**ISSN:**
Print - 2277 - 0593
Online - 2315 - 7461


**Journal of Natural Science, Engineering and Technology**

# AUTOMATED ASSESSMENT OF TOMATO RIPENESS: A YOLOv8 APPROACH FOR PRECISION

***S. OWOEYE, F. DURODOLA, S. ABDULKAREEM, E. ESAN, I. DADA, J. AKINTIBU, A. OGUNSILU**

Department of Mechatronics Engineering, Federal University of Agriculture, Abeokuta, Nigeria
*Corresponding Author:owoeyeso@funaab.edu.ng Tel: +2348034893328

## ABSTRACT

Tomatoes (*Solanum lycopersicum*) are important fruits globally and accurately assessing their ripeness is essential for optimizing harvesting processes and reducing food waste. Traditional farmers mostly depend on manual inspections, which are labour-intensive and prone to human error. This research employed the You Only Look Once version 8 (YOLOv8) algorithm to enhance detection accuracy and efficiency across various environmental conditions. A dataset of 8,579 images, consisting of both ripe and unripe tomatoes were curated, encompassing diverse lighting scenarios and backgrounds. Data augmentation and annotation were used to increase the model's robustness. The developed model attained a mean average precision (mAP) of 0.955 at an Intersection over Union (IoU) threshold of 0.5, demonstrating precision and recall rates of 92% and 94% for both ripe and unripe tomatoes, respectively. YOLOv8 performed well in image processing and classification, minimizing false positives and negatives, at a 0.7 confidence threshold with an F1-score of 0.90. Implementation of YOLOv8 enhanced detection capabilities and also aligned with the principles of precision agriculture, facilitating data-driven decision-making for farmers. This research has contributed to the body of knowledge on automated fruit classification systems, offering method that can be adapted for other agricultural products. By establishing a reliable framework for detecting ripe and unripe tomatoes, the study emphasized the potential of deep learning to revolutionize agricultural methods, thereby fostering sustainability and improving food security in a progressively competitive international marketplace.

**Keywords:** Algorithm; F1-score; Mean average precision; Object detection Tomato maturity; YOLOv8.

## INTRODUCTION

Tomato (*Solanum lycopersicum*) ranks among the most commonly cultivated and consumed fruits globally, playing a significant role in various culinary traditions and agricultural economics. In 2019 alone, farmers produced over 180 million tonnes of tomatoes worldwide, making it one of the most valuable crops in global agriculture (Panno *et al.,* 2021). The ability to accurately detect and classify ripe and unripe tomatoes is crucial for optimizing harvesting processes, ensuring quality control and reducing food waste. Conventional techniques for evaluating the ripeness of tomatoes typically depend on

manual examination, a process that can be laborious, time-consuming, and prone to human mistakes. There arises a growing concern in the application of cutting-edge technologies to automate this process.

Recent approaches in deep learning and computer vision have produced viable opportunities for the detection and classification of agricultural products. Among these developments, You Only Look Once version (YOLO) algorithm has surfaced as a powerful tool for real-time object detection tasks (Ali and Zhang, 2024). YOLOv8 shows better performance and faster results, which is useful in agriculture where farmers need to act quickly (Pativada, 2024; Sharma and Vardhan, 2024). Tests showed that YOLOv8 can identify seeds and plants with over 90% accuracy in real-time, making it practical for actual farm use. When detecting weeds, YOLOv8 achieved 82.5% precision, proving it works well for farm automation tasks (Giri, 2025).

The detection and classification of ripe and unripe tomato fruits using YOLOv8 represents important step in automating farming tasks. YOLOv8 has made significant advancements in real-time object detection. This model builds upon its predecessors by enhancing the backbone architecture and integrating novel components that improve feature extraction and processing capabilities (Zhong *et al.,* 2025). For instance, the C2f (Cross Stage Partial block with 2 convolutional layers and feature fusion) module in YOLOv8 accelerates training and enhances the extraction of key features while the transition to an anchor-free detection head allows for more efficient classification and regression tasks, ultimately boosting overall performance (Zhao *et al.,* 2024; Zhang *et al.,* 2024; Xiong *et al.,* 2025). These improvements make YOLOv8 particularly effective in complex environments, where traditional models may struggle with accuracy and speed (Jiang *et al.,* 2024; Li *et al.,* 2024; Chen *et al.,* 2024; Owoeye *et al.,* 2024).

Recent research by Xiao *et al.* (2023) examined the use of deep learning methods in agricultural settings, particularly focusing on detection and recognition of fruit for automated harvesting. The study highlighted that deep learning methods, especially Convolutional Neural Network (CNN), are widely used in fruit detection as a result of their capabilities to extract advanced features from the images. Their analysis show that CNN offer significant advantages over manual methods in terms of processing speed and detection accuracy. This study sets the stage for further exploration of more advanced architectures, such as YOLOv8, which promises real-time detection capabilities. Also, a study by Lawal *et al.* (2023) evaluated an improved YOLOv5s model in agricultural settings, achieving 96.0% mAP for fruit detection with real-time processing at 74 fps. Their lightweight model proved effective for complex farming conditions, demonstrating the advantages of YOLO architectures in dynamic agricultural environments.

In a comparative analysis, Wang *et al.* (2022) developed an improved Faster R-CNN model (MatDet) for tomato maturity detection, achieving 96.14% average precision (mAP) in complex agricultural scenarios. In a study by Mbouembe *et al.* (2023) proposed an improved YOLOv4-tiny model for tomato detection, which achieved 96.3% precision with significantly faster processing at 1.9 ms per image. Their results demonstrated that while Faster R-CNN variants excel in detection accuracy, YOLOv4 architectures provide superior speed performance for real-time applications. This finding suggests that en-

hancing YOLO architectures could yield even better results, which is one motivation for exploring YOLOv8 in this study. Joshi *et al.* (2023) discussed the importance of using standardized, diverse datasets for training deep learning models in agriculture. They demonstrated that agricultural-specific pre-training and proper data standardization can considerably boost model performance while reducing convergence time and training resources. Their research highlighted that the effectiveness of agricultural deep learning models is significantly influenced by training data quality and standardization approaches, which directly affect data efficiency and generalization capabilities across different agricultural conditions. The research indicated that models developed using low-quality annotations can achieve performance levels similar to those trained on high-quality data, implying that enhancing the variety of accessible datasets through appropriate standardization may bolster the robustness of models. This article aimed at developing and evaluating an effective computer vision-based system for the detection and classification of ripe and unripe tomato fruits using the YOLOv8. By leveraging on the deep learning capabilities and improved accuracy of YOLOv8, the study aimed to automate the identification process in agricultural settings, enhancing productivity, reducing manual labour, and supporting smart farming practices through precise fruit maturity assessment.

## METHODOLOGY

The methodology consisted of several essential phases: data collection, data preprocessing, model training, and evaluation. Each of these phases are of a high significant in guaranteeing the effectiveness and precision of the detection and classification system.

### *Data Collection and Pre-processing*

The initial phase entailed the collection of an extensive dataset comprising images of tomatoes. High-quality images of ripe and unripe tomatoes were collected from various sources, including local farms and online repositories. The total datasets used consisted of 8579 images. The dataset was designed to encompass a range of environmental conditions, lighting scenarios and backgrounds to improve the model robustness. Once images were collected, they passed through preprocessing; each image was annotated using annotation tool (makesense.ai), where bounding boxes were drawn around each tomato and classified as either ripe or unripe. This phase is essential, as precise annotations act as a noise filter and the foundational truth for training the YOLOv8 model. Annotated dataset was subsequently split to three: training (63.4%), validation (17.7%), and test (18.9%), this was to guarantee the model ability to effectively recognize new data. Images and their corresponding annotation files were reorganized to ensure consistency in the dataset structure. They were thereafter moved to appropriate folders and were subsequently resized to the dimensions specified for YOLOv8 (960 by 960 pixels), followed by the normalization of pixel values and the application of data augmentation methods, including rotation, flipping, and color adjustments, to increase variability (Figure 1).

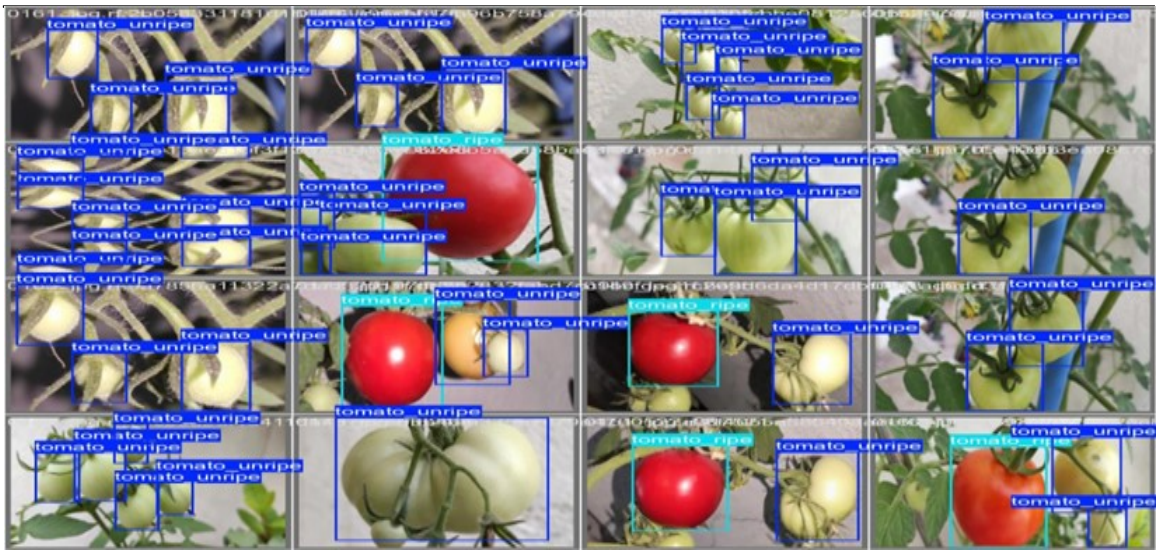


**Figure 1: Dataset Sample with ripe and unripe tomato bounding boxes**

***Model Configuration***

The YOLOv8 architecture (Figure 2) was used for this study due to its balance between speed and accuracy. The model configuration was set up to define parameters such as input size, batch size, and learning rate. Pre-trained weights on a similar dataset were utilized to initialize the model, allowing for transfer learning, which accelerates convergence and improves performance. Training the YOLOv8 model involved feeding the pre-processed training dataset into the network. The training process was monitored through metrics like loss and mean average precision (mAP). Early stopping was implemented to guide against the error of overfitting and the model was trained over several epochs until convergence was achieved. The model initialized with pre-trained weights, transferring 319 out of 355 parameters. Automatic Mixed Precision (AMP) was enabled to optimize training speed and reduce memory usage.

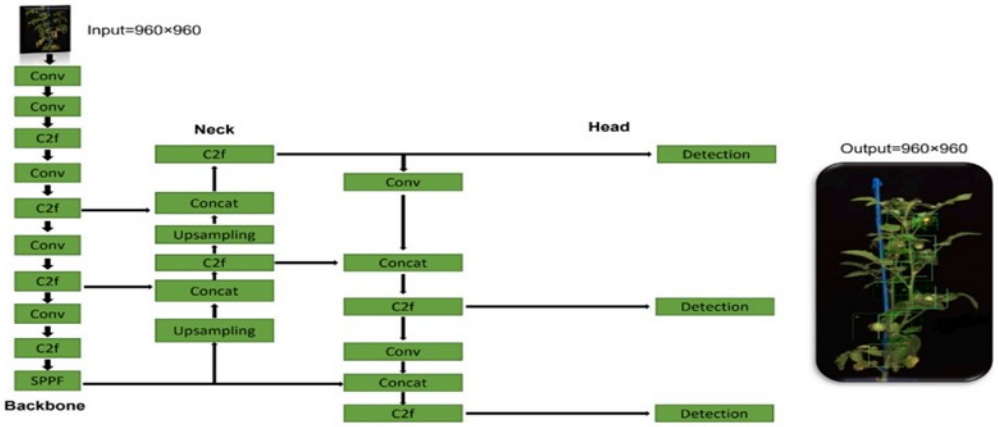


**Figure 2: YOLOv8 Architecture (Solimani *et al.,* 2024)**

***Model Training***

The YOLOv8n model was used for training with 3,011,238 parameters, where 319 out of 355 pre-trained weights were successfully transferred. The training was performed using Google Colab with Tesla T4 GPU (15102MiB), spanning 55 epochs with early stopping patience set to 10. Training environment utilized Ultralytics 8.3.63, Python 3.11.11, PyTorch 2.5.1+cu121 with CUDA support. The model was configured with batch size 16, image size 960 x 960, AdamW optimizer with automatically determined learning rate of 0.001667 and momentum of 0.9, and 8 workers for data loading. Automatic Mixed Precision (AMP) was enabled for optimized training performance (Figure 3).

```
yaml_file = "/content/drive/MyDrive/Tomato_Project/Datasets/tomato_grading-10/data.yaml"
Training_results = "/content/drive/MyDrive/Tomato_Project/Results"

!yolo task=detect mode=train model=yolov8n.pt data=$yaml_file epochs = 55 imgsz=640 project=$Training_results name=tomato_training_nano
```

**Figure 3: The executing code line**

**Model Evaluation**

Once the training phase was completed, the model was tested with the validation dataset to evaluate its performance. Important metrics, including F1-score, recall, precision, and mean Average Precision (mAP), were calculated to determine the effectiveness of the model in classifying and detecting ripe and unripe tomatoes. Binary cross entropy classifier was used as the objective function. Confusion matrix was generated to visualize the classification results, offering an understanding of the various errors made by the model. To optimize the performance of the YOLOv8 model, hyper parameter tuning was conducted. The aim was to determine the optimal set of hyper parameters that maximize the model's accuracy and minimize overfitting.

**Performance Metrics**

The study employed several performance metrics to evaluate its effectiveness in object detection tasks. They provided intuitions into the model's accuracy, precision and overall performance in identifying and localizing objects within images.

Intersection over Union (IoU)

The Intersection over Union (IoU) is an essential parameter employed to determine the extent of overlay between the predicted bounding box and the corresponding ground truth bounding box. An increased IoU signifies enhanced localization precision. The IoU can be obtained using equation (1) (Cheng *et al.,* 2021; Ali *et al.,* 2024):

$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union} \quad (1)$$

Precision and Recall

Precision measures the correctness of the positive predictions generated by the model, it is obtained using (2) (Solimani *et al.,* 2024; Cheng *et al.,* 2021):

$$\boldsymbol{Precision = \frac{True\ Positives}{True\ Positives + False\ Positives}} \qquad (2)$$

Recall assesses the capability of the model to recognize all pertinent instances, it is obtained using (3) (Park *et al.,* 2021):

$$\boldsymbol{Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}} \qquad (3)$$

Average Precision (AP)

AP is computed as the area under the precision-recall curve for individual class. It summarizes the precision and recall at different thresholds, providing a single score for each class. It can be obtained using (4) (Park *et al.,* 2021):

$$\boldsymbol{A_P = \frac{1}{N}\sum_{k=1}^{n}\left(P\,(k)\,.rel\,(k)\right)} \qquad (4)$$

where:

n = total number of retrieved items

N = total number of relevant items in the dataset

P(k) = precision at rank k

rel(k) = relevance indicator function:

$$\boldsymbol{rel\,(k) = \begin{cases} 1, & if\ item\ at\ rank\ k\ is\ relevant \\ 0, & otherwise \end{cases}}$$

Mean Average Precision (mAP)

mAP extends the concept of AP by averaging the AP values across all classes. It is often calculated at specific IoU thresholds, such as 0.5 (mAP@0.5) or averaged across multiple thresholds (mAP@0.5:0.95), it is calculated using (5) (Park *et al.,* 2021; Liu *et al.,* 2021):

$$\boldsymbol{mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i} \qquad (5)$$

N is the number of classes, and $\boldsymbol{AP_i}$ is the average precision for class i.

F1 Score

The F1 Score represents the harmonic mean of recall and precision, offering a balanced assessment of the model's performance, equation (6) is used to calculate the F1 score (Cheng *et al.,* 2021; Ali *et al.,* 2024):

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

## RESULTS

***Analysis of the dataset:***

The statistics of both ripe and unripe tomatoes indicate nearly equal counts, with approximately 9,000 instances each (Figure 4a). This distribution suggests no class imbalance, which is advantageous for machine learning classification tasks. The similar count for each category contributed to unbiased predictions. The visualization of the dataset depicted overlapping rectangles, centred within the plot. The rectangles varied in size, forming a gradient-like density toward the centre (Figure 4b), representing bounding box coordinates for objects in the image dataset. The concentration of rectangles in the middle suggests that objects are predominantly located near the centre of the frame. This pattern is common in datasets where subjects are intentionally framed for analysis. The variability in rectangle size reflects diverse object dimensions, though it can be seen that smaller objects dominate the dataset.

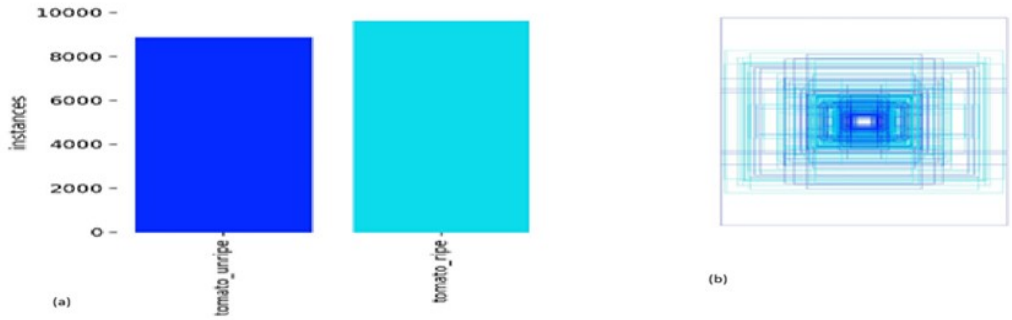


**Figure 4: Analyses of the Dataset (a) Bar Chart (b) Rectangular Overlay Plot**

***Training Metrics:***

The key training and validation metrics for the model gave insights into the model's learning process, data performance, and its readiness for deployment (Figure 5). The train/box loss graph quantifies how accurately the model predicts the bounding boxes of objects. The graph shows a decline in this loss from approximately 1.1 at the beginning to 0.7 by the end of 55 epochs. This consistent decrease is indicative of the model's improved ability to localize objects accurately in the images as training progresses. A smooth and continuous downward trend like this demonstrates an effective learning of spatial relationships and refining predictions

for object localization by the model. The observed drop in box loss shows the model's effectiveness in minimizing errors in bounding box placement. This is a crucial aspect of object detection models, as accurate bounding box predictions ensure precise object localization, a key requirement for downstream applications.

The train/cls loss graph tracks the classification loss during training; starting at a value of around 1.6, the loss decreased to approximately 0.4 by the final epoch. This rapid decline in the early epochs, followed by a smooth plateau, indicates that the model quickly learns to distinguish between classes: "Ripe Tomato" and "Unripe Tomato." The drop in classification loss indicates an enhancement in the model's capacity to differentiate between object categories. As the training progressed, the model gained confidence and accuracy in its predictions, which is critical for reliable object detection in real-world scenarios. The distribution focal loss (DFL), train/dfl loss is a measure of how well the model fine-tunes its bounding box regression predictions. It started at approximately 1.3 and decreased to a little above 1.0 by the end of training. The decline in this loss shows that the model continuously refines its bounding box predictions, ensuring precision and reducing localization errors. The improvement in DFL suggests that the model is optimizing its bounding box predictions incrementally, leading to better alignment of the predicted and ground-truth object locations.

The graph of metrics/precision (B) evaluated the number of the objects detected by the model that are relevant (Figure 5). Initially, the precision started at around 0.65 and increased, reaching above 0.90 towards the final epochs. This trend demonstrates that the model reduces the number of false positives over time, leading to more accurate detections. The high precision indicates that the model reliably detects objects with minimal false alarms. The metrics/recall (B) graph assessed the model's capacity to identify all relevant objects in the dataset. Starting at around 0.65, recall increased to approximately 0.90 by the end of training. This upward trend reflects the model's growing capability to detect a larger proportion of relevant objects, thereby reducing the likelihood of missing objects in the images. The improvement in recall implies that the model minimizes false negatives, ensuring that most objects of interest are detected. This is especially valuable in scenarios where missing objects could have critical consequences, such as agricultural yield estimation or quality control.

**Validation Metrics**

The val/box loss graph tracks the model's localization performance on unseen data. Similar to the training box loss, this metric decreased from 1.2 to below 0.7 (Figure 5). The alignment of trends between training and validation box losses suggests that the model generalizes well to new data. The consistent reduction in validation box loss confirms that the model's ability to predict bounding boxes extends beyond the training set. This indicates that the model is robust and unlikely to overfit. Also, the graph of val/cls loss mirrored the trend seen in the training classification loss, decreasing from 1.4 to around 0.4. This alignment further confirms the model's capability to generalize its learned classification skills to unseen data. A declining validation classification loss reinforces that the model can accurately classify objects in new images, making it suitable for real-world deployment. The validation distribution focal loss (val/dfl loss) decreased

from 1.4 to slightly above 1.0, aligning closely with the training DFL trend. This is an indication that the model consistently refines its predictions for bounding box regression across both training and validation datasets. The similarity between training and validation DFL trends signifies that the model does not overfit during fine-tuning of bounding box predictions. The mAP at IoU threshold 0.5, shown in metrics/mAP50(B), measured the overall accuracy of object detection in terms of both localization and classification. Starting at around 0.65, mAP50 increased to nearly 0.95 by the end of training. This notable enhancement underscores the model's capability to effectively balance precision and recall. A high mAP50 score signifies that the model excels in identifying objects with a satisfactory overlap between the predicted and actual bounding boxes. The metrics/mAP50-95 (B), which is the Mean Average Precision at IoU thresholds of 0.5 to 0.95 in 0.05 increments is a stricter measure of detection performance. Starting at 0.5, mAP50-95 increased to around 0.75. This demonstrates the model's robustness across varying levels of intersection-over-union. This is needed to detect objects of varying sizes and overlap ratios. The gradual increase in mAP50-95 reflects the model's versatility and adaptability in detecting objects under various conditions. This robustness is critical for deploying the model in diverse real-world environments. The alignment of training and validation metrics, along with their decline, suggests that the model is neither overfitting nor underfitting. The model achieved strong precision and recall scores, ensuring reliable and comprehensive object detection. High mAP scores confirm that the model balances localization and classification effectively, making it suitable for deployment and also making it a reliable tool for object detection tasks.

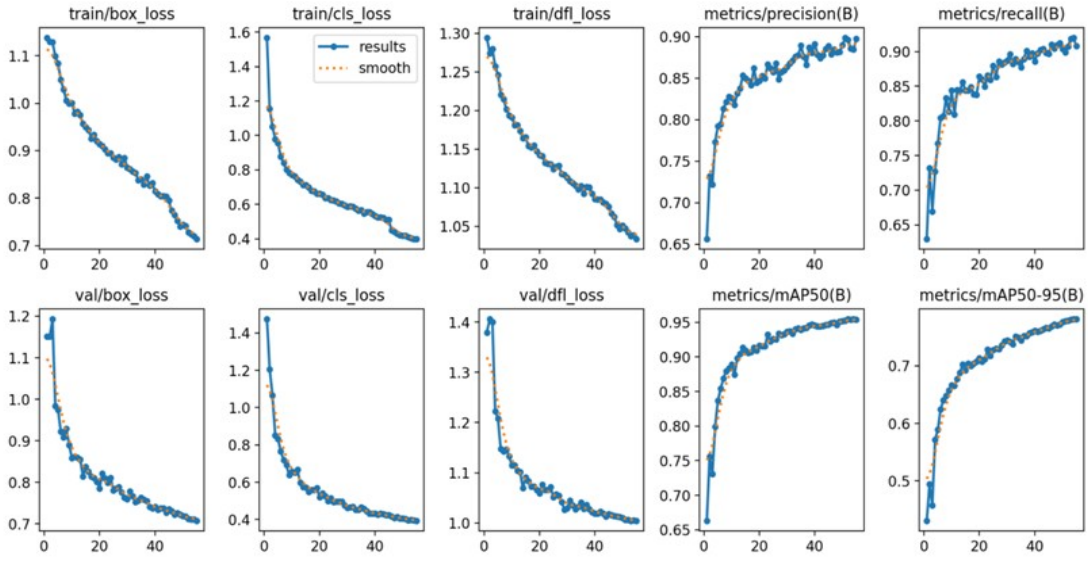


**Figure 5: Training and validation model graphs**

**Metrics Summary:**

The precision of the model for the two classes (ripe and unripe tomato) as a function of confidence thresholds was sketched (Figure 6). The dark blue curve represents the averaged precision across all classes. This shows a great degree of precision for both classes, with precision values increasing as the confidence threshold rises. At lower confidence thresholds (e.g. 0.2 to 0.5), the precision is moderate, ranging from approximately 0.4 to 0.8. This indicates that the model makes more false-positive predictions when it is less confident about its decisions. As the confidence threshold increased, particularly beyond 0.6, the precision for both classes improved significantly, nearing 1.0 at high confidence levels (e.g., 0.8 to 1.0).The average precision across all classes, remains slightly higher than the individual class curves. This suggests that the model performs consistently across both classes, with minimal variability. At the highlighted confidence threshold of 0.999, the precision reached a perfect score (1.0), indicating that at this level, all the model's predictions are correct, with no false positives.

The model demonstrated near-perfect precision at confidence thresholds above 0.8. This means it is highly reliable in distinguishing between unripe and ripe tomatoes when it is confident about its predictions. At lower confidence thresholds, the model sacrifices precision, making more incorrect predictions. This trade-off highlights the importance of selecting an appropriate confidence threshold depending on the application. The high average precision across all classes (dark blue curve) suggested that the model performs robustly, even when dealing with multiple classes. This indicates that the model has learned the distinctive features between the ripe and unripe tomatoes effectively, minimizing confusion between the classes.

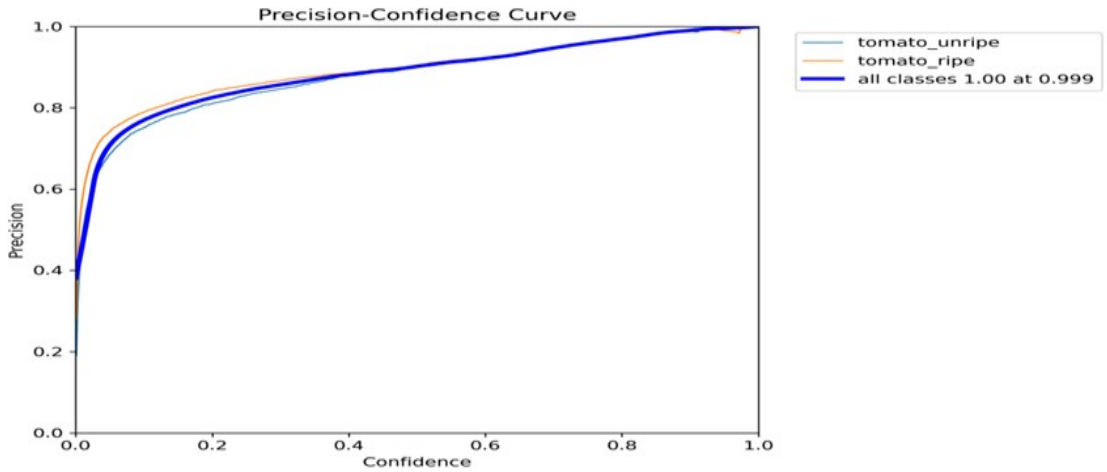


**Figure 6: Precision- Confidence Curve**

Recall started at nearly 1.0 when the confidence threshold is 0 (Figure 7), meaning that the model identified almost all the true positive cases when it is highly lenient, classifying nearly every instance as positive. As the confidence threshold increased, recall gradually decreased, dropping sharply near a confidence level of 0.8 (Figure 7). This drop occurred because the model became stricter, classifying fewer instances as positive, leading to missed true positives. For the two classes, the recall curves are similar, indicating consistent performance across both classes. However, the "ripe tomato" class maintained slightly higher recall at higher thresholds, suggesting that the model is slightly better at identifying ripe tomatoes at more restrictive confidence levels.

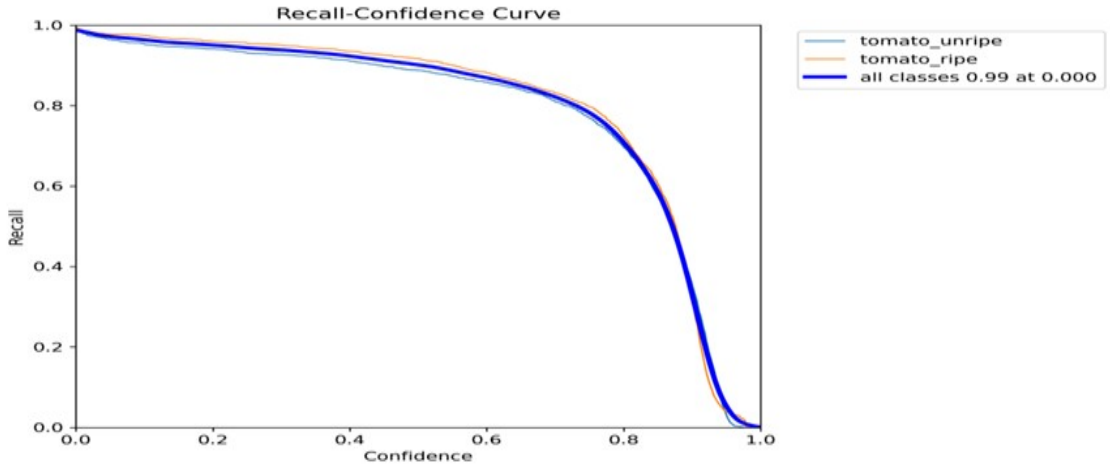


**Figure 7: Recall- Confidence Curve**

The Precision-Recall Curve showed the balance between precision and recall as the confidence threshold changes. The model for ripe tomatoes exhibits a greater precision in comparison to that of unripe tomatoes, indicating its superior reliability in accurately classifying ripe tomatoes without erroneously categorizing unripe ones. The overall mean Average Precision (mAP) across all classes stood at 0.955 with an Intersection over Union (IoU) threshold of 0.5 (Figure 8), underscoring the model's remarkable capability to maintain a balance between precision and recall. The curve consistently showed high precision for both classes over a broad spectrum of recall values. As recall approached 1.0, precision tends to decline, demonstrating that while the model identifies more true positives, it simultaneously increased the number of false positives. This trade-off is an inherent result of modifying the classification threshold. The F1-score curve for both ripe and unripe followed a bell shape, peaking at an optimal threshold of approximately 0.4 (Figure 9). At this threshold, the score for all classes is 0.90, reflecting an excellent balance between precision and recall. Beyond this threshold, the F1-score began to decline, indicating that the model either sacrifices recall (at higher thresholds) or precision (at lower thresholds). The “ripe tomato” class achieved a slightly higher F1-score compared to "unripe tomato" confirming its better overall perfor-

mance in classification. The peak F1-score of 0.90 at a threshold of 0.4 suggests that this confidence level is optimal for deploying the model in real-world scenarios. At this threshold, the model achieves a strong balance between precision and recall, ensuring high accuracy in sorting ripe and unripe tomatoes. This confidence level minimizes errors while maintaining efficiency in operations.

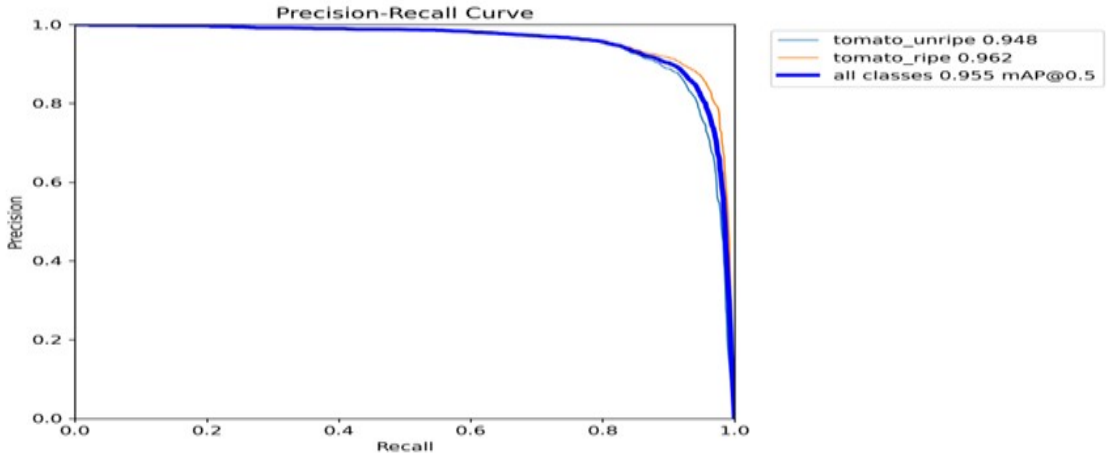


**Figure 8: Precision – Recall Curve**

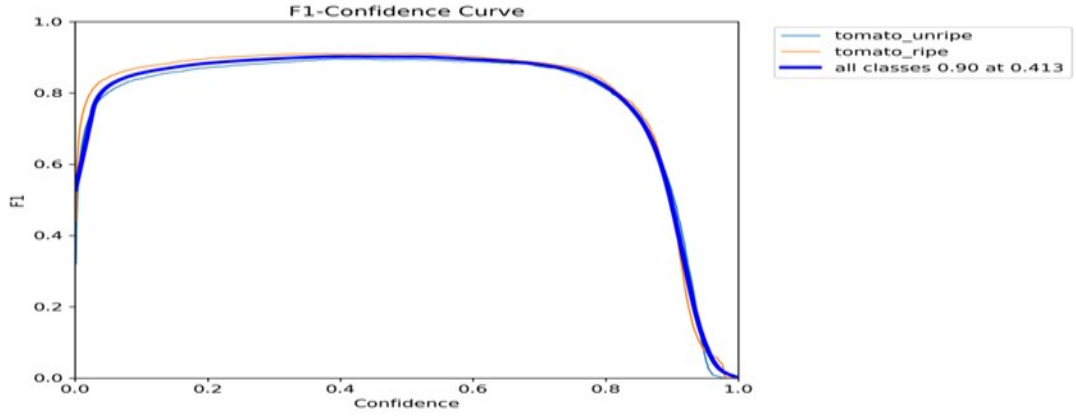


Figure 9: F1 – Confidence Curve

The normalized confusion matrix representing the performance of a classification model on three classes: unripe tomato, ripe tomato, and background. The matrix's values are normalized, meaning each cell shows the proportion of predictions relative to the actual instances of each class, instead of absolute counts. It can be deduced that 92% of unripe tomatoes samples were correctly classified, 94% of ripe tomato samples are correctly classified while only 48% of background samples are correctly classified

(Figure 10). The unripe tomatoes misclassified as ripe tomatoes were 2% while the misclassified as background were 6%. Also, the ripe tomatoes misclassified as unripe tomatoes were 2% and those misclassified as background were 4% (Figure 10). The model performed very well on ripe and unripe tomatoes with high classification accuracy, indicating that the features used for classification effectively differentiated between these two classes, showing its ability to handle subtle differences in features (e.g., colour, texture). The background class suffered from a high rate of misclassification, 52% of background samples are misclassified as unripe tomatoes, with 8% of background samples are misclassified as ripe tomatoes. This suggests that the model struggled to distinguish non-tomato regions from tomato-related regions. The central clustering of objects in the dataset (from the earlier analysis) may cause the model to associate many non-tomato regions (background) with tomatoes due to overlapping spatial and visual features.

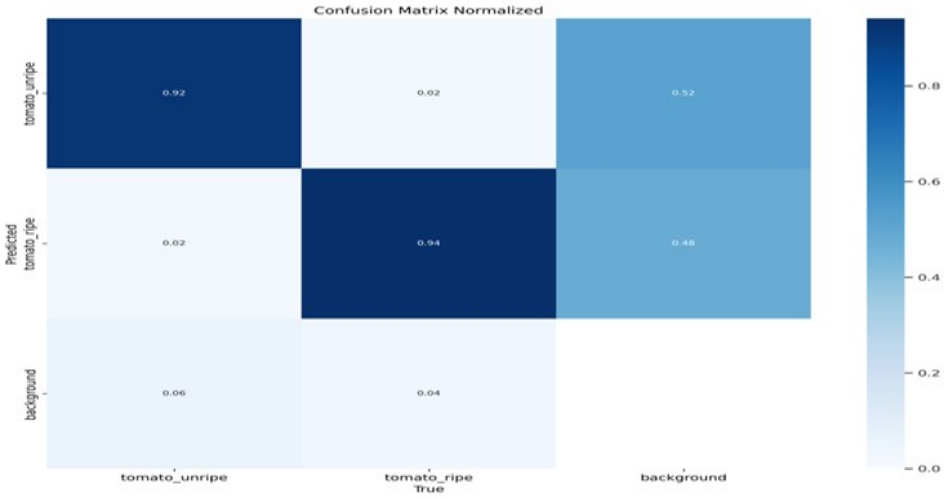


**Figure 10: Normalized Confusion Matrix**

## DISCUSSION

The present study developed and evaluated a YOLOv8-based object detection model for automated assessment of tomato ripeness, achieving strong performance metrics that compare favourably with contemporary deep learning approaches in agricultural computer vision. The achieved mAP50 represents a detection accuracy that is competitive with, and in some aspects superior to, recent YOLOv8 implementations for fruit ripeness assessment. Liang *et al.* (2025) reported that their improved YOLOv8n model for cherry tomato ripening detection achieved accuracy improvements of 3.18% over baseline following the integration of ADown down-sampling modules, Slim-Neck architectures, and efficient multi-scale attention (EMA) mechanisms. While their architectural modifications yielded incremental gains, the baseline performance they improved upon was substantially lower than the metrics reported in the present study, suggesting that the standard YOLOv8 architec-

ture is remarkably well-suited to binary ripeness classification tasks without extensive customization. Though Wang *et al.* (2023) using R-CNN for tomato maturity classification was able to achieve a mAP of 96.14%, this slight increase might be due to different stages of maturity used and the composition of the dataset.

In the domain of citrus fruit detection, Li *et al.* (2025) employed an improved YOLOv8n -seg model incorporating GhostConv replacements, convolutional block attention module (CBAM) mechanisms, and dedicated small-object detection layers. Their model achieved an overall mAP50 of 94.43%. The present study's mAP50 of 95% is directly comparable to this performance, despite fundamental differences in fruit morphology (citrus versus tomato) and the additional complexity of stem localization in their work. This parity indicates that YOLOv8 generalizes effectively across diverse fruit types when trained on well-annotated, balanced datasets. A comparative investigation by Putra *et al.* (2025) examined YOLOv8 and detection transformer (DETR) architectures for multi-level tomato ripeness detection across four maturity stages. Their findings demonstrated that YOLO architectures provide distinct advantages for real-time responsiveness and deployment on resource-limited hardware, making them particularly suitable for mobile automation and field-based harvesting systems. These observations corroborate the practical deployment potential of the present model, particularly given the strong precision-recall balance evidenced by an F1-score peak of 0.90 at a confidence threshold of 0.4.

The mAP50-95 value achieved in this study is particularly noteworthy, as this stricter metric evaluates detection performance across ten IoU thresholds ranging from 0.50 to 0.95 in 0.05 increments, thereby penalizing imprecise localization. Research on tomato instance segmentation by Wei *et al.* (2025) using an improved YOLOv8s-seg model (ACP-Tomato-Seg) reported mAP50-95 improvements of 8.3% over baseline following the introduction of adaptive and oriented feature refinement modules and custom multi-scale pooling modules. The baseline mAP50-95 for standard YOLOv8s-seg, inferred from the reported 8.3% improvement, was lower than achieved in the present study. This comparison underscores the exceptional localization precision of the current model, which attained this strict metric without specialized architectural modifications for multi-scale feature extraction.

The importance of mAP50-95 as a performance metric in agricultural applications cannot be overstated. For robotic harvesting systems, bounding box precision directly influences gripper positioning accuracy, and even modest localization errors can result in grasp failure or fruit damage. The high mAP50-95 value reported suggests that the model consistently produces tight, well-aligned bounding boxes across varying object sizes and overlap conditions, a critical requirement for downstream automation tasks. This precision-recall trade-off is characteristic of well-optimized object detectors and reflects the inherent tension between minimizing false positives and minimizing false negatives. The F1-score peak at a confidence threshold of 0.4 represents the optimal operating point that balances these competing objectives. For practical deployment, the selection of an appropriate confidence threshold must be guided by the specific operational context. In automated harvesting systems, high precision is typically prioritized to avoid

the economic waste of attempting to harvest unripe fruits, suggesting that a higher confidence threshold (e.g., >0.8) might be appropriate despite the associated reduction in recall. Conversely, for yield estimation applications where comprehensive counting is paramount, high recall may be prioritized, favouring a lower threshold near the F1-optimal value of 0.4. The model's strong performance across the entire confidence spectrum provides end-users with the flexibility to tune this parameter according to their specific accuracy requirements.

## CONCLUSION

This study on detecting and classifying ripe and unripe tomato fruits using YOLOv8 demonstrates the significant benefits that deep learning can bring to agricultural practices. By using YOLOv8's capabilities, this research successfully automates the traditionally labor-intensive process of assessing tomato ripeness. The developed model provides a robust framework that can be integrated with automated harvesting hardware to identify and harvest ripe tomatoes, significantly improving efficiency and accuracy in farming operations.

The results show that YOLOv8 achieves excellent detection performance for both ripe and unripe tomatoes, while maintaining high precision. The model effectively minimizes human error and addresses critical challenges found in traditional agricultural methods. The balanced precision-recall performance demonstrates the model's reliability for real-world deployment in tomato farms.

## LIMITATIONS

The most significant limitation identified in this study is the confusion between background regions and tomato classes, with only 48% of background samples correctly classified. The normalized confusion matrix revealed that 52% of background samples were misclassified as unripe tomatoes, while 8% were misclassified as ripe tomatoes. This pattern of background confusion warrants careful consideration, as it directly impacts the model's false positive rate in field deployments.

Future research should focus on expanding the training dataset to include more diverse tomato varieties and environmental conditions, which will enhance the model's robustness and accuracy across different scenarios. Also, implementing the developed YOLOv8 model on robotic harvesting systems would further advance precision agriculture practices.

## REFERENCES

**Ali, U., Ismail, M. A., Habeeb, R. A. A., S. R. A. Shah** 2024. "Performance evaluation of YOLO models in plant disease detection", *Journal of Informatics and Web Engineering 3*(2): 199-211.

**Ali, M. L., Z. Zhang** 2024. "The YOLO framework: A comprehensive review of evolution, applications, and benchmarks in object detection". *Computers 13*(12): 336.

**Chen, Z., Feng, J., Yang, Z., Wang, Y., M. Ren** 2024. "YOLOv8-ACCW: Lightweight grape leaf disease detection method based on improved YOLOv8", *IEEE Access* 12: 123595-123608.

**Cheng, B., Girshick, R., Dollár, P., Berg, A. C., A. Kirillov** 2021. "Boundary IoU: Improving object-centric image segmentation evaluation", In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 15334-15342.

**Giri, K. J.** 2025. "SO-YOLOv8: A novel deep learning-based approach for small object detection with YOLO beyond COCO". *Expert Systems with Applications* 280: 127447.

**Jiang, H., Li, J., & H. Tian** 2024. "Algorithm for Student Behavior Detection Based on YOLOv8", In 2024 3rd International Conference on Cloud Computing, Big Data Application and Software Engineering (CBASE), 45-48. IEEE.

**Joshi, A., Kumar, V., Patel, D., & R. Singh** 2023. "Standardizing and centralizing datasets for efficient training of agricultural deep learning models", *Plant Phenomics* 5: 0084.

**Lawal, O. M., Zhu, S., K. Cheng** 2023. "An improved YOLOv5s model using feature concatenation with attention mechanism for real-time fruit detection and counting", *Frontiers in Plant Science* 14: 1153505. **https://doi.org/10.3389/fpls.2023.1153505**.

**Li, H., Wu, A., Jiang, Z., Liu, F., M. Luo** 2024. "Improving object detection in YOLOv8n with the C2f-f module and multi -scale fusion reconstruction", *In 2024 IEEE 6th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*. 374-379. IEEE.

**Li, H., Yin, Z., Zuo, Z., Pan, L., J. Zhang** 2025. Precision citrus segmentation and stem picking point localization using improved YOLOv8n-seg algorithm. *Frontiers in Plant Science 16*: 1655093.

**Liang, X., Jia, H., Wang, H., Zhang, L., Li, D., Wei, Z., M. Yang** 2025. ASE-YOLOv8n: A method for cherry tomato ripening detection. *Agronomy* 15(5): 1088.

**Liu, Y., Pu, H., & D.W. Sun** (2021). "Efficient extraction of deep image features using convolutional neural network (CNN) for applications in detecting and analysing complex food matrices". *Trends in Food Science & Technology 113*: 193-204.

**Mbouembe, P. L. T., Liu, G., Sikati, J., Kim, S. C., J. H. Kim** 2023. An efficient tomato-detection method based on improved YOLOv4-tiny model in complex environment", *Frontiers in Plant Science* 14: 1150958. https://doi.org/10.3389/fpls.2023.1150958.

**Owoeye, S., Durodola, F., Adetoro, T., Adegbenro, Kehinde., B. Bisiriyu** 2024. "Identification of Maize and Weed Using Machine Learning Model Implemented Via YOLOv5", *FULafia Journal of Science and Technology* 8(2): 49- 55.

**Panno, S., Davino, S., Caruso, A. G., Bertacca, S., Crnogorac, A., Mandić, A., & S. Matić** 2021. "A review of the most common and economically important diseases that undermine the cultivation of tomato crop in the Mediterranean basin", *Agronomy 11*(11): 2188.

Park, S. S., Tran, V. T., D. E. Lee 2021. "Application of various yolo models for computer vision-based real-time pothole detection". *Applied Sciences* 11(23): 11229.

**Pativada, P. K.** 2024. "Real-time detection and classification of plant seeds using YOLOv8 object detection model", Doctoral dissertation, Kansas State University.

**Putra, M. R. H., Setiawan, D., Purnama, R. A., Apriana, V., Santoso, R.** 2025. Implementation of YOLOv8 and DETR for Multi-Level Tomato Ripeness Detection

with Real-Time Bounding Boxes. *Jurnal Teknologi dan Open Source 8*(2): 1045-1053.

**Sharma, S., M. Vardhan** 2024. "Advancing precision agriculture: Enhanced weed detection using the optimized YOLOv8t model", *Arabian Journal for Science and Engineering* 1-18.

**Solimani, F., Cardellicchio, A., Dimauro, G., Petrozza, A., Summerer, S., Cellini, F., & V. Renò** (2024). "Optimizing tomato plant phenotyping detection: Boosting YOLOv8 architecture to tackle data complexity", *Computers and Electronics in Agriculture* 218: 108728.

**Wang, Z., Ling, Y., Wang, X., Meng, D., Nie, L., An. G., & X. Wang** (2022). "An improved Faster R-CNN model for multi-object tomato maturity detection in complex scenarios", *Ecological Informatics* 72; 101886.

**Wei, J., Sun, Y., Luo, L., Ni, L., Chen, M., You, M., & Gong, H.** (2025). Tomato ripeness detection and fruit segmentation based on instance segmentation. *Frontiers in Plant Science 16*: 1503256.

**Xiao, F., Liu, J., Wang, H., Y. Zhang** (2023). "Fruit Detection and Recognition Based on Deep Learning for Automatic Harvesting: An Overview and Review", *Agronomy* 13(6): 1625–1625. https://doi.org/10.3390/agronomy13061625.

**Xiong, L., He, Y., Lai, H., S. Wang** 2025. "Intelligent Classroom Management System Based on YOLOv8", *In International Conference on Computer Engineering and Networks*. 402-413. Springer, Singapore.

**Zhang, S., Li, P., Yu, W., G. Zhang** 2024. "Pedestrian Detection Algorithm of YOLOV8 Based on Feature Enhancement", In: *Proceedings of the 2024 3rd Asia Conference on Algorithms, Computing and Machine Learning*. 90-95.

**Zhao, Y., Sun, F., X. Wu** 2024. "FEB-YOLOv8: A multi-scale lightweight detection model for underwater object detection", *PLOS ONE*, 19(9), e0311173, 2024.

**Zhong, J., Qian, H., Wang, H., Wang, W., & Y. Zhou** (2025). "Improved real-time object detection method based on YOLOv8: a refined approach". *Journal of Real-Time Image Processing* 22(1): 1-13.

*(Manuscript received: 17th December, 2025; accepted: 20th April, 2026).*