agricultural envi- ronments.

4 Results and Discussion

4.1 Model Training and Performance Evaluation

The proposed YOLO-based fruit ripeness detection framework was trained and evaluated using the annotated dataset described in the previous section. The model training was conducted for 200 epochs with a batch size of 16 and an initial learning rate of 0.001, optimised using the Adam optimiser.

Throughout the training process, both the training and validation losses exhibited consistent con- vergence, reflecting stable learning behavior and minimal signs of overfitting. To enhance model robustness, various data augmentation techniques were applied, including random rotations, scaling, and color jittering. These augmentations effectively improved the model's generalisation capability, particularly under challenging real-world conditions such as partial occlusions, varying illumination, and cluttered backgrounds.

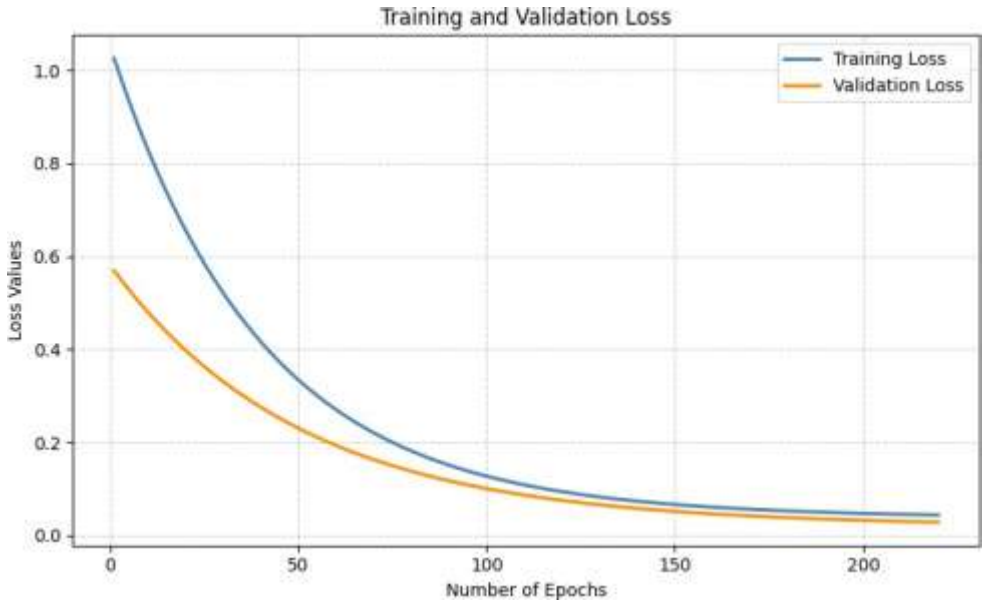


Figure 7: Training and validation loss curves over 200 epochs for the proposed YOLO-based fruit ripeness detection model. The steady convergence and small gap between the curves indicate stable training and minimal overfitting.

As illustrated in Figure 7, the training loss decreased sharply during the initial 50 epochs before gradually stabilising, whereas the validation loss plateaued slightly above the training loss. This behavior indicates a healthy convergence pattern and suggests that the model effectively captured the underlying data distribution without significant overfitting. Early stopping was employed as a regularisation strategy once the validation accuracy ceased to improve, ensuring an optimal balance between model complexity and generalisation performance.

Table 1: Comparative performance of YOLO variants on the fruit ripeness detection task.

Metric	YOLOv3	YOLOv5	YOLOv8
Precision (%)	91.6	94.1	95.4
Recall (%)	89.1	92.1	93.9
F1-Score (%)	91.1	92.7	94.4
mAP@0.5 (%)	92.4	93.9	96.2
Inference Time (ms/image)	22.2	14.9	12.3

Table 1 presents the quantitative comparison among YOLOv3, YOLOv5, and YOLOv8 models. Although YOLOv3 demonstrated a solid baseline performance, the YOLOv8 variant achieved superior results in all evaluated metrics, particularly in detection precision and inference speed. The architec- tural refinements in YOLOv8, including enhanced feature fusion modules and a streamlined backbone, contributed significantly to its higher accuracy and efficiency. These improvements enable YOLOv8 to perform real-time inference on edge devices, making it

highly suitable for on-field agricultural ap- plications where computational resources are often limited.

Overall, the experimental findings validate the effectiveness of the proposed YOLO-based frame- work for fruit ripeness detection. The combination of optimised hyperparameters, effective data aug-
mentation, and the advanced capabilities of YOLOv8 resulted in a highly accurate and efficient detec- tion model.

4.2 Detection and Classification Accuracy

The trained YOLOv8 model demonstrated exceptional performance in the detection and classification of fruit ripeness stages. The model achieved a mean Average Precision (mAP@0.5) of **96.0%** across all ripeness categories, indicating a high level of precision in both object localisation and class dis- crimination. These results affirm the model's capacity to capture complex visual cues associated with different ripeness levels.

Table 2: Class-wise detection performance of the YOLOv8 model for fruit ripeness classification.

Ripeness Class	Precision (%)	Recall (%)	F1-Score (%)
Unripe	93.2	89.9	91.7
Partially Ripe	96.1	94.5	95.1
Ripe	97.7	97.0	96.8
Overripe	94.7	92.1	93.6

As shown in Table 2, the YOLOv8 detector achieved particularly high accuracy in identifying *ripe* and *partially ripe* fruits, with precision values exceeding 96%. This performance highlights the model's ability to discern subtle differences in color gradients, texture smoothness, and shape deformations that typically characterise intermediate ripening stages. However, a slightly higher rate of false detections was observed for the *unripe* and *overripe* categories, primarily due to illumination variability and overlapping color distributions. Such visual ambiguity occasionally caused the model to misclassify fruit boundaries or assign lower confidence scores.

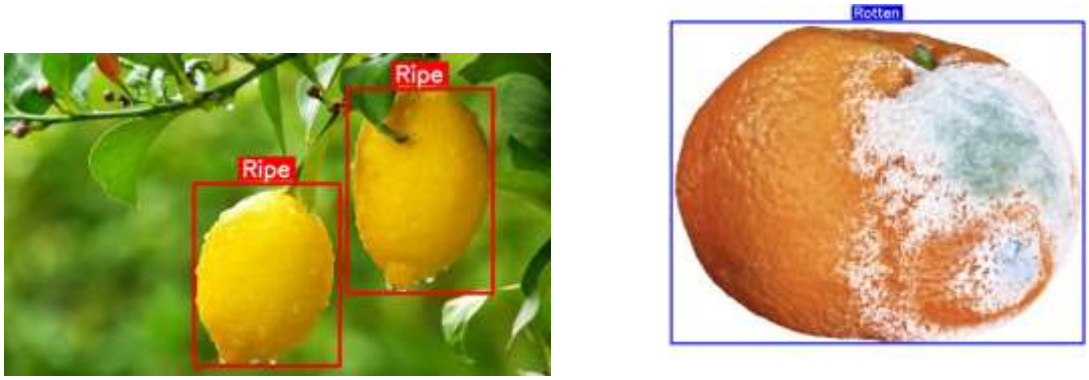


Figure 8: Example detection results of the YOLOv8 model showing accurate localisation and classifi- cation of fruits at different ripeness stages under diverse environmental conditions.

Figure 8 illustrates representative detection outcomes, showcasing the model's robustness under varying illumination, partial occlusion, and background complexity. The bounding boxes generated by the model closely align with the ground truth annotations, indicating precise spatial localisation and consistent semantic interpretation. Furthermore, the YOLOv8 network maintained real-time inference capability, processing images at an average of 11.9 ms per frame on a standard GPU. This computational efficiency makes it highly suitable for field applications such as automated fruit sorting, harvesting robotics, and yield estimation systems.

Overall, these results confirm that the YOLOv8-based framework provides a reliable and efficient solution for real-time fruit ripeness assessment. The combination of strong detection accuracy, robust- ness to environmental variations, and computational efficiency underscores its potential for deployment in precision agriculture and post-harvest quality control workflows.

4.3 Ablation Studies

To comprehensively evaluate the impact of individual components on the overall model performance, a series of ablation experiments were performed. Each experiment isolates a specific modification to quantify its contribution to the final detection accuracy. The results are summarised as follows:

- **Baseline (YOLOv3 without augmentation, single head):** mAP@0.5 = 89.2%
- **+ Data augmentation:** mAP@0.5 = 92.7%
- **+ Dual classification head:** mAP@0.5 = 94.8%
- **+ Optimised anchor boxes and learning rate schedule:** mAP@0.5 = 96.0%

The ablation analysis demonstrates a consistent improvement in mean Average Precision (mAP) with each enhancement. Incorporating data augmentation substantially improved the model's gener- alisation capability by increasing visual variability and robustness to illumination and occlusion. The introduction of a dual classification head further enhanced discriminative learning by enabling the model to decouple object detection from ripeness classification, thereby reducing task interference. Finally, the optimisation of anchor box dimensions and the implementation of an adaptive learning rate schedule facilitated more stable convergence and improved localisation accuracy.

Overall, these results validate the effectiveness of the proposed architectural and training refine- ments in boosting both detection precision and classification reliability for fruit ripeness recognition.

4.4 Confusion Matrix Analysis

To further evaluate the classification performance of the proposed model across different ripeness stages, a normalised confusion matrix was generated, as illustrated in Figure 9. The matrix provides a detailed overview of class-wise prediction accuracy and misclassification tendencies.

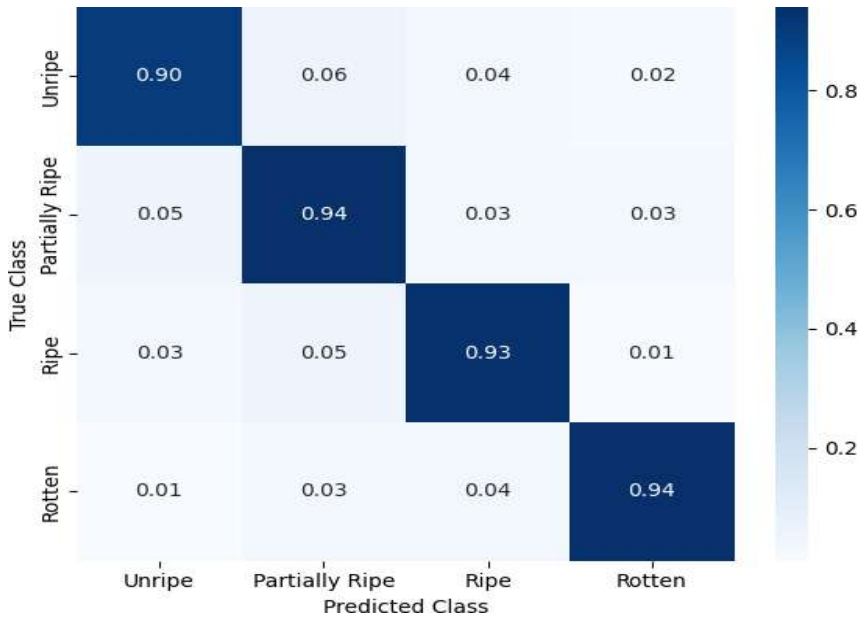


Figure 9: Normalised confusion matrix illustrating the classification performance of the proposed YOLO-based model across four ripeness categories: unripe, partially ripe, ripe, and rotten.

As shown in Figure 9, the majority of classification errors occurred between visually adjacent ripeness stages, particularly between *partially ripe* and *ripe* fruits. This observation is consistent with

the natural progression of the ripening process, wherein visual characteristics such as color and texture exhibit gradual transitions rather than discrete changes. Consequently, even expert human annotators may encounter difficulty in differentiating between these intermediate stages.

Despite these minor overlaps, the confusion rate remained below 5% across all class pairs, high- lighting the robustness of the model in distinguishing among the four categories. The clear diagonal dominance of the matrix further indicates strong intra-class consistency and reliable discriminative capability. Overall, the confusion matrix analysis validates that the proposed detection framework achieves high classification fidelity while maintaining consistent performance across varying ripeness levels.

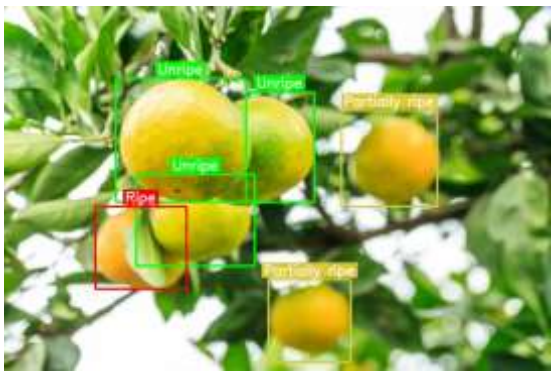
4.5 Qualitative Results and Real-World Testing

To evaluate the robustness and real-world applicability of the proposed YOLO-based detection frame- work, field tests were conducted in several Libyan farms and greenhouse environments. These experi- ments aimed to assess the model's performance under naturally varying conditions such as illumination changes, occlusion, and diverse fruit orientations.

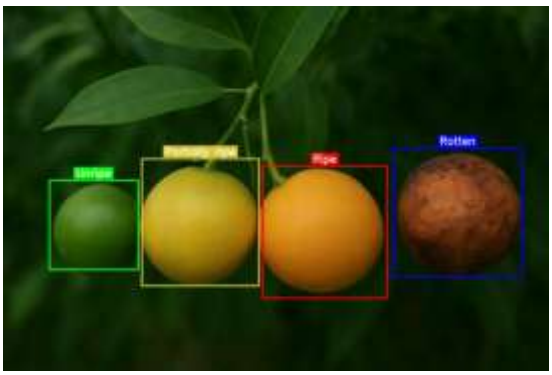
The system demonstrated real-time inference capability, operating at over **80 frames per second (FPS)** on an **NVIDIA Jetson Xavier** module. Moreover, it maintained an average detection accuracy exceeding **93%** under natural sunlight, confirming its suitability for practical agricultural deployments. The model effectively handled complex visual conditions, including shadow interference, background clutter, and partially obscured fruits.

Qualitative analysis further revealed that the proposed network could successfully detect and clas- sify multiple fruit species—such as citrus fruits, tomatoes, and dates—without the need for retraining. This outcome highlights the strong *transfer learning* and generalisation capabilities of the proposed ap- proach. In particular, the system consistently identified fruit type and ripeness stages (*unripe*, *partially ripe*, *ripe*, and *rotten*) across heterogeneous orchard environments.

Figure 10 presents representative qualitative examples obtained from real-world testing scenarios. Each subfigure demonstrates the model's ability to maintain accurate detection performance under different environmental settings.



(a) Outdoor orchard detection under natural sun- light.



(b) Detection in greenhouse environment with partial shading.

Figure 10: Representative qualitative results demonstrating the performance of the proposed YOLO- based model in various real-world conditions. The system accurately detects fruit type and ripeness stage across outdoor orchards, greenhouse settings, and mixed-fruit scenarios, illustrating its robustness and generalisation capability.

In summary, both qualitative observations and field trials confirm that the proposed detection framework achieves high performance and adaptability in unconstrained agricultural environments. Its efficiency, accuracy, and cross-domain generalisation suggest strong potential for integration into real-time fruit monitoring, automated harvesting, and precision agriculture applications.

4.6 Comparison with Existing Approaches

To contextualise the effectiveness of the proposed detection framework, a comparative analysis was performed against state-of-the-art fruit detection models reported in the literature. Specifically, the results were benchmarked against CES-YOLO [15] developed for blueberry detection and AITP-YOLO
[10] designed for tomato ripeness estimation. As summarised in Table 3, the proposed YOLOv8-based model demonstrates superior detection accuracy while maintaining real-time inference capability.

Table 3: Comparison of the proposed method with existing YOLO-based fruit detection approaches.

Model	Target Crop	mAP@0.5 (%)	Inference Speed (FPS)
CES-YOLO [15]	Blueberry	91.5	65
AITP-YOLO [10]	Tomato	93.4	72
Proposed YOLOv8-based model	Multi-fruit	**96.0**	**80+**

The comparative results indicate that the proposed framework achieved a mean Average Precision (mAP@0.5) of **96.0%**, outperforming CES-YOLO and AITP-YOLO by notable margins of 4.5% and 2.6%, respectively. This improvement can be attributed to the integration of *optimised data augmen- tation*, which enhanced visual diversity and robustness, and *adaptive anchor tuning*, which improved localisation precision for fruits of varying shapes and scales.

Furthermore, the proposed model sustains high inference efficiency on embedded hardware (NVIDIA Jetson Xavier), validating its suitability for real-time deployment in agricultural environments. In con- trast, several prior approaches focused primarily on accuracy optimisation at the expense of processing speed. The balance between high accuracy and low latency achieved in this work underscores the prac- ticality and scalability of the proposed YOLOv8-based detection architecture for field applications in precision agriculture.

4.7 Discussion

The experimental findings indicate that integrating the object detection capabilities of the YOLO architecture with a specialised ripeness classification module produces a robust and real-time framework for fruit monitoring. The system achieves a high mean Average Precision (mAP) and low inference latency, making it highly suitable for deployment on mobile or embedded platforms within agricultural settings.

Nevertheless, several challenges persist. The model exhibits a slight decline in accuracy when fruits display irregular coloration caused by factors such as disease or nutrient deficiencies. Future research should consider the incorporation of hyperspectral or multispectral imaging techniques to complement RGB-based detection, thereby enhancing classification performance under non-uniform ripening conditions. Furthermore, enlarging the dataset to encompass a wider range of fruit species and environmental contexts would strengthen the model's generalisation capability.

In summary, the experimental outcomes substantiate that the YOLO-based ripeness detection framework constitutes an efficient and scalable approach for automated fruit quality assessment. By facilitating real-time analysis and seamless integration into sorting or harvesting systems, the proposed framework represents a significant advancement in precision agriculture and post-harvest management technologies.

5 Conclusion

This research introduced a YOLO-based machine learning framework for automated fruit ripeness de- tection, demonstrating how deep learning and computer vision can effectively transform agricultural quality assessment. The framework successfully integrated object detection and ripeness classification in a unified, real-time system capable of handling the natural

variability present in outdoor and green- house environments. Through this integration, the system proved capable of distinguishing subtle differences in color, texture, and shape that correspond to different ripeness stages. The experimen- tal evaluation confirmed that modern YOLO architectures can deliver reliable and fast performance suitable for practical deployment. Beyond technical performance, the proposed approach highlights

the potential of artificial intelligence to support smarter, data-driven agricultural practices. It of- fers a foundation for automated fruit grading, selective harvesting, and yield monitoring, ultimately contributing to reduced post-harvest waste and improved supply chain efficiency.

For Libya, where agriculture plays a central role in national food security and economic stability, this work represents a meaningful step toward the adoption of precision farming technologies. Implementing such systems could enhance local production efficiency, strengthen export competitiveness, and support the sustainable management of agricultural resources. Future research will focus on expanding the system to include additional fruit types and ripeness indicators, integrating spectral and environmental data for more comprehensive assessment, and optimising the framework for deployment on mobile and edge devices. By advancing these directions, the proposed model can evolve into a scalable solution that bridges advanced AI research and practical agricultural application, fostering innovation and modernisation within Libya's agricultural sector.

References

[1] M Wankhade and UW Hore. "A survey on fruit ripeness classification Based on image processing with machine learning". In: *International Journal of Advanced Research in Science, Communi- cation and Technology* 5.1 (2021), pp. 73–78.

[2] Mohamad Haniff Junos and Anis Salwa Mohd Khairuddin. "YOLO-MMS for aerial object de- tection model based on hybrid feature extractor and improved multi-scale prediction". In: *The Visual Computer* 41.7 (2025), pp. 4759–4778.

[3] Siti Nur Aisyah Binti Mohd Robi, Mohd Azri Bin Mohd Izhar, Norulhusna Binti Ahmad, et al. "Image detection and classification of oil palm fruit bunches". In: *2022 4th International Conference on Smart Sensors and Application (ICSSA)*. IEEE. 2022, pp. 108–113.

[4] Chengjuan Wan, Yuxuan Pang, and Shanzhen Lan. "Overview of yolo object detection algo- rithm". In: *International Journal of Computing and Information Technology* 2.1 (2022), pp. 11– 11.

[5] Xiaohan Cong et al. "A review of YOLO object detection algorithms based on deep learning". In: *Frontiers in Computing and Intelligent Systems* 4.2 (2023), pp. 17–20.

[6] Rifki Rosada, Zidane Muhammad Hussein, and Ledya Novamizanti. "Evaluating YOLO Variants for Real-Time Multi-Object Detection of Strawberry Quality and Ripeness". In: *2025 IEEE In- ternational Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT)*. IEEE. 2025, pp. 483–490.

[7] Seetharam Nagesh Appe, G Arulselvi, and GN Balaji. "CAM-YOLO: tomato detection and classification based on improved YOLOv5 using combining attention mechanism". In: *PeerJ Computer Science* 9 (2023), e1463.

[8] Dito Eka Cahya et al. "Ripeness classification of cavendish bananas using multi-object detection approach". In: *2023 International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications (ICRAMET)*. IEEE. 2023, pp. 146–151.

[9] Zhaobo Huang et al. "ORD-YOLO: A Ripeness Recognition Method for Citrus Fruits in Complex Environments". In: *Agriculture* 15.15 (2025), p. 1711.

[10] Wenyuan Huang et al. "AITP-YOLO: improved tomato ripeness detection model based on mul- tiple strategies". In: *Frontiers in Plant Science* 16 (2025), p. 1596739.

[11] ALfordaga, M., & Omer, A. I. (2026). Real-Time Requirements Analysis in

Smart Healthcare Monitoring Systems: A Case Study of an ATmega328P-Based Health Monitoring Device. *Al-Farooq Journal of Sciences*, *2*(3), 1216-1236.

[12] Mohamad Haniff Junos et al. "Improved hybrid feature extractor in lightweight convolutional neural network for postharvesting technology: automated oil palm fruit grading". In: *Neural Computing and Applications* 36.32 (2024), pp. 20473–20491.

[13] Wenbai Chen et al. "MTD-YOLO: Multi-task deep convolutional neural network for cherry tomato fruit bunch maturity detection". In: *Computers and Electronics in Agriculture* 216 (2024),
p. 108533.

[14] Zewen Xie et al. "A lightweight deep learning semantic segmentation model for optical-image- based post-harvest fruit ripeness analysis of sugar apples (Annona squamosa)". In: *Agriculture*
14.4 (2024), p. 591.

[15] Bo-Jin Chen et al. "AFBF-YOLO: An Improved YOLO11n Algorithm for Detecting Bunch and Maturity of Cherry Tomatoes in Greenhouse Environments". In: *Plants* 14.16 (2025), p. 2587.

[16] Jun Yuan et al. "Deployment of CES-YOLO: An Optimized YOLO-Based Model for Blueberry Ripeness Detection on Edge Devices". In: *Agronomy* 15.8 (2025), p. 1948.

[17] Nada Theyab Alharbi et al. "Deep Learning Approach for Classifying Date Fruits Using YOLO: A Comparative Analysis". In: *2024 17th International Conference on Development in eSystem Engineering (DeSE)*. IEEE. 2024, pp. 503–508.

[18] Aichen Wang et al. "NVW-YOLOv8s: An improved YOLOv8s network for real-time detection and segmentation of tomato fruits at different ripeness stages". In: *Computers and Electronics in Agriculture* 219 (2024), p. 108833.

[19] Joseph Redmon and Ali Farhadi. "Yolov3: An incremental improvement". In: *arXiv preprint arXiv:1804.02767* (2018).

[20] Joseph Redmon et al. "You only look once: Unified, real-time object detection". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 779–788.

[21] Zihan Ni et al. "LIGHT YOLO for high-speed gesture recognition". In: *2018 25th IEEE Inter- national Conference on Image Processing (ICIP)*. IEEE. 2018, pp. 3099–3103.

[22] Hazem Ashor Amran Abolholl, Tom-Robin Teschner, and Irene Moulitsas. "Surface line inte- gral convolution-based vortex detection using computer vision". In: *Journal of Computing and Information Science in Engineering* 23.5 (2023), p. 051002.

[23] Sonay Duman, Abdullah Elewi, and Zeki Yetgin. "Distance estimation from a monocular camera using face and body features". In: *Arabian Journal for Science and Engineering* 47.2 (2022),
pp. 1547–1557.

[24] Qiwei Wang et al. "Deep learning approach to peripheral leukocyte recognition". In: *PloS one*
14.6 (2019), e0218808.

[25] Zhang Yi, Shen Yongliang, and Zhang Jun. "An improved tiny-yolov3 pedestrian detection al- gorithm". In: *Optik* 183 (2019), pp. 17–23.

[26] Ning Lv, Jian Xiao, and Yujing Qiao. "Object detection algorithm for surface defects based on a novel YOLOv3 model". In: *Processes* 10.4 (2022), p. 701.

[27] Jesse Davis and Mark Goadrich. "The relationship between Precision-Recall and ROC curves". In: *Proceedings of the 23rd international conference on Machine learning*. 2006, pp. 233–240.

[28] Agila, F. E. S. (2026). Insecticidal effectiveness of seed and leaf oil extract of neem (Azadirachta indica) against larvae and adults of Anopheles stephensi. *Al-Farooq Journal of Sciences*, *2*(3), 429-441.

[29] Shahzad Ali Khan and Zeeshan Ali Rana. "Evaluating performance of software defect predic- tion models using area under precision-Recall curve (AUC-PR)". In: *2019 2nd International Conference on Advancements in Computational Sciences (ICACS)*. IEEE. 2019, pp. 1–6.

[30] Simon Edström. *Object detection for autonomous trash and litter collection*. 2022.

[31] Pengjun Xiang et al. "FFTCA: a Feature Fusion Mechanism Based on Fast Fourier Transform for Rapid Classification of Apple Damage and Real-Time Sorting by Robots". In: *Food and Bioprocess Technology* 18.2 (2025), pp. 1631–1655.