Automating Tomato Ripeness Classification and Counting with YOLOv9

Hoang-Tu Vo, Kheo Chau Mui, Nhon Nguyen Thien, Phuc Pham Tien
Information Technology Department
FPT University, Cantho city, Vietnam

***Abstract*—This article proposes a novel solution to the long-standing issue of ripe (or manual) tomato monitoring and counting, often relying on visual inspection, which is both time-consuming, requires a lot of labor and prone to inaccuracies. By leveraging the power of artificial intelligence (AI) and image analysis techniques, a more efficient and precise method for automating this process is introduced. This approach promises to significantly reduce labor requirements while enhancing accuracy, thus improving overall quality and productivity. In this study, we explore the application of the latest version of YOLO (You Only Look Once), specifically YOLOv9, in automating the classification of tomato ripeness levels and counting tomatoes. To assess the performance of the proposed model, the study employs standard evaluation metrics including Precision, Recall, and mAP50. These metrics provide valuable insights into the model's ability to accurately detect and count tomatoes in real-world scenarios. The results indicate that the YOLOv9-based model achieves superior performance, as evidenced by the following evaluation metrics: Precision: 0.856, Recall: 0.832, and mAP50: 0.882. By leveraging YOLOv9 and comprehensive evaluation metrics, this research aims to provide a robust solution for automating tomato monitoring processes. Additionally, by offering the future integration of robotics, the collection phase can further optimize efficiency and enable the expansion of cultivation areas.**

Keywords*—Tomato monitoring; manual counting; Artificial Intelligence (AI); Image analysis techniques; YOLO; YOLOv9

I. Introduction

Tomatoes offer not just delightful flavor but also contain crucial nutrients. They are a great source of vitamin C, which supports the immune system and promotes healthy skin. Additionally, tomatoes contain lycopene, a powerful antioxidant that may help reduce the risk of certain cancers and protect against heart disease [1], [2]. Tomatoes have an attractive moisture content of 95%, with a carbohydrate content of 3%, protein at 1.2%, and total lipids making up 1%. Furthermore, they offer minerals including calcium (Ca), magnesium (Mg), phosphorus (P), potassium (K), sodium (Na), zinc (Zn), and manganese (Mn). In addition to minerals, tomatoes provide essential vitamins such as vitamins A and C, thiamin, riboflavin, niacin, pantothenic acid, and pyridoxine. [3]. In 2020, the largest producers of tomatoes worldwide were as follows: China took the top spot, producing an impressive 64,866 million tons of tomatoes in a single year. India came in second, producing approximately 20,573 million tons of tomatoes, while Turkey ranked third, with a tomato production of 13,204 million tons [4].

The current problem of identifying, manually counting, and classifying the ripeness of tomatoes persists as a significant challenge in agricultural practices. Traditional methods rely heavily on manual labor, making the process time-consuming, labor-intensive, and prone to errors. The current challenges in tomato handling demand innovative solutions that combine machine learning, image processing, and automation to enhance efficiency, reduce errors, and improve overall productivity in the tomato industry.

Manual counting, which involves counting tomatoes manually during harvesting or quality control, is labor-intensive and inefficient. It leads to inaccuracies due to fatigue, distractions, and variations in human perception. Automating counting using image processing, machine learning, or deep learning could alleviate this issue. A study [5] focused on detecting and counting tomato fruits in greenhouses utilizing deep learning.

Accurately categorizing tomatoes into ripeness stages (such as unripe, ripe, and overripe) plays a pivotal role in sorting, storage, and distribution within the agricultural industry. However, manual classification suffers from inconsistency due to human subjectivity. To address this, researchers have proposed innovative approaches, including Machine Learning (ML), Convolutional Neural Networks (CNNs) and Deep Learning-based methods (DL), which demonstrate promising results in fruit classification and ripeness determination [6], [7]. A study [8] using the Cascaded Object Detector (COD) and a composition of traditional custom image processing methods. The COD method achieved 95% accuracy in detecting ripe tomatoes, outperforming the traditional Color Segmentation Method.

This study introduces a more efficient and accurate approach to automating the monitoring process. The utilization of the latest version of YOLO, specifically YOLOv9, enables the classification of tomato ripeness levels and facilitates tomato counting. The main contribution of the study are:

- Introducing a novel solution to the longstanding problem of manual tomato monitoring and counting, addressing issues of time consumption, labor intensity, and inaccuracies associated with visual inspection methods.
- Leveraging artificial intelligence (AI) and image analysis techniques to develop a more efficient and precise method for automating tomato monitoring processes, promising to significantly reduce labor requirements while enhancing accuracy and overall quality and productivity.
- Exploring the application of the latest version of YOLO, specifically YOLOv9, in automating the classification of tomato ripeness levels and counting toma-

toes, demonstrating the potential of advanced deep learning techniques in agricultural applications.

- Evaluating the performance of the proposed model using standard evaluation metrics such as Precision, Recall, and mAP50, providing valuable insights into its effectiveness in accurately detecting and counting tomatoes in real-world scenarios.

The paper is organized as follows: In Section II, we present a thorough literature review. Section III outlines the Automated Tomato Ripeness Classification and Counting Methodology, including details about the dataset, data preparation, and evaluation metrics for the model. Moving on to Section IV, we delve into the experimental system and present the final results. Finally, Section V summarizes our study's findings and offers concluding remarks. Lastly, Section VI outlines potential avenues for future study.

II. Related Works

Computer vision has emerged as a powerful tool in modern agriculture, revolutionizing the way crops are monitored and managed [9], [10], [11], [12], [13], [14] from object detection algorithms based on traditional methods to modern approaches such as CNN and deep learning.

The authors in the article [15] introduces an automated multi-class classification method for evaluating tomato ripeness using color features and employing Principal Components Analysis (PCA), Support Vector Machines (SVMs), and Linear Discriminant Analysis (LDA) algorithms for feature extraction and classification.

In this study [16], the authors utilize digital image processing techniques to describe and extract color statistics (RGB, HSI, and Lab* color spaces) from tomato images. They employ supervised and unsupervised classification algorithms such as K-NN, MLP neural networks, and K-Means for classifying Milano and Chonto tomatoes.

Liu et al. in this study [17] propose an algorithm for automatic tomato detection in regular color images, utilizing Histograms of Oriented Gradients (HOG) descriptor trained with a SVM classifier, along with a coarse-to-fine scanning method and False Color Removal (FCR) technique to enhance accuracy. The proposed algorithm demonstrates a significant improvement in tomato detection compared to other methods, achieving high recall, precision, and F1 score percentages of 90.00%, 94.41%, and 92.15%, respectively, in test images.

This study [18] proposes a Tomato Classification model utilizing Machine Learning algorithms such as Decision Tree (DT), Logistic Regression (LR), Gradient Boosting (GB), Random Forest (RF), SVM, K-NN, and XG Boost to determine tomato maturity stages, with Random Forest achieving the highest accuracy of 92.49% among the classifiers tested.

The author in the research [19] aims to apply deep Transfer Learning (TL) to classify tomatoes into maturity classes, employing three TL approaches—VGG, Inception, and ResNet—ultimately revealing VGG 19 as the top performer.

The authors in this work [20] utilize an improved DenseNet architecture to address the challenges of accurately classifying tomato ripeness in complex images, incorporating structured sparse operations to enhance feature propagation and reduce data storage, as well as introducing the Focal loss function to mitigate dataset imbalance and improve classification accuracy in their tomato detection system.

Utilizing TL with VGG16 for Fruit Ripeness Detection. This study [21] demonstrates that DL employing TL consistently outperforms traditional ML approaches utilizing traditional feature extraction for fruit ripeness detection.

The author's primary objective in this study [22] is to introduce a new method for sorting and grading tomato quality, the approach integrates pre-trained CNNs for feature extraction with conventional ML algorithms (such as SVM, RF, and k-Nearest Neighbors (KNN)) to enhance classification accuracy. Among the hybrid models proposed, the CNN-SVM method demonstrates superior performance, achieving high accuracies in both binary and multiclass classification tasks, particularly when utilizing Inceptionv3 as the feature extractor.

The authors in [23] introduce four distinct deep learning frameworks, Utilizing a combination of Yolov5m and deep learning models—specifically ResNet50, ResNet-101, and EfficientNet-B0 - the model successfully classified tomatoes on the vine into three distinct classes: ripe, immature, and damaged. The evaluation results indicated that the ResNet-50 and EfficientNet-B0 achieved impressive overall accuracy of 98%, while the Yolov5m and ResNet-101 models demonstrated accuracy of 97%.

This study [24] explores tomato segmentation and detection across various maturity stages, utilizing both a mask R-CNN and a YOLOv8 model. Evaluation metrics show that mask R-CNN achieved 67.2% average precision with 78.9% recall, and 92.1% IoU average precision with 91.4% recall, while YOLOv8 demonstrated superior performance, With coefficients of determination measuring 0.809 for ripe, 0.897 for half-ripe, and 0.968 for green categories.

Liu, Guoxu, et al. in this study [25] introduces an enhanced fruit detection model named YOLO-Tomato, derived from YOLOv3. YOLO-Tomato integrates a dense architecture into YOLOv3, enabling feature reuse and enhancing model accuracy, while also employing circular bounding boxes for more accurate localization of tomatoes.

The authors in this research [26] enhanced YOLOv5 to identify four distinct stages of tomato ripeness: mature green, breaker, pink, and red. [27] introduces a novel lightweight enhanced algorithm based on YOLOv5 to achieve real-time tracking and identify the ripeness of tomato fruits, achieved by reconstructing YOLOv5's backbone network utilizing the bneck module of MobileNetV3.

This study presents a more efficient and accurate method to automate the monitoring process. By taking advantage of the latest version of YOLO, specifically YOLOv9, it allows classification of tomato ripeness levels and simplifies tomato counting.

III. Methodology

A. The Process of Gathering and Preparing Data

This research utilizes the FruitDetectionv3 Image Dataset, accessible at [28], which consists of three classes (Tomato

Fig. 1. Sample images of tomatoes at different maturity levels from this dataset.

Tomato Fully-ripe   Tomato Semi-ripe   Tomato Unripe

Fig. 2. Tomato ripening level.

Fully-ripe, Tomato Semi-ripe, and Tomato Unripe) and the total number of images in this dataset is 2610. The dataset is divided into three sets: the training set comprises 2283 pictures (87%), The validation set includes 217 pictures (8%), and the testing set contains 110 pictures (4%). Each image in the dataset has a size of 640x640 pixels. Augmentations are applied to enhance the dataset, including Horizontal Flip, Rotation between -15° and +15°, and Shear of ±15° horizontally and vertically. These augmentations aim to increase the variability of the dataset and improve the robustness of the model in real-world scenarios. Sample images of tomatoes at different maturity levels from this dataset are shown in Fig. 1 and samples of Tomato Ripening Level are shown in Fig. 2.

B. *Overall Methodology*

Object detection techniques are frequently classified into one-stage and two-stage approaches. YOLO (You Only Look Once) [29] and SSD (Single Shot MultiBox Detector) [30] are prominent examples of one-stage methods. These methods directly predict bounding boxes and class labels in a single forward pass through the neural network. They are faster in terms of inference time since they avoid the region proposal step. Faster R-CNN (Region-based Convolutional Neural Network) [31] exemplifies the two-stage approach. In the first stage, Faster R-CNN proposes region proposals using a Region Proposal Network (RPN). In the second stage, these proposals are refined to obtain accurate bounding boxes and class predictions. One-stage methods prioritize speed and simplicity, while two-stage methods focus on accuracy at the cost of increased complexity and computation time. In real-time applications where speed is essential, such as autonomous vehicles and surveillance, one-stage methods like YOLO or SSD should be considered.

If achieving high accuracy is crucial and computational resources are available, consider using two-stage methods such as Faster R-CNN. Hence, in this study, we utilize the state-of-the-art one-stage object detection method, YOLOv9, to automate the classification and counting of tomato maturity.

YOLOv9 is a remarkable advancement in real-time object detection technology [32]. YOLOv9 is the latest version of YOLO, released in February 2024, YOLOv9 introduces groundbreaking techniques such as Programmable Gradient Information -PGI and the Generalized Efficient Layer Aggregation Network - GELAN.

1) Programmable Gradient Information - PGI: During the forward pass in neural networks, information can get diluted or lost due to transformations within the layers. This phenomenon is known as the information bottleneck. Gradients provide essential information for updating network weights during training. Accurate gradients are crucial for effective learning. PGI ensures that gradient information is preserved throughout the network. It prevents the loss of critical input information during backpropagation. By maintaining reliable gradient information, PGI helps the model learn more effectively and improves its ability to recognize objects. The YOLOv9 Programmable Gradient Information (PGI) Architecture is shown in Fig. 3. The PGI primarily comprises three components: The main branch, An auxiliary reversible branch, and Multi-level auxiliary information.

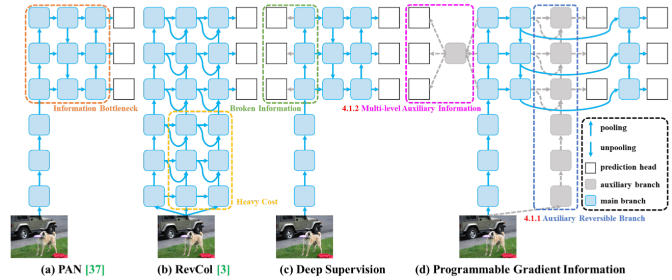

Fig. 3. YOLOv9 PGI Architecture [32].

2) Generalized Efficient Layer Aggregation Network - GELAN: GELAN is a novel architectural advancement, it combines principles from two existing techniques: CSPNet (Cross Stage Partial Network) and ELAN (Efficient Layer Aggregation Network). GELAN is a lightweight network architecture designed based on gradient path planning. It efficiently aggregates information across layers. It prioritizes lightweight design, fast inference, and accuracy. GELAN directly tackles the information bottleneck problem, leading to improved efficiency and accuracy in real-time object detection. The architecture of GELAN within YOLOv9 is shown in Fig. 4.

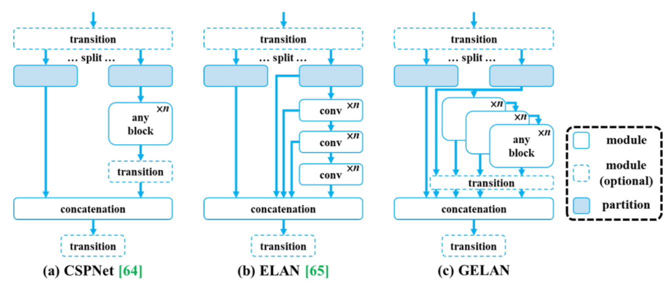

Fig. 4. The architecture of GELAN within YOLOv9 [32].

Information of randomly initialized weight output feature maps across various deep learning network architectures are shown in Fig. 5. From Fig. 5, it's observable that the GELAN architecture retains a significant amount of information from the input data after going through the feed-forward process

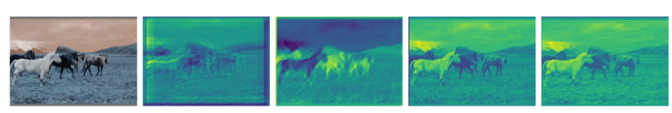

Fig. 5. Information of randomly initialized weight output feature maps across various deep learning network architectures [32].

The analysis of YOLOv9 in comparison to state-of-the-art (SOTA) models demonstrates notable enhancements across diverse metrics. YOLOv9 surpasses current methodologies in parameter efficiency, demanding fewer parameters while either maintaining or enhancing accuracy levels. Comparison of cutting-edge real-time object detectors with YOLOv9 is shown in Fig. 6. YOLOv9 stands out as an innovative model, combining PGI and GELAN to redefine the boundaries of efficiency and accuracy in object detection. Therefore, in this study we use YOLOv9 to count and classify the ripeness level of tomatoes.

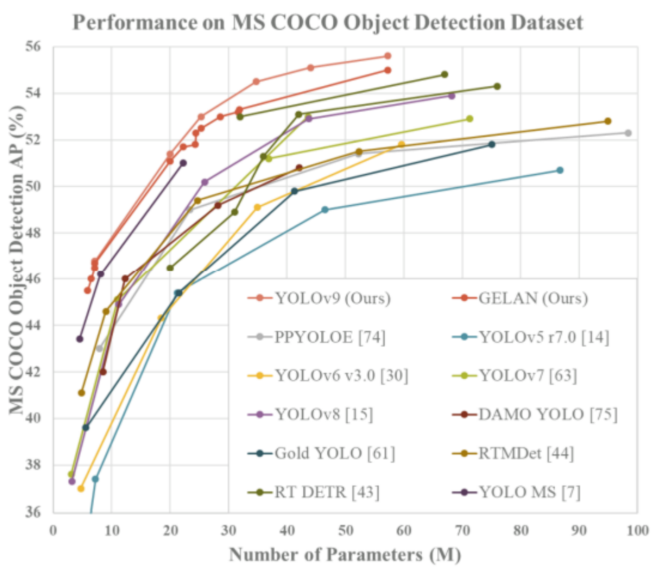

Fig. 6. Comparison of cutting-edge real-time object detectors with YOLOv9 [32].

C. Performance Evaluation Measures

Assessing classification models entails a thorough examination using several essential metrics. Precision, which gauges the correctness of positive predictions among all predicted positives, and recall, which highlights the proportion of accurately predicted positives among all actual positives, play crucial roles. Lastly, The mean average precision (mAP) metric is utilized. It evaluates the detected bounding box by comparing it with the ground-truth box and assigns a corresponding score.

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

$$mAP = \frac{1}{N}\sum_{i=1}^{N} APi \tag{3}$$

In which, FP represent False Positive, TN denote True Negative, TP signify True Positive, and FN indicate False Negative. AP is Average Precision, APi denotes the average precision value for the i-th category, N is number of classes.

IV. Results

A. Environmental Settings

Our experimental procedures were conducted on the Kaggle platform to acquire the experimental outcomes. The research employed a Tesla T4x2 GPU with 30GB of memory, while the system itself possessed 29GB of RAM. GPU information is presented in Fig. 7.

```
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 535.129.03             Driver Version: 535.129.03   CUDA Version: 12.2     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M | Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf  Pwr:Usage/Cap |         Memory-Usage | GPU-Util  Compute M. |
|                                |                      |               MIG M. |
|===============================+======================+======================|
|   0  Tesla T4             Off | 00000000:00:04.0 Off |                    0 |
| N/A   36C    P8      9W /  70W |      0MiB / 15360MiB |      0%      Default |
|                                |                      |                  N/A |
+-------------------------------+----------------------+----------------------+
|   1  Tesla T4             Off | 00000000:00:05.0 Off |                    0 |
| N/A   37C    P8      9W /  70W |      0MiB / 15360MiB |      0%      Default |
|                                |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

Fig. 7. GPU information used in training models.

B. Experiment

The hyperparameters of the model for automating tomato maturity classification and counting using YOLOv9 are shown in the Table I.

TABLE I. Configuration Parameters for Model Training

Parameters	Value
Batch-size	16
Epochs	100
Image-size	640 × 640
Learning rate (LR)	0.01
Momentum	0.937
Warmup epochs	3
Weight decay	0.0005
Optimizer	Stochastic Gradient Descent (SGD)

1) Comparative Analysis of YOLOv8 and YOLOv9 for tomato counting and ripeness classification in image processing: In this research, the goal was to develop an efficient and accurate model for counting and classifying the ripeness level of tomatoes in images. In this experiment, we utilized two popular object detection frameworks: YOLOv8 and YOLOv9. Both models were trained on a dataset of tomato images. The training process involved fine-tuning the pre-trained YOLOv8 and YOLOv9 architectures on the tomato dataset. The models were optimized for counting and classifying the ripeness level of tomatoes. After training, we evaluated the performance of both models on a separate testing set. The table presents the results of the comparison between YOLOv8 and YOLOv9 methods on the testing set, shown in Table II. The results presented in Table II show that YOLOv9 outperformed YOLOv8 in terms of tomato accuracy classification. it's evident that the YOLOv9 model generally outperforms the YOLOv8 model in terms of class precision, recall, and mAP50 across all tomato ripeness categories. Additionally, the YOLOv9 model achieves comparable or better performance with a slightly smaller model size, indicating potential efficiency improvements.

2) Plots describe the training and validation performance of the YOLOv9 model for counting and classifying tomato ripeness levels: Fig. 8 displays instances of class distribution and visualizations related to object detection.

TABLE II. The Table Presents the Results of the Comparison between YOLOv8 and YOLOv9 Methods on the Testing Set

Models	Class	Precision	Recall	mAP50	Model-Size
YOLOv8	All	0.837	0.771	0.825	52Mb
	Tomato Fully-ripe	0.856	0.822	0.859	
	Tomato Semi-ripe	0.773	0.710	0.772	
	Tomato Unripe	0.881	0.781	0.844	
YOLOv9	All	0.856	0.832	0.882	51.5Mb
	Tomato Fully-ripe	0.860	0.840	0.894	
	Tomato Semi-ripe	0.815	0.785	0.829	
	Tomato Unripe	0.894	0.870	0.925	

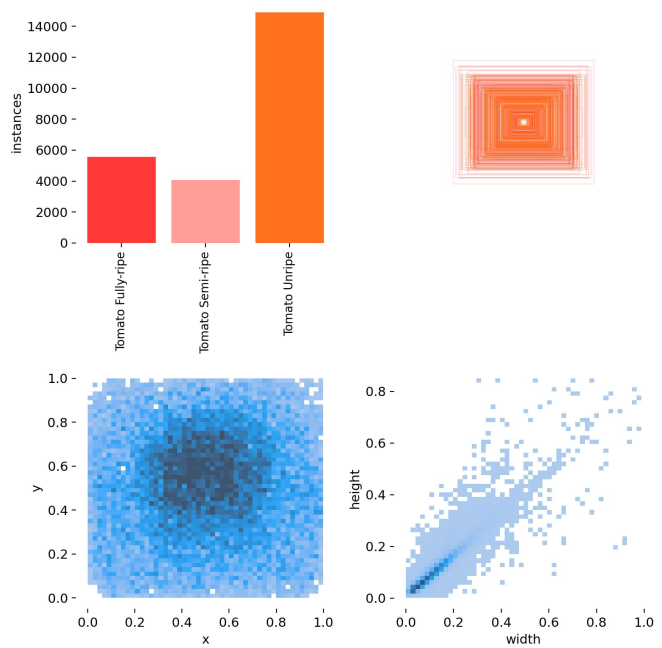

Fig. 8. Overview instances of class distribution and visualizations related to object detection.

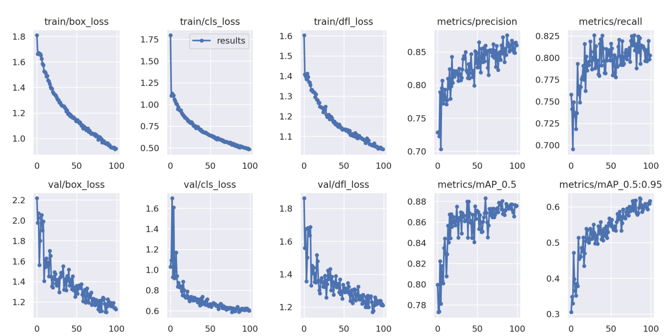

Fig. 9. Visualizations of mean Average Precision (mAP) and loss trends post-training the YOLOv9 model for counting and classifying tomato ripeness levels.

Fig. 9 depicts a series of eight graphs representing different metrics during the training and validation phase of the model to count and classify tomato ripeness.The metrics include: Box Loss, Classification Loss, Distribution Focal Loss, Precision, Recall, Mean Average Precision (mAP). Overall, the graphs show a positive trend over epochs. Decreasing values for loss metrics (Box Loss, Classification Loss, Distribution Focal Loss) indicate model improvement. Increasing precision, recall, and mAP values suggest better performance.

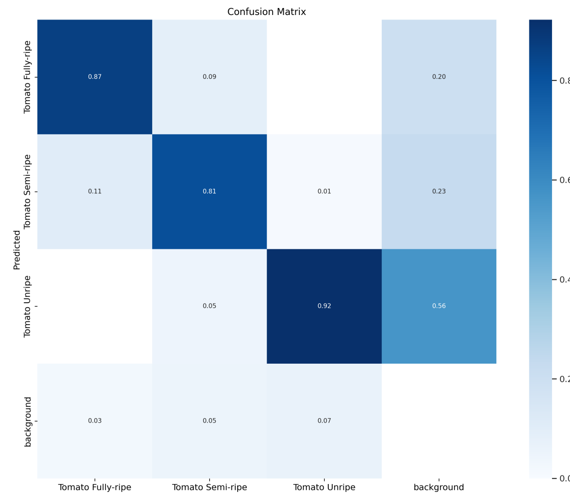

Fig. 10. Confusion matrix of model for counting and classifying tomato ripeness levels.

As we can see from the Confusion matrix in Fig. 10, the tomato ripeness counting and classification model gives the best results in the “Tomato Unripe” class, followed by the “Tomato Fully-ripe” class, and finally, the “Tomato Semi-ripe” class. Precision-Recall Curve is presented in Fig. 11.

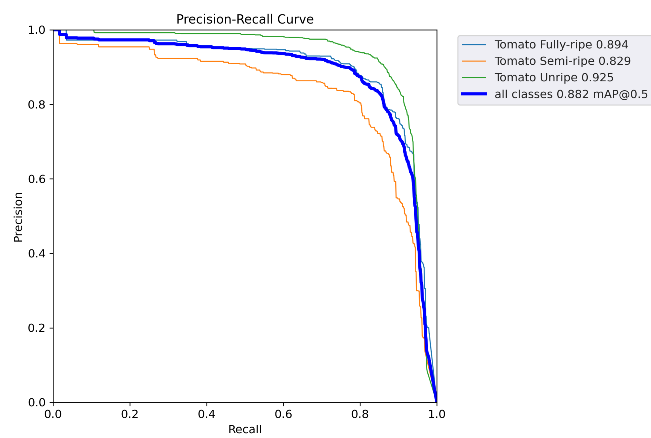

Fig. 11. Precision-Recall curve.

3) The practical effectiveness of detecting and counting tomato ripeness: To validate the practical performance of the built model, six pictures were randomly chosen for this study. These images were sourced from the internet to test the detection and counting of tomato ripeness. The results are depicted in Fig. 12.

V. Conclusion

In conclusion, this article presents a groundbreaking solution to the longstanding challenges associated with manual tomato monitoring and counting. Traditionally, these tasks

Fig. 12. The tomato ripeness counting and classification results.

have been labor-intensive, time-consuming, and prone to inaccuracies due to their reliance on visual inspection. However, by harnessing the capabilities of artificial intelligence (AI) and image analysis techniques, a more efficient and precise method for automating this process is introduced.

The proposed approach, which leverages the latest version of YOLO, specifically YOLOv9, demonstrates promising results in automating the classification of tomato maturity levels and accurately counting tomatoes. Through the utilization of standard evaluation metrics such as Precision, Recall, and mAP50, the study provides valuable insights into the model's performance in real-world scenarios.

The integration of YOLOv9 and comprehensive evaluation metrics aims to offer a robust solution for automating tomato monitoring processes, thereby significantly reducing labor requirements and enhancing accuracy. Furthermore, the potential future integration of robotics in the collection phase presents an opportunity to further optimize efficiency and enable the expansion of cultivation areas.

In essence, this research not only addresses the immediate need for more efficient tomato monitoring methods but also lays the foundation for advancements in agricultural automation, ultimately contributing to improved quality, productivity, and sustainability in tomato cultivation.

Besides, the study also evaluates the use of the latest version of the YOLO (version 9) model on this task to compare the results with the previous version.

VI. Future Works

While the current study demonstrates the effectiveness of the YOLOv9-based model in automating tomato monitoring and counting tasks, several avenues for future research exist to further enhance the proposed solution and extend its applicability. Future research could focus on refining the AI algorithms used for tomato classification and counting. Exploring alternative deep learning architectures or incorporating ensemble techniques may improve the model's performance, particularly in challenging environments with varying lighting conditions or occlusions. Expanding the scope of automation by integrating robotic systems for tomato harvesting and data collection represents a promising direction for future research. By developing autonomous robotic platforms equipped with AI-enabled vision systems, the efficiency and accuracy of tomato cultivation and monitoring processes can be further enhanced.

References

[1] Sanjiv Agarwal and Akkinappally Venketeshwer Rao. Tomato lycopene and its role in human health and chronic diseases. *Cmaj*, 163(6):739–744, 2000.

[2] Md Yousuf Ali, Abu Ali Ibn Sina, Shahad Saif Khandker, Lutfun Neesa, EM Tanvir, Alamgir Kabir, Md Ibrahim Khalil, and Siew Hua Gan. Nutritional composition and bioactive compounds in tomatoes and their impact on human health and disease: A review. *Foods*, 10(1):45, 2020.

[3] Bahare Salehi, Razieh Sharifi-Rad, Farukh Sharopov, Jacek Namiesnik, Amir Roointan, Madhu Kamle, Pradeep Kumar, Natália Martins, and Javad Sharifi-Rad. Beneficial effects and potential risks of tomato consumption for human health: An overview. *Nutrition*, 62:201–208, 2019.

[4] Tomato production by country, available online: https://worldpopulationreview.com/country-rankings/tomato-production-by-country. Accessed: 2024-02-29.

[5] Manya Afonso, Hubert Fonteijn, Felipe Schadeck Fiorentin, Dick Lensink, Marcel Mooij, Nanne Faber, Gerrit Polder, and Ron Wehrens. Tomato fruit detection and counting in greenhouses using deep learning. *Frontiers in plant science*, 11:571299, 2020.

[6] Enoc Tapia-Mendez, Irving A Cruz-Albarran, Saul Tovar-Arriaga, and Luis A Morales-Hernandez. Deep learning-based method for classification and ripeness assessment of fruits and vegetables. *Applied Sciences*, 13(22):12504, 2023.

[7] Nur Azizah Ayunda, Emy Haryatmi, and Tri Agus Riyadi. Classification of tomato ripeness based on convolutional neural network methods. *Journal of Information Systems and Informatics*, 5(4):1658–1675, 2023.

[8] Kazy Noor E Alam Siddiquee, Md Shabiul Islam, Mohammad Yasin Ud Dowla, Karim Mohammed Rezaul, and Vic Grout. Detection, quantification and classification of ripened tomatoes: a comparative analysis of image processing and machine learning. *IET Image Processing*, 14(11):2442–2456, 2020.

[9] Hoang-Tu Vo, Luyl-Da Quach, and Tran Ngoc Hoang. Ensemble of deep learning models for multi-plant disease classification in smart farming. *International Journal of Advanced Computer Science and Applications*, 14(5), 2023.

[10] Hoang-Tu Vo, Nhon Nguyen Thien, and Kheo Chau Mui. Tomato disease recognition: Advancing accuracy through xception and bilinear pooling fusion. *International Journal of Advanced Computer Science and Applications*, 14(8), 2023.

[11] Juliana Freitas Santos Gomes and Fabiana Rodrigues Leta. Applications of computer vision techniques in the agriculture and food industry: a review. *European Food Research and Technology*, 235:989–1000, 2012.

[12] Diego Inácio Patrício and Rafael Rieder. Computer vision and artificial intelligence in precision agriculture for grain crops: A systematic review. *Computers and electronics in agriculture*, 153:69–81, 2018.

[13] Ronald TOMBE. Computer vision for smart farming and sustainable agriculture. In *2020 IST-Africa Conference (IST-Africa)*, pages 1–8. IEEE, 2020.

[14] Hoang-Tu Vo, Nhon Nguyen Thien, and Kheo Chau Mui. A deep transfer learning approach for accurate dragon fruit ripeness classification and visual explanation using grad-cam. *International Journal of Advanced Computer Science & Applications*, 14(11), 2023.

[15] Nashwa El-Bendary, Esraa El Hariri, Aboul Ella Hassanien, and Amr Badr. Using machine learning techniques for evaluating tomato ripeness. *Expert Systems with Applications*, 42(4):1892–1905, 2015.

[16] Wolffang D Niño Pacheco and Fabián R Jiménez López. Tomato classification according to organoleptic maturity (coloration) using machine learning algorithms k-nn, mlp, and k-means clustering. In *2019 XXII Symposium on Image, Signal Processing and Artificial Vision (STSIVA)*, pages 1–5. IEEE, 2019.

[17] Guoxu Liu, Shuyi Mao, and Jae Ho Kim. A mature-tomato detection algorithm using machine learning and color analysis. *Sensors*, 19(9):2023, 2019.

[18] Kamalpreet Kaur and OP Gupta. A machine learning approach to determine maturity stages of tomatoes. *Oriental journal of computer science and technology*, 10(3):683–690, 2017.

[19] Ninja Begum and Manuj Kumar Hazarika. Maturity detection of tomatoes using transfer learning. *Measurement: Food*, 7:100038, 2022.

[20] Jun Liu, Jie Pi, and Liru Xia. A novel and high precision tomato maturity recognition algorithm based on multi-level deep residual network. *Multimedia Tools and Applications*, 79:9403–9417, 2020.

[21] Jasman Pardede, Benhard Sitohang, Saiful Akbar, and Masayu Leylia Khodra. Implementation of transfer learning using vgg16 on fruit ripeness detection. *Int. J. Intell. Syst. Appl*, 13(2):52–61, 2021.

[22] Hassan Shabani Mputu, Ahmed Abdel-Mawgood, Atsushi Shimada, and Mohammed S Sayed. Tomato quality classification based on transfer learning feature extraction and machine learning algorithm classifiers. *IEEE Access*, 2024.

[23] Quoc-Hung Phan, Van-Tung Nguyen, Chi-Hsiang Lien, The-Phong Duong, Max Ti-Kuang Hou, and Ngoc-Bich Le. Classification of tomato fruit using yolov5 and convolutional neural network models. *Plants*, 12(4):790, 2023.

[24] Jean Carlo Camacho and Manuel Eugenio Morocho-Cayamcela. Mask r-cnn and yolov8 comparison to perform tomato maturity recognition task. In *Conference on Information and Communication Technologies of Ecuador*, pages 382–396. Springer, 2023.

[25] Guoxu Liu, Joseph Christian Nouaze, Philippe Lyonel Touko Mbouembe, and Jae Ho Kim. Yolo-tomato: A robust algorithm for tomato detection based on yolov3. *Sensors*, 20(7):2145, 2020.

[26] Renzhi Li, Zijing Ji, Shikang Hu, Xiaodong Huang, Jiali Yang, and Wenfeng Li. Tomato maturity recognition model based on improved yolov5 in greenhouse. *Agronomy*, 13(2):603, 2023.

[27] Taiheng Zeng, Siyi Li, Qiming Song, Fenglin Zhong, and Xuan Wei. Lightweight tomato real-time detection method based on improved yolo and mobile deployment. *Computers and Electronics in Agriculture*, 205:107625, 2023.

[28] project cangp. Fruitdetectionv3 (3 class) dataset. https://universe.roboflow.com/project-cangp/fruitdetectionv3-3-class, apr 2023. visited on 2024-03-01.

[29] Juan Terven, Diana-Margarita Córdova-Esparza, and Julio-Alejandro Romero-González. A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas. *Machine Learning and Knowledge Extraction*, 5(4):1680–1716, 2023.

[30] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 21–37. Springer, 2016.

[31] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[32] Chien-Yao Wang, I-Hau Yeh, and Hong-Yuan Mark Liao. Yolov9: Learning what you want to learn using programmable gradient information. *arXiv preprint arXiv:2402.13616*, 2024.