Journal of Artificial Intelligence and Engineering Applications

Website: https://ioinformatic.org/

June 2026. Vol. 5. No. 3; e-ISSN: 2808-4519

Performance of Yolov8 Algorithm and Real-Time Detection Transformer in Tomato Ripeness Detection System

Intan Safira[1]*, Wahyu Fuadi[2], Lidya Rosnita[3]

[1,2,3]*Teknik Informatika, Teknik, Universitas Malikussaleh*
Intan.210170198@mhs.unimal.ac.id[1*], wahyu.fuadi@unimal.ac.id[2], lidyarosnita@unimal.ac.id[3]

Abstract

Tomato ripeness sorting is still widely carried out manually and subjectively, which can lead to inconsistencies in the quality of the sorting results. In addition, the manual process requires more time and has the potential to cause errors in classifying tomato ripeness levels. Therefore, an automatic detection system based on digital images is needed to provide more accurate and consistent detection results. This study aims to analyze and compare the performance of the You Only Look Once (YOLOv8) and Real-Time Detection Transformer (RT-DETR) algorithms in detecting and classifying tomato ripeness levels based on digital images. The research method used is an experimental method consisting of dataset collection, data labeling, image augmentation, data splitting into training, validation, and testing sets, as well as model training using Google Colab. The tomato ripeness levels were classified into six classes to provide a more detailed representation compared to previous studies. Model performance evaluation was carried out using accuracy, precision, and recall metrics. The results showed that YOLOv8 achieved a precision value of 64.8%, recall of 70%, and accuracy of 50.7%. Meanwhile, RT-DETR demonstrated better performance with a precision of 80.8%, recall of 84%, and accuracy of 70%. Based on these results, RT-DETR is considered superior in providing more accurate and consistent predictions, making it more potential to be implemented in a digital image-based tomato sorting system to improve the efficiency and quality of agricultural products.

***Keywords**: YOLOv8, RT-DETR, object detection, tomato ripeness, computer vision*

1. Introduction

The agricultural sector in Indonesia continues to experience rapid development along with the increasing demand for automation to improve production efficiency and product quality. The implementation of technologies such as computer vision and Artificial Intelligence (AI) has been widely applied to optimize various agricultural processes, including harvest sorting and quality assessment [1]. These innovations reduce dependence on manual labor, which is often slow and inconsistent. Through automation, agricultural industries can produce higher-quality products that are more competitive in the global market [2].

Tomatoes are agricultural commodities with high economic value, and their nutritional content, particularly vitamin C, is highly dependent on the level of ripeness [3]. Harvesting tomatoes either too early or too late may result in reduced nutritional quality. In Indonesia, tomato sorting still faces significant challenges because it is generally performed manually and relies on subjective visual observations [4]. Such conditions are vulnerable to worker fatigue and decreased concentration, leading to inconsistent sorting results and an increased risk of spoilage during storage and distribution. Furthermore, tomatoes are climacteric fruits that continue to ripen after harvest. Therefore, errors in determining the maturity level at harvest can significantly affect fruit quality, seed quality, and productivity in subsequent planting seasons [5].

These challenges highlight the urgent need for a sorting system that is not only fast but also capable of providing objective and consistent results. Without technological support, sorting processes may continue to produce inconsistent product quality, negatively affecting market value and competitiveness [6]. In large-scale distribution systems, inaccurate ripeness classification can also lead to economic losses due to increased spoilage rates and quality degradation during storage [7]. Therefore, the adoption of artificial intelligence-based systems represents a strategic and innovative solution to address real-world challenges in modern agriculture [8].

You Only Look Once (YOLO) is one of the most widely used object detection algorithms due to its ability to identify objects quickly and accurately in digital images. Previous studies have demonstrated that YOLO outperforms conventional methods in terms of both accuracy and training speed [9]. Its primary advantage lies in its single-stage detection approach, which processes the entire image globally in a single pass, enabling efficient and highly accurate object detection. Moreover, YOLO exhibits strong generalization capabilities, allowing it to adapt effectively to data variations and changing environmental conditions [10]. Since its introduction in 2015, YOLO has undergone continuous improvements, resulting in more advanced versions that are extensively utilized in various computer vision applications [11].

A study conducted by [12] demonstrated that YOLOv8 can effectively detect and classify tomato ripeness into four maturity categories. These findings indicate the significant potential of deep learning technology to enhance the efficiency and accuracy of sorting processes in the agricultural sector. However, the study was limited by its reliance on a single algorithm and a relatively general classification scheme, without exploring more detailed ripeness levels or comparing its performance with other object detection algorithms. This limitation provides an important opportunity for further research aimed at developing a more optimal and competitive detection system.

Based on previous studies, deep learning-based object detection algorithms have shown considerable potential for identifying tomato ripeness levels. However, most existing studies employ relatively simple classification schemes, generally consisting of only four ripeness categories, which are insufficient to represent the complete ripening process required for more accurate sorting and distribution activities. In addition, studies specifically comparing the performance of YOLOv8 and Real-Time Detection Transformer (RT-DETR) using a more detailed ripeness classification system and comprehensive evaluation metrics, including accuracy, precision, recall, and inference time, remain limited. Therefore, this study aims to compare the performance of YOLOv8 and RT-DETR in detecting six levels of tomato ripeness to determine the most effective model for supporting automated tomato sorting systems.

2. Research Methodology

This study employs a deep learning-based approach for data processing in object detection and classification tasks. The algorithms used in this research are You Only Look Once (YOLO) and Real-Time Detection Transformer (RT-DETR) as object detection models. These algorithms were selected due to their capability to perform real-time object detection with high accuracy, making them suitable for achieving optimal detection and classification performance.

2.1. Real-Time Detection Transformer (RT-DETR)

Real-Time Detection Transformer (RT-DETR) is an object detection model designed to support real-time detection by leveraging the fundamental concepts of Detection Transformer (DETR) and Transformer architecture. The model adopts an end-to-end approach, enabling simultaneous prediction of object classes and bounding box locations within a single integrated framework. This approach has proven effective in improving both efficiency and accuracy in real-time object detection systems [13].

The architecture of RT-DETR consists of three main components, namely the backbone, encoder, and decoder, each of which plays a crucial role in the object detection process. The backbone is responsible for extracting important visual features from input images using a Convolutional Neural Network (CNN) [14]. The extracted features are then processed by a Transformer-based encoder to capture global relationships among features through a self-attention mechanism, which is essential for understanding object contexts comprehensively. Finally, the decoder utilizes the encoder outputs together with object queries to generate more precise predictions of object locations and categories through a combination of cross-attention and self-attention mechanisms. The integration of these three components enables RT-DETR to achieve an optimal balance between inference speed and detection accuracy, making it highly suitable for real-time applications [15].

The self-attention mechanism is a critical component of the model, allowing the system to evaluate relationships among feature elements more effectively. Mathematically, the self-attention operation can be expressed as follows:

$$Attention(Q,K,V) = softmax\left(\frac{QK^{T}}{\sqrt{d_k}}\right)V \tag{1}$$

Where:
Q (Query) : Represents the query vectors or features that require attention.
K (Key) : Represents the key vectors used as references for attention calculation.
V (Value) : Contains the information that will be aggregated based on the attention scores.
d_k : The dimensionality of the query and key vectors, used for score normalization to stabilize training.

Softmax : Converts attention scores into a probability distribution.

$QK^T/\sqrt{d_k}$: Produces a similarity matrix between query and key vectors.
V : Filters and combines value information according to the attention weights.

The encoder output is combined with object queries and processed by the decoder using cross-attention to iteratively refine class and bounding box predictions. RT-DETR also employs an IoU-aware query selection strategy to identify the most informative queries, thereby improving convergence speed and detection accuracy. The model loss function is defined as a combination of classification loss and bounding box regression loss:

$$L_{\text{total}} = L_{\text{cls}} + \lambda L_{\text{bbox}} \tag{2}$$

Where:
Lcls : Classification loss that measures the accuracy of object category predictions.
Lbbox : Bounding box regression loss that measures localization and size prediction errors.
λ : Balancing coefficient that controls the contribution of each loss component.

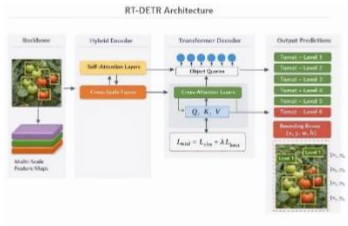


Fig.1: RT-DETR Architecture [16]

Figure 1 illustrates the architecture of Real-Time Detection Transformer (RT-DETR), which consists of three primary components: a CNN backbone, an efficient hybrid encoder, and a transformer decoder operating in an end-to-end manner. Tomato images are first processed by the backbone to extract visual features, which are then refined by the encoder to generate contextual feature representations. Subsequently, the decoder, together with object queries, simultaneously predicts object classes and bounding box coordinates, resulting in accurate object detection and classification [16].

2.2. YOLO

YOLO (You Only Look Once) is a neural network-based object detection algorithm designed to effectively learn patterns from data and identify objects rapidly in real-time applications. The YOLO algorithm was first introduced in 2015 through the research paper entitled "You Only Look Once: Unified, Real-Time Object Detection," which marked a significant milestone in the development of object detection technology by providing a more efficient and accurate approach compared to previous methods. Since then, YOLO has become one of the most widely adopted object detection techniques in various computer vision applications [17].

YOLO has undergone significant advancements since its initial introduction, with multiple versions developed to improve its performance. Each new version has introduced substantial enhancements in terms of accuracy, detection speed, and computational efficiency, making the algorithm increasingly reliable for diverse implementations. Its ability to process images in real time using deep neural networks has established YOLO as one of the most important and relevant methods in computer vision, particularly for rapid and accurate object identification and localization under various conditions [18].

The YOLO algorithm detects objects by predicting bounding box locations, confidence scores, and class probabilities for each object within predefined grid cells. YOLO employs a convolutional neural network that produces an output matrix represented as:

$$Y = S \times S \times B \times (5 + C) \tag{3}$$

Where:
S : Number of grid cells in the image.
B : Number of bounding boxes predicted for each grid cell.
C : Number of object classes to be detected.

3. Results and Discussion

The dataset was divided into three subsets: training data, validation data, and testing data. The data distribution showed that approximately 87% of the dataset was allocated for training, 8% for validation, and 4% for testing. In terms of the number of images, the training dataset consisted of 1,578 images, the validation dataset contained 151 images, and the testing dataset comprised 76 images.

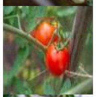
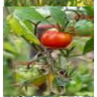
3.1. YOLOv8 Model Testing and Evaluation Results

The testing process using the YOLOv8 algorithm was conducted to comprehensively evaluate the model's ability to detect and classify tomato ripeness levels into six categories, namely Level 1 through Level 6. This process was crucial because it utilized testing data that were completely separate from the training dataset, thereby providing an objective assessment of the model's performance. The testing data served as the primary reference for evaluating prediction accuracy on both static images and real-time camera-based inputs.

During the testing phase, the model's performance was evaluated by comparing the detection results generated by YOLOv8 with the ground truth obtained from the annotation (.txt) files associated with the testing dataset. These annotation files contained predefined tomato ripeness labels and served as the primary reference for assessing the accuracy of the model's predictions. This comparison process was essential for accurately identifying and calculating the values of True Positive (TP), False Positive (FP), and False Negative (FN).

True Positive (TP) represents the number of tomato objects that were correctly detected and classified according to the ground truth labels, thereby reflecting the model's capability to accurately recognize objects. In contrast, False Positive (FP) indicates detection errors in which the model identifies objects that do not correspond to the actual labels, including cases where non-tomato objects are incorrectly detected as tomatoes. Meanwhile, False Negative (FN) represents instances in which the model fails to detect tomato objects that are actually present in the testing data. These three metrics constitute critical components of the evaluation process because they directly reflect the accuracy, reliability, and consistency of the model in object detection and classification tasks.

Table 1: YOLOv8 Detection Results Based on Ground Truth and Predicted Labels
Source: Author's Analysis, 2026

Image	*Ground Truth*	Prediction	Conf (%)	TP	FP	FN
	5, 5	6, 5, 1, 4	42.80 %	1	3	1
	5, 4, 6, 1	6, 5, 1, 5, 3	84.12 %	3	2	1
	5, 5	6, 6, 1	92.95 %	0	3	2
	6, 6	6, 6	92.02 %	2	0	0
	5, 4, 6	4, 6, 4, 5	88.28 %	3	1	0
	4, 1, 4	1, 5, 6, 5	61.28 %	1	3	2
	6, 1, 1, 2	1, 6, 1	84.05 %	3	0	1
	6, 5, 6, 2	6, 5, 3, 6, 2	68.11 %	4	1	0
	5, 1, 1	6, 1, 1	70.20 %	2	1	1
	6, 1	6, 1	94.50 %	2	0	0

Based on the True Positive (TP), False Positive (FP), and False Negative (FN) values presented in Table 1, the performance evaluation metrics of the YOLOv8 model were calculated, including Accuracy, Precision, and Recall. This stage is highly important as it provides a comprehensive assessment of the model's performance, ranging from its accuracy in detecting tomato objects, its ability to minimize detection errors, to its effectiveness in identifying all objects present in the test images. The results of this evaluation serve as the primary basis for assessing the reliability of the model and determining its suitability for implementation in a digital image-based tomato sorting system.

Table 2: Performance Evaluation Metrics of the YOLOv8 Model
Source: Author's Analysis, 2026

TP	FP	FN	*Accuracy* (%)	*Precision* (%)	*Recall* (%)
1	3	1	20.00	25.00	50.00
3	2	1	50.00	60.00	75.00
0	3	2	0.00	0.00	0.00
2	0	0	100.00	100.00	100.00
3	1	0	75.00	75.00	100.00
1	3	2	16.67	25.00	33.33
3	0	1	75.00	100.00	75.00
4	1	0	80.00	80.00	100.00
2	1	1	50.00	66.67	66.67
2	0	0	100.00	100.00	100.00

In general, the testing results presented in Table 2 indicate that the YOLOv8 algorithm demonstrates a significant capability in detecting and classifying tomato ripeness levels, although its accuracy varies across different testing scenarios. These variations are influenced by several critical factors, including the number of objects within the image, lighting conditions, image acquisition angles, and the presence of other objects that may lead to detection errors. Nevertheless, the performance exhibited by YOLOv8 remains satisfactory and consistent in recognizing tomato ripeness levels based on the testing data used. This finding serves as an important indicator that the model possesses strong potential for further development, particularly through evaluation using external datasets to support the implementation of a more accurate and reliable system in real-world environments.

3.2. RT-DETR Model Testing and Evaluation Results

The testing process using the RT-DETR algorithm was conducted to evaluate the model's capability in detecting and classifying tomato ripeness levels into six categories, namely Level 1 through Level 6. The evaluation was performed using the same testing dataset employed for the YOLOv8 model, including both static image inputs and real-time camera captures, to ensure an objective performance comparison. Unlike YOLOv8, which is categorized as a one-stage detector, RT-DETR utilizes a Transformer-based approach in its detection process. This architectural difference results in distinct detection characteristics, particularly in handling variations in the number of objects and object positions within an image. The detection results generated by the RT-DETR model were subsequently compared with the ground truth data to obtain the values of True Positive (TP), False Positive (FP), and False Negative (FN) for each test sample. These values were then used as the basis for calculating the model performance evaluation metrics.

Table 3: RT-DETR Detection Results Based on Ground Truth and Model Predictions
Source: Author's Analysis, 2026

Image	***Ground Truth***	**Prediction**	**Conf (%)**	**TP**	**FP**	**FN**
	5, 5	5, 1, 6, 1, 6, 5	58.62 %	2	4	0
	5, 4, 6, 1	6, 5, 1, 5, 1	79.13 %	3	2	1
	5, 5	6, 6, 1	79.42 %	0	3	2
	6, 6	6, 6	62.33 %	2	0	0
	5, 4, 6	6, 5, 4	84.69 %	3	0	0
	4, 1, 4	1, 5, 5, 1, 6, 1, 5, 1, 1, 1, 5, 1	68.91 %	1	11	2
	6, 1, 1, 2	6, 1, 5, 1, 1	66.11 %	3	2	1
	6, 5, 6, 2	6, 5, 3, 5, 5, 5	66.96 %	2	4	2
	5, 1, 1	6, 1, 1	91.02 %	2	1	1
	6, 1	6, 1	89.40 %	2	0	0

Based on the True Positive (TP), False Positive (FP), and False Negative (FN) values presented in Table 3, the performance evaluation metrics of the RT-DETR model were calculated, including Accuracy, Precision, and Recall. This stage is particularly important because it provides a comprehensive assessment of the model's performance, ranging from its accuracy in detecting tomato objects and its ability to minimize detection errors to its effectiveness in identifying all objects present in the test images. The results of this evaluation serve as the primary basis for assessing the reliability of the model and determining its suitability for implementation in a digital image-based tomato sorting system.

Table 4: Performance Evaluation Metrics of the RT-DETR Model
Source: Author's Analysis, 2026

TP	**FP**	**FN**	***Accuracy* (%)**	***Precision* (%)**	***Recall* (%)**
2	4	0	33.33	33.33	100.00
3	2	1	50.00	60.00	75.00
0	3	2	0.00	0.00	0.00
2	0	0	100.00	100.00	100.00

3	0	0	100.00	100.00	100.00
1	11	2	7.14	8.33	33.33
3	2	1	50.00	60.00	75.00
2	4	2	25.00	33.33	50.00
2	1	1	50.00	66.67	66.67
2	0	0	100.00	100.00	100.00

In general, the testing results presented in Table 4 indicate that the RT-DETR algorithm demonstrates a significant capability in detecting and classifying tomato ripeness levels, although its accuracy varies across different testing scenarios. These variations are influenced by several critical factors, including the number of objects within the image, lighting conditions, image acquisition angles, and the presence of other objects that may lead to detection errors. Nevertheless, RT-DETR shows strong potential for implementation in a web-based tomato ripeness detection system, particularly in real-time detection scenarios.

3.3. Comparison of YOLOv8 and RT-DETR Model Testing and Evaluation Results

Following the testing phase using the YOLOv8 and RT-DETR models, the next step involved conducting a comparative analysis of the performance of both models based on the evaluation results obtained. This analysis aims to provide a comprehensive overview of the performance of each model in detecting and classifying tomato ripeness levels within the developed web-based system.

Table 5: Comparative Summary of YOLOv8 and RT-DETR Model Evaluation Results
Source: Author's Analysis, 2026

Method	Total TP	Total FP	Average Accuracy (%)	Average Precision (%)	Average Recall (%)
YOLO	30	18	57.00	63.70	71.60
RT-DETR	30	32	63.04	55.10	70.44

Based on Table 5, it can be observed that the YOLOv8 and RT-DETR models exhibit different performance characteristics in the tomato ripeness detection process. YOLOv8 achieved an average accuracy of 57.00% with a precision of 63.70%, whereas RT-DETR obtained a higher average accuracy of 63.04% but a lower precision of 55.10%. The higher precision value achieved by YOLOv8 indicates that this model is more effective in minimizing detection errors in the form of false positives, which is also reflected in its lower number of false positive detections compared to RT-DETR. Furthermore, YOLOv8 achieved a slightly higher recall value (71.60%) than RT-DETR (70.44%), demonstrating a better capability to correctly detect tomato objects. Overall, YOLOv8 provides a more balanced performance between detection precision and completeness, while RT-DETR excels in terms of average accuracy but tends to generate a greater number of false positives.

The testing results were also visually presented through images processed by YOLOv8 and RT-DETR, displaying bounding boxes and tomato ripeness labels. This visualization aims to illustrate the performance differences between the two algorithms in terms of object detection count, object localization, and classification accuracy.

The visual detection results show that both YOLOv8 and RT-DETR generate bounding boxes and ripeness level labels for tomato objects in the test images. Differences can be observed in the number of detected objects, bounding box positions, and predicted labels, including several cases of overlapping bounding boxes and objects that were not detected by one of the algorithms.

Table 6: Comparison of Visual Tomato Ripeness Detection Results Using YOLOv8 and RT-DETR Source: Author's Analysis, 2026

YOLOv8	RT-DETR
	
	
	

The determination of tomato ripeness levels was based on the dataset labels, which served as the ground truth for comparing the prediction results generated by YOLOv8 and RT-DETR, as presented in the following table.

Table 7: Comparison of Ground Truth Labels and Prediction Results Generated by the YOLOv8 and RT-DETR Models
Source: Author's Analysis, 2026

Ground Truth (GT)	YOLOv8	RT-DETR
6,6,6,6,6,6	6,6,6,6,6,5,6	6,6,6,6,6,6
5,6,5,6	5,5,5,4,5	6,5,6,5
6,6,6,6,6,6,1,1,1,1	6,6,6,1,6,1,6,1,1	6,6,6,1,1,6,6,6,1,1,1
6,6,3	6,6,2	6,6,3
6,6,6,4,1	6,1,5,6,6,5	1,4,6,6,5,5
1,1,1,1,5,5,6,2	1,1,6,1,6,1,3,5,5,6	1,1,5,1,6,1,5,6,5
6,6,6,6,6,6,6,6,6,6,5,5,4,1	6,5,1,6,6,4,6,6,5,5,6,6,5,6	6,6,1,6,6,6,6,4,6,6,4,5,6,6,6,6

Based on Table 7, the system was tested using new images that were not included in the training dataset, employing both the YOLOv8 and RT-DETR algorithms. The performance of both models was evaluated based on the values of True Positive (TP), False Positive (FP), and False Negative (FN), which were subsequently used to calculate the performance metrics, including Precision, Recall, and Accuracy.

Table 8: TP, FP, FN Values Obtained from the System Evaluation
Source: Author's Analysis, 2026

YOLOv8			RT-DETR		
TP	**FP**	**FN**	**TP**	**FP**	**FN**
6	1	0	6	0	0
2	3	2	4	0	0
9	0	1	10	1	0
2	1	1	3	0	0
2	4	3	3	3	2
5	5	3	5	3	3
9	5	5	11	3	3
35	19	15	42	10	8

Based on Table 8, the True Positive (TP), False Positive (FP), and False Negative (FN) values obtained from the tomato ripeness detection system testing using images that were not included in the dataset were used to calculate evaluation metrics for measuring the performance of each algorithm. The evaluation metrics included Precision, Recall, and Accuracy, which were calculated using the following standard formulas:

$$Precision = \frac{TP}{TP + FP}, Recall = \frac{TP}{TP + FN}, Accuracy = \frac{TP}{TP + FP + FN}$$

For the YOLOv8 algorithm, with a total of TP = 35, FP = 19, and FN = 15, the following results were obtained:

$$Precision_{YOLOv8} = \frac{35}{35 + 19} = \frac{35}{54} = 0{,}6481 = 64{,}8\%$$

$$Recall_{YOLOv8} = \frac{35}{35 + 15} = \frac{35}{50} = 0{,}7 = 70\%$$

$$Accuracy_{YOLOv8} = \frac{35}{35 + 19 + 15} = \frac{35}{69} = 0{,}5072 = 50{,}7\%$$

Meanwhile, for the RT-DETR algorithm, with a total of TP = 42, FP = 10, and FN = 8, the following results were obtained:

$$Precision_{RT-DETR} = \frac{42}{42 + 10} = \frac{42}{52} = 0{,}8077 = 80{,}8\%$$

$$Recall_{RT-DETR} = \frac{42}{42 + 8} = \frac{42}{50} = 0{,}84 = 84\%$$

$$Accuracy_{RT-DETR} = \frac{42}{42 + 10 + 8} = \frac{42}{60} = 0{,}7 = 70\%$$

RT-DETR demonstrated better performance than YOLOv8, achieving a Precision of 80.8%, Recall of 84.0%, and Accuracy of 70.0%, whereas YOLOv8 achieved a Precision of 64.8%, Recall of 70.0%, and Accuracy of 50.7%. These results indicate that RT-DETR is more accurate and consistent in detecting tomato ripeness on previously unseen data, with fewer prediction errors compared to YOLOv8.

4. Conclusion

Based on the results of this study, it can be concluded that both algorithms, namely YOLOv8 and Real-Time Detection Transformer (RT-DETR), demonstrate good performance in detecting and classifying tomato ripeness levels using digital images. YOLOv8 achieved a precision of 64.8%, a recall of 70.0%, and an accuracy of 50.7%, indicating that the model is relatively stable in maintaining a balance between detection precision and completeness while being more effective in minimizing false positive errors. On the other hand, RT-DETR exhibited superior performance, achieving a precision of 80.8%, a recall of 84.0%, and an accuracy of 70.0%. These results indicate that RT-DETR is capable of providing more accurate and consistent predictions, particularly when detecting objects in previously unseen data that were not used during the training process. Therefore, although YOLOv8 offers advantages in terms of detection efficiency and balanced performance, RT-DETR is considered a more suitable approach for implementation in digital image-based tomato sorting systems that require high levels of accuracy and reliability.

5. Suggestions

Based on the findings of this study, several recommendations can be proposed for future research. First, the size of the dataset can be increased by incorporating a wider variety of tomato images captured under different lighting conditions, viewing angles, backgrounds, and ripeness levels to improve the model's generalization capability. Second, future studies may utilize external testing datasets collected from different locations or sources to evaluate the robustness and reliability of the models under real-world conditions. Third, further system development can be conducted by comparing RT-DETR and YOLOv8 with other object detection algorithms, such as YOLOv11, Faster R-CNN, or SSD, in order to identify models with superior performance. In addition, the system can be further enhanced by integrating tomato counting and automatic sorting features based on ripeness levels, thereby supporting harvesting and distribution processes in a more effective and efficient manner.

References

[1] A. A. Pratiwi, A. Rahmawati, A. Kurniasih, E. Nurhillah, N. D. Permatasari, And Y. Budiawati, “Peran Artificial Intelligence Dalam Prediksi Hasil Tanaman: Meta-Sintesis Literatur Terkini,” *J. Pertanian, Peternakan, Perikan.*, Vol. 4, No. 1, Pp. 1–15, 2025, Doi: 10.3766/Hibrida.V1i2.3753.

[2] R. Efendi And D. Sagita, “Teknologi Pertanian Masa Depan Dan Peranannya Dalam Menunjang Ketahanan Pangan,” *Sultra J. Mech. Eng.*, Vol. 1, No. 1, Pp. 1–12, 2022, Doi: Doi.Org/10.54297/Sjme.V1i1.297.

[3] E. Andriani, H. Nur, H. M. Arif, And A. Anjaly, “Pengolahan Selai Tomat Sebagai Prodak Bernilai Ekonomi Tinggi,” Vol. 3, No. May, Pp. 128–134, 2025, Doi: Doi.Org/10.61231/Jp2m.V3i2.3661.

[4] N. Cornelia, Sintiawati, A. Febriansyah, And N. Khasanah, “Implementasi Sensor Berat Dan Gas Pada Sistem Sortasi Tomat Otomatis,” Vol. 17, No. 1, Pp. 170–177, 2026, Doi: Doi.Org/10.31602/Tji.V17i1.21294.

[5] A. S. Herawati, D. A. P. Pancayudha, H. Faidillah, And A. D. Putra, “Penerapan Sistem Otomasi Dalam Penyortiran Tomat Untuk Meningkatkan Produktivitas Pemasaran Hasil Pertanian,” Vol. 02, No. 02, Pp. 90–97, 2025, Doi: Doi.Org/10.61105/Jieti.V2i2.197.

[6] A. O. Gumelar, S. P. Wirman, And N. Fitriya, “Pengembangan Sistem Penyortir Telur Otomatis Berbasis Iot Dengan Integrasi Sensor Massa Dan Ukuran,” Vol. 6, No. 3, Pp. 230–245, 2025.

[7] Lusiyana, E. A. Fasha, R. F. Susanto, D. A. J. D. Putri, F. Fadilatunnisyah, And Y. Budiawati, “Artificial Intelligence Untuk Optimasi Penyimpanan Pasca Panen: Telaah Literatur Sistematis,” *Hibrida J. Pertanian, Peternakan, Perikan.*, Vol. 4, No. 1 Se-Articles, Pp. 51–60, 2025, [Online]. Available: Https://Cibangsa.Com/Index.Php/Hybridajournal/Article/View/828

[8] R. Naturizal, W. Fuadi, And L. Rosnita, “Sistem Pendeteksi Tingkat Kesegaran Daging Ayam Pada Citra Menggunakan Metode Convolutional Neural Network (Cnn) Berbasis Android,” Vol. 8, No. 2, Pp. 301–312, 2024.

[9] D. H. Saputra, B. Imran, And Juhartini, “Object Detection Untuk Mendeteksi Citra Buah Buahan Menggunakan Metode Yolo,” *J. Kecerdasan Buatan Dan Teknol. Inf.*, Vol. 2, No. 2, Pp. 70–80, 2023, Doi: Doi.Org/10.69916/Jkbti.V2i2.18.

[10] R. Praseli, N. S. Fatonah, D. Aryani, And H. D. Ariessanti, “Implementasi Perbandingan Yolo V8 , V9 , Dan V11 Dalam Penerapan Tata Tertib K3 : Deteksi Penggunaan Helm Keselamatan Di Lingkungan Konstruksi,” Vol. 6, No. April, Pp. 513–523, 2026, Doi:

Doi.Org/10.57152/Malcom.V6i2.2554.
[11] Yanto, F. Aziz, And Irmawati, "Yolo-V8 Peningkatan Algoritma Untuk Deteksi Pemakaian Masker Wajah," Vol. 7, No. 3, Pp. 1437–1444, 2023, Doi: Doi.Org/10.36040/Jati.V7i3.7047.
[12] F. A. Siwaka, S. D. Panjaitan, And R. R. Yacoub, "Implementasi Algoritma Deep Learning Yolov8 Pada Alat Sortir Kematangan Buah Tomat," Vol. 7, No. 10, Pp. 1–11, 2025, Doi: Doi.Org/10.2238/52eqv641.
[13] T. Raharjo, H. P. Putro, And H. E. Sari, "Early Detection Of Tomato Plant Diseases Using The Real-Time Detection Transformer (Rt-Detr) Model," Vol. 5, No. 1, Pp. 584–592, 2026, Doi: Doi.Org/10.51903/286bt086.
[14] N. Aula, M. Ula, And L. Rosnita, "Analisis Sentimen Review Customer Terhadap Perusahaan Ekspedisi Jne , J & T Express Dan Pos Indonesia Menggunakan Metode Support Vector Machine (Svm) Analysis Of Customer Review Sentiment To Jne , J & T Express And Pos Indonesia Expedition Companies Using Svm Method," Vol. 9, No. 1, Pp. 81–86, 2023.
[15] Adrik Fikhtiyaaril Amro And O. V. Putra, "Pengembangan Pengendali Karakter Permainan Menggunakan Suara Real-Time Berbasis Transformer," *Pros. Semin. Nas. Sains Dan Teknol.*, Vol. 15, No. 1 Se-Articles, Pp. 277–282, Jan. 2026, Doi: 10.36499/Psnst.V15i1.14846.
[16] Y. Zhao, W. Lv, S. Xu, J. Wei, G. Wang, Q. Dang, And J. Chen, "Detrs Beat Yolos On Real-Time Object Detection," In Proc. Ieee/Cvf Conf. Comput. Vis. Pattern Recognit. (Cvpr), Pp. 16965–16974, 2024.
[17] A. N. Sugandi, B. Hartono, And K. Kunci, "Implementasi Pengolahan Citra Pada Quadcopter Untuk Deteksi Manusia Menggunakan Algoritma Yolo," Pp. 13–14, 2022, Doi: Doi.Org/10.35313/Irwns.V13i01.4186.
[18] A. A. Permana, M. R. Muttaqin, And M. A. Sunandar, "Sistem Deteksi Api Secara Real Time Menggunakan Algoritma You Only Look Once (Yolo) Versi 8," Vol. 8, No. 5, Pp. 10395–10400, 2024.