

'Improved Yolov5-Based Tomato Disease Detection Combined With Yolov8

Mr. Milan Gohel¹, Dr. Hiren Kavathiya²

¹Department Of Computer Engineering, Atmiya University Atmiya University, India

²Department Of Computer Science Atmiya University Atmiya University, India

ABSTRACT

Targeting four major categories—healthy, blossom end rot, splitting rotation, as well as sun-scaled rotation—this study investigates the application of deep learning algorithms for precise tomato disease identification. Using a specially created tomato disease dataset, the study assesses the effectiveness of two light object identification algorithms, YOLOv5l and YOLOv8l. To establish a baseline for comparison, both models were first trained without data augmentation. Roboflow was used to apply a number of data augmentation techniques that effectively expanded and diversified the dataset in order to increase model generalization. These augmentations were designed to make the models more resilient to changes in lighting, plant orientation, and background conditions. After applying data augmentation, the models were retrained, and a thorough evaluation of their performance was conducted across all illness categories. The findings showed that both models' detection accuracy had significantly increased, confirming the importance of augmentation in enhancing model reliability. Overall, YOLOv8l consistently outperformed YOLOv5l by a small margin, especially when background images were excluded from the evaluation.

Keywords- tomato disease detection; Roboflow; YOLOv5; YOLOv8; accuracy.

I. INTRODUCTION

Tomato cultivation faces substantial threats from various diseases, which not only diminish the crop's visual quality and productivity but also create significant economic challenges for farmers worldwide. Computers can discover intricate patterns from massive datasets thanks to deep learning, which is modeled after the human brain and has emerged as a highly effective approach for detecting tomato diseases. These systems are able to recognize minute visual cues that signify the presence of a disease by training algorithms on large datasets that include thousands of photos of both healthy as well as diseased tomato plants, including both leaves as well as fruit. Once trained, such models can rapidly and accurately diagnose tomato diseases directly in the field. Early detection is crucial, allowing farmers to implement targeted treatments or isolate infected plants to prevent disease spread and safeguard their harvest. Beyond identifying problems, deep learning systems can also recommend appropriate interventions depending on the type and severity of the disease.

In the past, complex machine learning algorithms, which frequently needed substantial feature engineering and human extraction of picture features—were crucial for the early detection of fruit diseases. The introduction of Convolutional Neural Networks (CNNs), however, has revolutionized fruit disease detection by automating feature learning and improving both efficiency and accuracy. Studies [1–5] have shown that CNNs, modeled after the human visual system, excel at extracting patterns and features from images, laying the foundation for advanced object detection algorithms capable of automated tomato disease recognition. In general, object detection techniques fall into two categories: one-stage and two-stage. R-CNN, Fast R-CNN, Faster R-CNN, and other two-stage algorithms [6–11], create region proposals first, then categorize them. These models are highly accurate but computationally intensive, limiting their real-time applicability. One-stage methods, like SSD [6–7] and the YOLO series [8–18], perform detection in a single step, offering faster inference suitable for practical deployment. The

YOLO family has evolved through several iterations—including YOLO v3-5,7 and 8 [10,12,13, 14–16,17,18]—each iteration improving detection speed and accuracy.

In this work, we concentrate on YOLO-based tomato disease detection algorithms, namely YOLOv5l and YOLOv8l, which were trained on a meticulously selected custom dataset. By comparing their performance metrics, we aim to identify the most effective approach for accurately detecting tomato diseases. This work goes beyond mere identification; Through early, accurate, and automated disease diagnosis, it aims to improve overall tomato quality as well as increase crop yields. In the context of actual tomato disease control, the results from both YOLOv5l as well as YOLOv8l are examined and contrasted, providing insightful information about the relative advantages and disadvantages of each model.

YOLOv8 [18], each enhancing the effectiveness of object detection tasks.

This work examines the application of YOLO-based algorithms, namely YOLOv5l as well as YOLOv8l, trained on a carefully crafted unique dataset, for tomato disease diagnosis. By comparing these models' performance measures, the best strategy for this task could be found. By encouraging early and accurate diagnosis, this research aims to enhance tomato quality overall and boost crop yields, going beyond merely identifying problems. This is accomplished by combining state-of-the-art object detection methods with the effectiveness and accuracy of deep learning. Additionally, the outcomes from YOLOv5l as well as YOLOv8l are examined and contrasted, offering insightful information on their advantages and disadvantages in relation to tomato disease diagnosis.

II. LITERATURE REVIEW

In agriculture, a variety of artificial intelligence methods have been investigated [19]. For instance, CNN-based approaches have been applied to classify potato plant diseases into 15 categories [20]. In another study [21], To distinguish between healthy and diseased apple leaves, three prediction models were employed: CNN, SVM, and KNN. In a similar vein, five deep learning architectures were used to detect illness in banana leaves: VGG16, ResNet18, ResNet50, ResNet152, as well as InceptionV3 [22]. Tuta Absoluta pest-infected tomato leaves have also been identified using CNN segmentation techniques [23].

YOLO has been used in a number of investigations to identify plant diseases. For instance, a dataset of 4,000 digital photos from Bangladeshi tea gardens that represented five different kinds of leaf illnesses was used to train YOLOv7 [24]. In terms of parameters like detection precision, recall, precision, mean Average Precision (mAP), and F1-score, the model performed well. To improve harvesting and classification procedures, YOLOv8s has been used for automated tomato detection [18]. With a mAP of 93.4%, YOLOv8s' key components—DSConv, DPAG, and FEM—contributed to increased accuracy and efficiency.

Additionally, YOLOv5m has been used in conjunction with various architectures, including ResNet50, ResNet101, as well as EfficientNet-B0, to categorize vine tomato fruits in three groups: damaged, ripe, and immature [25]. YOLOv5m produced 100% prediction accuracy for ripe as well as immature tomatoes when combined with ResNet101, but EfficientNet-B0 produced 94% accuracy for damaged tomatoes. ResNet50, EfficientNet-B0, YOLOv5m, as well as ResNet101 networks had individual testing accuracies of 98%, 98%, 97%, and 97%, respectively. highlighting the effectiveness of integrating YOLO with deep learning architectures for precise plant and fruit disease detection.

III. MATERIALS

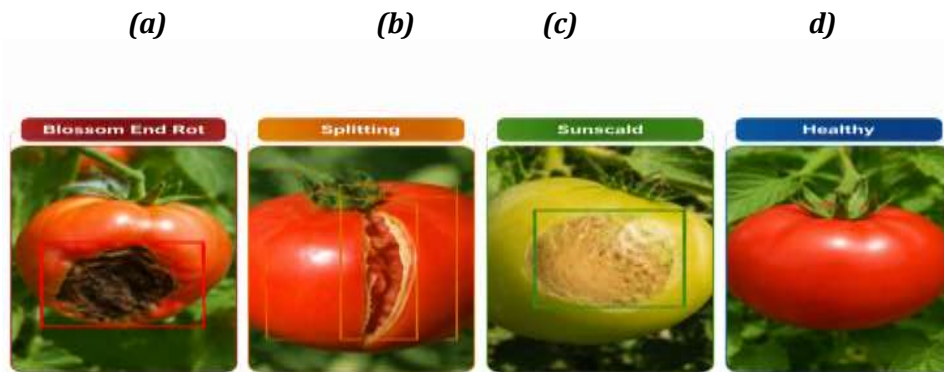
A. Data Collection

Images from two primary sources were used to create a bespoke dataset of tomato fruit disease detection: carefully selected high-quality images from online resources and includes pertinent publicly accessible datasets from earlier investigations. As shown in Figure 1, the gathered photos were divided into two main categories:

- **Healthy:** Pictures of tomatoes that show no symptoms of illness fall into this category.
- **Diseased:** Images of tomatoes with different disease signs make up this category.

About 1,600 photos of both healthy and ill tomatoes were collected for each category in order to ensure a fair representation across classes. The dataset preparation and organization were performed using the Roboflow platform, ensuring high-quality and standardized image inputs for model training.

Fig. 1. Tomato categories: (a) Blossom end rotation, (b) Splitting, (c) sun scaled rotation, (d) healthy.

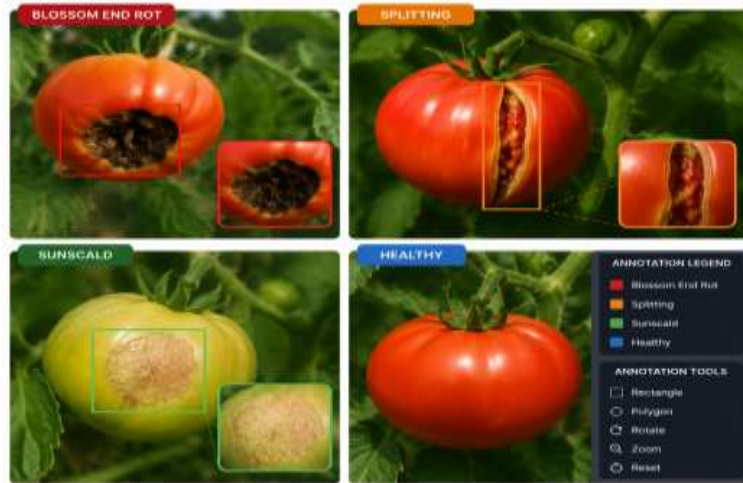


B. Image Annotation

The dataset preparation followed a structured procedure consisting of several key steps:

- **Image display:** The Roboflow platform loaded and displayed each image separately for processing.
- **Manual object annotation:** To precisely specify each tomato's position and size within the picture, a rectangular bounding box were manually painted around it. This careful annotation is critical for ensuring precise detection and classification during model training.
- **Label assignment:** Each tomato was given a unique label that reflected its state. "Healthy," "Blossom end rot rotation," "Splitting rotation," and "Sun-scaled rotation" were the labels used to indicate the existence or nonexistence of disease as well as its particular type.
- **Annotation visualization:** The annotated dataset is depicted in Figure 2, which also shows the locations of the bounding boxes surrounding each tomato and the distribution of labels.

Fig. 2. Annotated image.



C. Data Resize

All images in the collected dataset were resized within Roboflow, standardizing them to a consistent 640 x 640 pixel resolution in order to guarantee model training consistency.

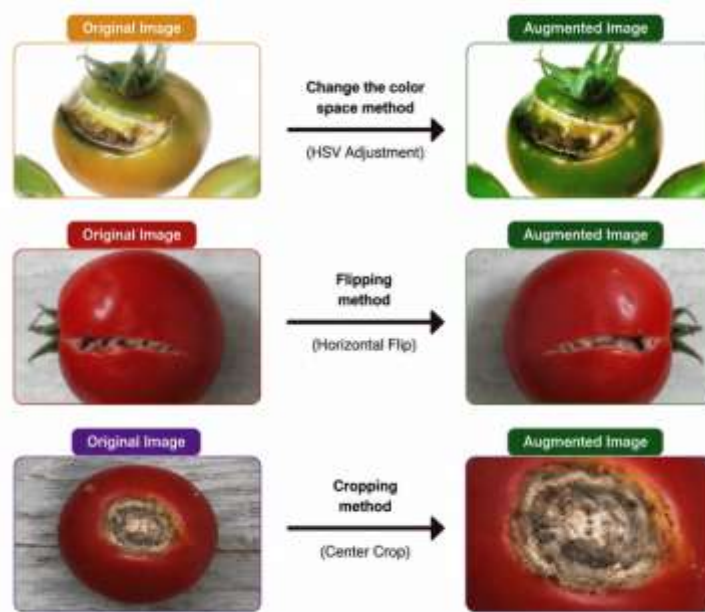
D. Data Preprocessing

To enable efficient testing, validation, and training for the tomato fruit diagnosis models, the collected dataset was methodically divided into three distinct subsets.

- The Training set (70%): This subset serves as the main source of data for the model's training, enabling it to identify patterns and extract crucial characteristics necessary for distinguishing between tomatoes that are healthy and those that are not.
- Validation set (20%): In order to adjust hyperparameters and avoid overfitting, this subset is used to track the model's performance during training. It guarantees the model's successful generalization to fresh, untested data.
- Test set (10%): Kept completely separate from the training process, The model's final assessment is conducted using this subset. It gives an objective assessment of the model's performance and predicts how well it will work in practical applications.

E. Data Augmentation

Several data augmentation approaches were used to improve the dataset after the photos were meticulously annotated in Roboflow. This process not only increases the overall volume of data but also introduces diversity in image content, helping the model become more robust. Data augmentation also lowers the chance of overfitting, a common problem when training models on small datasets, by simulating changes in lighting, orientation, and background.

Fig. 3. Data augmentation example.

The many data augmentation methods used, such as flipping, cropping, rotation, as well as color space changes, are shown in Figure 3. By creating additional images over the original dataset, these techniques broaden the dataset's diversity and provide the model more scenarios for learning from. Following resizing and augmentation, the dataset expanded to over 3,000 images. As illustrated in Figure 4, the improved dataset [26] was then split into three separate subsets: training, validation, and testing, ensuring effective model development and evaluation.

Fig. 4. Dataset split after resizing and data augmentation.

IV. PROPOSED METHODS

The curated and augmented dataset was accessed and managed through Roboflow, while model training was performed using Google Colab can take use of its complimentary access to powerful GPUs. A 12GB NVIDIA Tesla T4 GPU was used for all training and testing, allowing the models to be efficiently trained over approximately two hours for 50 epochs.

Separate models got trained to each approach in order to evaluate the dataset's suitability for YOLOv5l and YOLOv8l. This comparative method made it possible to thoroughly assess each model's accuracy and efficacy on the unique dataset. Although YOLOv5 and YOLOv8 are both very powerful object detection frameworks, their underlying designs differ, giving them different advantages and disadvantages. Table I summarizes the fundamental distinctions between the models, while Table II highlights the similarities in the proposed YOLO-based methods.

TABLE I.YOLOV5 AND YOLOV8 DIFFERENCES

Architecture	YOLOv8l	YOLOv5l
Backbone	For feature extraction, it makes use of EfficientNet-Lite, which offers the best possible compromise between computational efficiency and accuracy.	It incorporates multiple backbone options, including CSPDarknet53 and EfficientNet, enabling customization to achieve the preferred balance between speed and accuracy..
Neck	It employs a straightforward Path Aggregation Network (PANet) for feature fusion, emphasizing computational efficiency.	It utilizes a more advanced combination of the Spatial Attention Module (SAM) and PANet to generate richer and more informative feature representations.
Head	It simplifies the training process by using an anchor-free method that predicts bounding box centers and sizes directly.	It uses an anchor-based approach, which relies on predefined anchors for bounding box prediction and can achieve higher accuracy when anchors are carefully designed.
Overall	It focuses on simplicity and efficiency, delivering strong performance while reducing training complexity and enabling potentially faster inference.	It emphasizes accuracy and flexibility by providing multiple backbone and neck options for customization, However, this could lead to slower inference and need more training resources.

TABLE II.YOLOV5 AND YOLOV8 SIMILARITY

Architecture	Make use of YOLOv5l and YOLOv8l
Backbone	Backbone of CSPDarknet53
Anchor Boxes	to increase the precision of object detection.

Non-Maximum Suppression (NMS)	to prevent the same object from being detected more than once.
Post-processing	to increase object detecting accuracy.
Optimizer	Adam optimizer for model training.
Activation Function	Their architecture has a mish activation function.

YOLOv8l's lightweight architecture and quicker inference capabilities make it ideal for real-time applications whereby speed is crucial. In contrast, YOLOv5l may be more suitable for scenarios that prioritize high accuracy and have sufficient computational resources, as it offers flexible backbone options and can achieve superior accuracy, particularly when carefully designed anchors are employed.

V. EXPERIMENTAL RESULT

A. Results

YOLOv5l and YOLOv8l were first trained for tomato disease diagnosis without the need of data augmentation. The performance metrics for both models, such as recall, precision, as well as mAP, are shown in Table III. As observed, YOLOv8l demonstrated higher accuracy than YOLOv5l, although the overall performance was below expectations. To address this, data augmentation was applied to enrich the dataset with diverse image variations. After retraining on the augmented dataset, both YOLOv5l and YOLOv8l achieved significantly improved results, as shown in Table IV. This substantial performance boost highlights the effectiveness of data augmentation and establishes this approach as the preferred method for accurate tomato disease detection in the given context.

Fig. 5. YOLOv8 architecture.

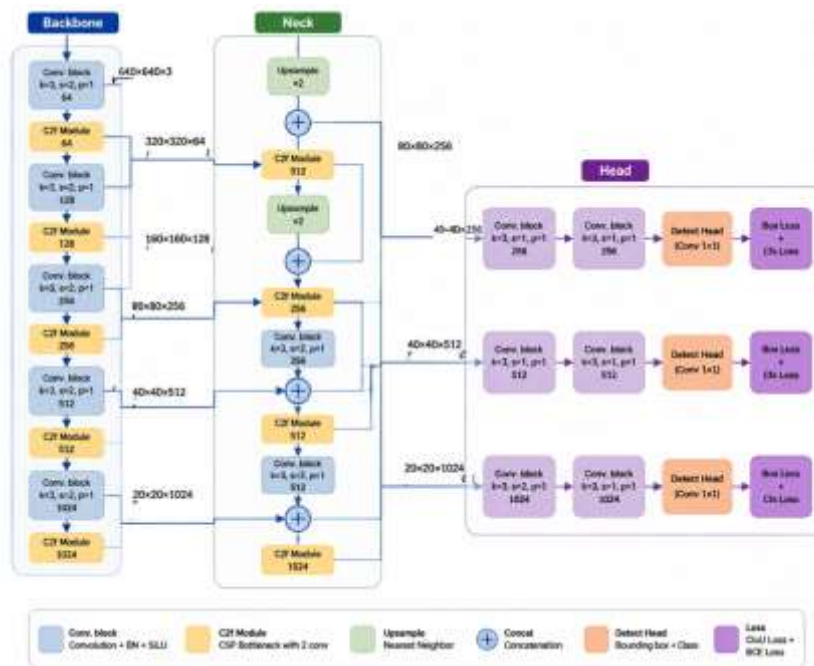


Fig. 6. YOLOv5 architecture.

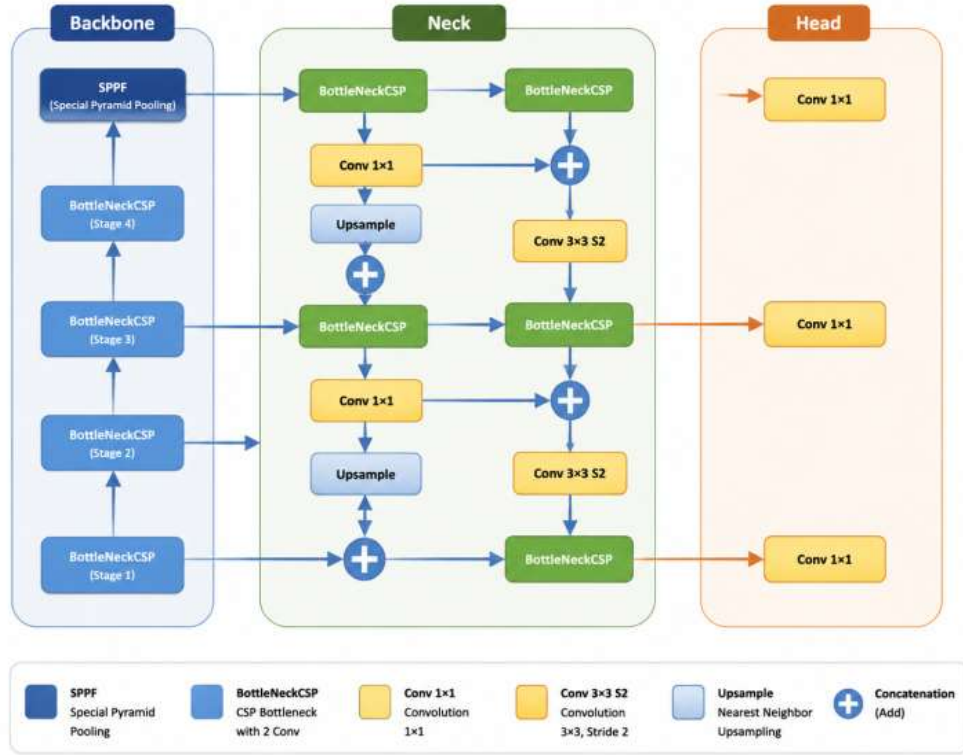


TABLE III. YOLOV5 AND YOLOV8 PERFORMANCE WITHOUT DATA AUGMENTATION

	Precision	Recall	mAP50	mAP50-95
YOLOv5l	69.80%	62.30%	67.90%	49.50%
YOLOv8l	79.20%	70.10%	78.90%	55.00%

TABLE IV. YOLOV5 AND YOLOV8 PERFORMANCE WITH DATA AUGMENTATION

	Precision	Recall	mAP50	mAP50-95
YOLOv5l	89.30%	74.40%	85.20%	58.50%
YOLOv8l	91.60%	83.10%	88.50%	60.00%

B. Evaluation Metrics

The confusion matrices of models trained on the augmented and non-augmented datasets are shown in Figures 7 and 8, respectively, and provide some noteworthy insights:

- **Increased True Positives (TP):** Both YOLOv5l and YOLOv8l exhibit higher TP values for classes including blossom end rot rotation, splitting rotation, as well as sun-scaled rotation following data augmentation, indicating an improved capacity to accurately diagnose these diseases.
- **Reduced False Positives (FP):** Compared to YOLOv5l, YOLOv8l shows lower FP in the majority of classes, suggesting increased accuracy in disease categorization after augmentation.

These results imply that YOLOv8l performs better overall, with improved TP and FP rates leading to greater disease detection accuracy. Model accuracy was computed

using the confusion matrices both prior to and after data augmentation in order to quantitatively assess this improvement. Because accuracy offers a thorough evaluation of the models' performance across all illness classes, it was chosen as the metric. Under these various conditions, accuracy was calculated using equation (1):

After examining the confusion matrices, a significant quantity of backdrop pictures—containing no relevant objects—were observed. These images have the potential to distort accuracy measurements. While a few background images may have negligible effect, a larger proportion can lead to underestimating the model's true performance. To account for this, accuracy was calculated in two ways:

- **Excluding background images:** By excluding pictures that do not have any illness, this method makes it easier to see how well the model can recognize real cases of illness.
- **Including background images:** This approach provides a more thorough evaluation of the overall performance of the model by taking into account all photos, including those without illness.

Table V presents the accuracy values obtained using both approaches. YOLOv8l consistently outperforms YOLOv5l, achieving accuracy improvements of 3.67% and 9.25% before and after data augmentation, respectively. Figure 9 highlights the substantial impact of augmented data on disease detection accuracy. YOLOv5l initially achieves 86.6% accuracy on the non-augmented dataset, which rises significantly to 95.85% following data augmentation.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FN} + \text{FP}} \quad (1)$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FN} + \text{FP}} \quad (1)$$

After using data augmentation, YOLOv8's accuracy increases from 94.3% to 97.9%. These significant improvements highlight how expanding the dataset can improve model performance.

where FN stands for False Negatives and TN for True Negatives.

Fig. 7. Prior to augmentation, confusion matrices for the custom dataset



(a) YOLOv8, (b) YOLOv5.

Fig. 8. Confusion matrices following augmentation on customary dataset (a) YOLOv8, (b) YOLOv5.

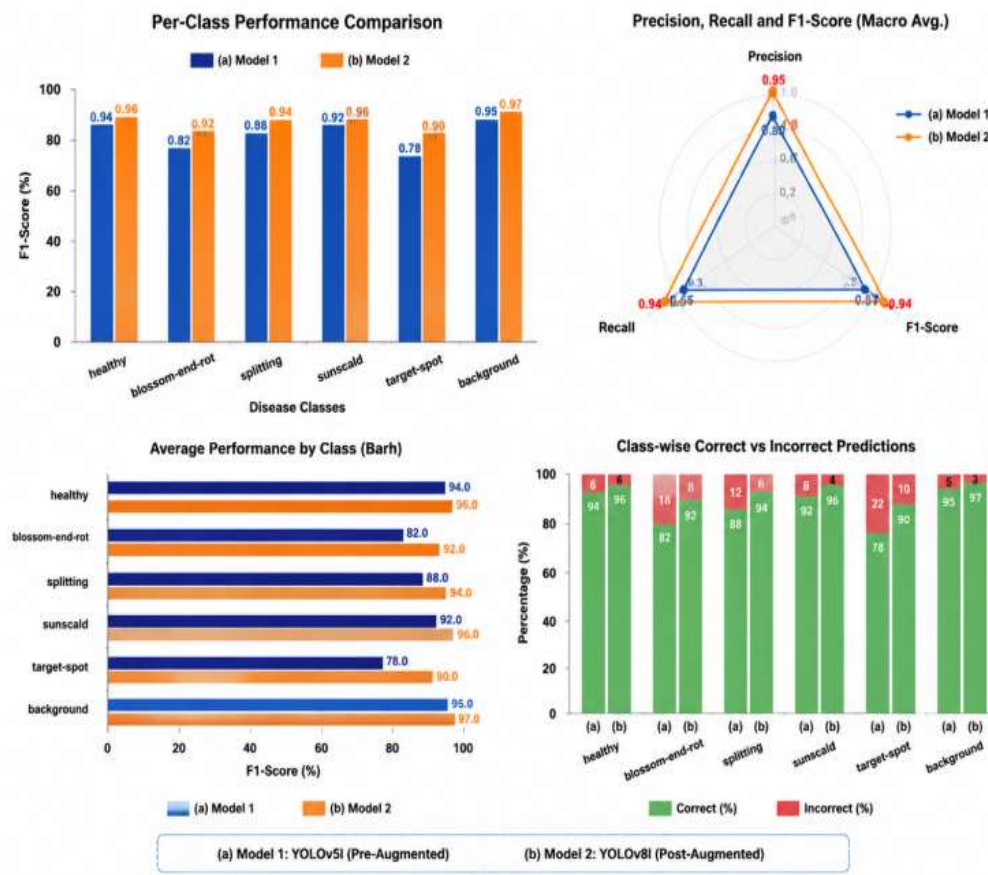
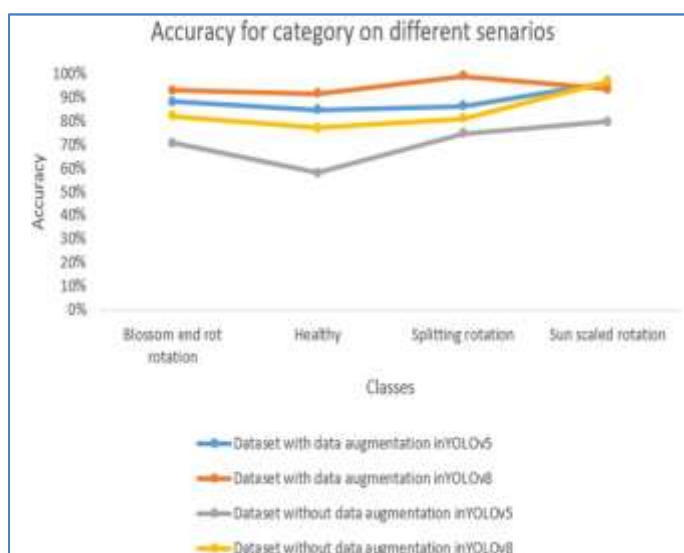


TABLE V. ACCURACY PERFORMANCE

	Blossom end rot rotation	Healthy	Splitting rotation	Sun scaled rotation
Dataset before augmentation YOLOv5	70.75%	58.06%	74.74%	79.82%
Dataset before augmentation YOLOv8	82.14%	77.14%	81.08%	96.97%
Dataset after augmentation YOLOv5	88.04%	84.78%	86.32%	96.00%
Dataset after augmentation YOLOv8	92.78%	91.57%	98.88%	93.41%

Fig. 9. Accuracy for every category in various contexts.



VI. CONCLUSION

This work uses deep learning methods in conjunction with data augmentation to address the problem of tomato disease diagnosis. A custom dataset was carefully designed and compiled in Roboflow, consisting of more than 1,600 photos that were automatically split into training as well as validation sets. This dataset was used to train and assess the YOLOv5l and YOLOv8l models, which at first showed encouraging accuracies of 86.6% as well as 94.3%, respectively. A number of data augmentation methods, including flipping, cropping, rotation, as well as color space changes, were used to increase the dataset's size and diversity in order to further improve performance. Significant improvements were obtained by retraining the models across the augmented dataset: YOLOv5l's accuracy increased from 86.6% to 95.85%, while YOLOv8l rose from 94.3% to an impressive 97.9%.

Analysis of the confusion matrices revealed a reduction in false positives, demonstrating increased accuracy in the classification of diseases. YOLOv8l performed better overall in a number of categories, including true positives, false positives, and overall accuracy, particularly when background images were excluded from the evaluation. These findings demonstrate how well data augmentation can improve deep learning models' capacity to identify tomato diseases. The algorithms were able to identify a variety of tomato illnesses with excellent accuracy and reliability by increasing the dataset's diversity and reducing background noise.

Future research will aim to further refine this detection system by improving the handling of background images and enhancing the YOLO architecture. Efforts will include generating realistic disease-free backgrounds, targeted augmentations to reduce disease-related biases, and by using multiscale feature fusion and attention mechanisms to improve disease-specific focus and identify important visual patterns.

REFERENCES

1. Y. Gulzar, "Fruit Image Classification Model Based on MobileNetV2 with Deep Transfer Learning Technique," *Sustainability*, vol. 15, no. 3, Jan. 2023, Art. no. 1906, <https://doi.org/10.3390/su15031906>.
2. M. Afonso et al., "Tomato Fruit Detection and Counting in Greenhouses Using Deep Learning," *Frontiers in Plant Science*, vol. 11, Nov. 2020, <https://doi.org/10.3389/fpls.2020.571299>.
3. G. Moreira, S. A. Magalhães, T. Pinho, F. N. dos Santos, and M. Cunha, "Benchmark of

- Deep Learning and a Proposed HSV Colour Space Models for the Detection and Classification of Greenhouse Tomato," *Agronomy*, vol. 12, no. 2, Feb. 2022, Art. no. 356, <https://doi.org/10.3390/agronomy12020356>.
4. Y. Mu, T.-S. Chen, S. Ninomiya, and W. Guo, "Intact Detection of Highly Occluded Immature Tomatoes on Plants Using Deep Learning Techniques," *Sensors*, vol. 20, no. 10, Jan. 2020, Art. no. 2984, <https://doi.org/10.3390/s20102984>.
5. S. A. Magalhães et al., "Evaluating the Single-Shot MultiBox Detector and YOLO Deep Learning Models for the Detection of Tomatoes in a Greenhouse," *Sensors*, vol. 21, no. 10, Jan. 2021, Art. no. 3569, <https://doi.org/10.3390/s21103569>.
6. F. Zeng, Y. Liu, Y. Ye, J. Zhou, and X. Liu, "A detection method of Edge Coherent Mode based on improved SSD," *Fusion Engineering and Design*, vol. 179, Jun. 2022, Art. no. 113141, <https://doi.org/10.1016/j.fusengdes.2022.113141>.
7. H. Peng et al., "General improved SSD model for picking object recognition of multiple fruits in natural environment," *Transactions of the Chinese Society of Agricultural Engineering*, vol. 34, no. 16, pp. 155–162, 2018.
8. P. Jiang, D. Ergu, F. Liu, Y. Cai, and B. Ma, "A Review of Yolo Algorithm Developments," *Procedia Computer Science*, vol. 199, pp. 1066–1073, Jan. 2022, <https://doi.org/10.1016/j.procs.2022.01.135>.
9. J. Wu, Z. Kuang, L. Wang, W. Zhang, and G. Wu, "Context-Aware RCNN: A Baseline for Action Detection in Videos," in *Computer Vision – ECCV 2020*, Glasgow, UK, 2020, pp. 440–456, https://doi.org/10.1007/978-3-030-58595-2_27.
10. G. Liu, J. C. Nouaze, P. L. Touko Mbouembe, and J. H. Kim, "YOLO-Tomato: A Robust Algorithm for Tomato Detection Based on YOLOv3," *Sensors*, vol. 20, no. 7, Jan. 2020, Art. no. 2145, <https://doi.org/10.3390/s20072145>.
11. B. Hu and J. Wang, "Detection of PCB Surface Defects With Improved Faster-RCNN and Feature Pyramid Network," *IEEE Access*, vol. 8, pp. 108335–108345, 2020, <https://doi.org/10.1109/ACCESS.2020.3001349>.
12. Y. Yang, J. Li, J. Nie, S. Yang, and J. Tang, "Cotton Stubble Detection Based on Improved YOLOv3," *Agronomy*, vol. 13, no. 5, May 2023, Art. no. 1271, <https://doi.org/10.3390/agronomy13051271>.
13. R. Gai, N. Chen, and H. Yuan, "A detection algorithm for cherry fruits based on the improved YOLO-v4 model," *Neural Computing and Applications*, vol. 35, no. 19, pp. 13895–13906, Jul. 2023, <https://doi.org/10.1007/s00521-021-06029-z>.
14. T. Saidani, R. Ghodhbani, A. Alhomoud, A. Alshammari, H. Zayani, and M. B. Ammar, "Hardware Acceleration for Object Detection using YOLOv5 Deep Learning Algorithm on Xilinx Zynq FPGA Platform," *Engineering, Technology & Applied Science Research*, vol. 14, no. 1, pp. 13066–13071, Feb. 2024, <https://doi.org/10.48084/etasr.6761>.
15. R. Rajamohan and B. C. Latha, "An Optimized YOLO v5 Model for Tomato Leaf Disease Classification with Field Dataset," *Engineering, Technology & Applied Science Research*, vol. 13, no. 6, pp. 12033–12038, Dec. 2023, <https://doi.org/10.48084/etasr.6377>.
16. T. Saidani, "Deep Learning Approach: YOLOv5-based Custom Object Detection," *Engineering, Technology & Applied Science Research*, vol. 13, no. 6, pp. 12158–12163, Dec. 2023, <https://doi.org/10.48084/etasr.6397>.
17. J. Zhou, Y. Zhang, and J. Wang, "RDE-YOLOv7: An Improved Model Based on YOLOv7 for Better Performance in Detecting Dragon Fruits," *Agronomy*, vol. 13, no. 4, Apr. 2023, Art. no. 1042, <https://doi.org/10.3390/agronomy13041042>.
18. N. C. Eli-Chukwu, "Applications of Artificial Intelligence in Agriculture: A Review," *Engineering, Technology & Applied Science Research*, vol. 9, no. 4, pp. 4377–4383, Aug. 2019, <https://doi.org/10.48084/etasr.2756>.

20. A. Abbas, U. Maqsood, S. U. Rehman, K. Mahmood, T. AlSaedi, and
21. M. Kundi, "An Artificial Intelligence Framework for Disease Detection in Potato Plants," *Engineering, Technology & Applied Science Research*, vol. 14, no. 1, pp. 12628–12635, Feb. 2024, <https://doi.org/10.48084/etasr.6456>.
22. S. Alqethami, B. Almtanni, W. Alzhrani, and M. Alghamdi, "Disease Detection in Apple Leaves Using Image Processing Techniques," *Engineering, Technology & Applied Science Research*, vol. 12, no. 2,
23. pp. 8335–8341, Apr. 2022, <https://doi.org/10.48084/etasr.4721>.
24. S. L. Sanga, D. Machuve, and K. Jomanga, "Mobile-based Deep Learning Models for Banana Disease Detection," *Engineering, Technology & Applied Science Research*, vol. 10, no. 3, pp. 5674–5677, Jun. 2020, <https://doi.org/10.48084/etasr.3452>.
25. L. Loyani and D. Machuve, "A Deep Learning-based Mobile Application for Segmenting Tuta Absoluta's Damage on Tomato Plants," *Engineering, Technology & Applied Science Research*, vol. 11, no. 5, pp. 7730–7737, Oct. 2021, <https://doi.org/10.48084/etasr.4355>.
26. M. J. A. Soeb et al., "Tea leaf disease detection and identification based on YOLOv7 (YOLO-T)," *Scientific Reports*, vol. 13, no. 1, Apr. 2023, Art. no. 6078, <https://doi.org/10.1038/s41598-023-33270-4>.
27. G. Yang, J. Wang, Z. Nie, H. Yang, and S. Yu, "A Lightweight YOLOv8 Tomato Detection Algorithm Combining Feature Enhancement and Attention," *Agronomy*, vol. 13, no. 7, Jul. 2023, Art. no. 1824, <https://doi.org/10.3390/agronomy13071824>.
28. Q. H. Phan, V. T. Nguyen, C. H. Lien, T. P. Duong, M. T. K. Hou, and
29. N. B. Le, "Classification of Tomato Fruit Using YOLOv5 and Convolutional Neural Network Models," *Plants*, vol. 12, no. 4, Jan. 2023, Art. no. 790, <https://doi.org/10.3390/plants12040790>.
30. "deseaseTomato Image Dataset." Feb. 17, 2024, [Online]. Available: <https://universe.roboflow.com/datasetsnbu/deseasetomato/dataset/8>.