Improved Tomato Disease Detection with YOLOv5 and YOLOv8

Rabie Ahmed
Department of Computer Science, Faculty of Computing and Information Technology, Northern Border University, Saudi Arabia | Department of Mathematics and Computer Science, Faculty of Science, Beni-Suef University, Egypt
rabie.ahmed@nbu.edu.sa (corresponding author)

Eman H. Abd-Elkawy
Department of Computer Science, Faculty of Computing and Information Technology, Northern Border University, Saudi Arabia | Department of Mathematics and Computer Science, Faculty of Science, Beni-Suef University, Egypt
eman.hassan@nbu.edu.sa

Received: 14 March 2024 | Revised: 22 March 2024 | Accepted: 24 March 2024

 DOI: https://doi.org/10.48084/etasr.7262

ABSTRACT

This study delves into the application of deep learning for precise tomato disease detection, focusing on four crucial categories: healthy, blossom end rot, splitting rotation, and sun-scaled rotation. The performance of two lightweight object detection models, namely YOLOv5l and YOLOv8l, was compared on a custom tomato disease dataset. Initially, both models were trained without data augmentation to establish a baseline. Subsequently, diverse data augmentation techniques were obtained from Roboflow to significantly expand and enrich the dataset content. These techniques aimed to enhance the models' robustness to variations in lighting, pose, and background conditions. Following data augmentation, the YOLOv5l and YOLOv8l models were re-trained and their performance across all disease categories was meticulously analyzed. After data augmentation, a significant improvement in accuracy was observed for both models, highlighting its effectiveness in bolstering the models' ability to accurately detect tomato diseases. YOLOv8l consistently achieved slightly higher accuracy compared to YOLOv5l, particularly when excluding background images from the evaluation.

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

I. INTRODUCTION

Tomato production is threatened by diseases. These damaging factors not only affect their appearance and productivity, but also pose significant economic difficulties for producers around the world. Deep Learning techniques, inspired by the human brain, enable computers to learn and identify patterns from large datasets and can be used in tomato disease detection. Deep learning algorithms are trained on a large dataset consisting of thousands of images of healthy and diseased tomato plants, including leaves and fruits, to detect tomato diseases. Algorithms learn to identify minor visual cues that indicate the presence of a disease by studying these images. Once trained, these algorithms are utilized to diagnose tomato diseases with speed and precision in the field. The early detection of a disease is essential. Farmers can quickly respond by employing specific treatments or isolating affected plants to reduce the spread of disease and protect their valuable crops. Not only does deep learning involve recognizing issues, but also providing suggestions for appropriate courses of action depending on the disease's nature and severity.

The early recognition of fruit diseases relies heavily on intricate machine learning algorithms, often requiring complex feature engineering and manual extraction processes. However, with the emergence of Convolutional Neural Networks (CNNs), there has been a revolutionary shift in fruit recognition techniques. In [1-5], CNNs, designed to mimic the visual processing of the human brain, were proven to be highly adept at recognizing patterns and features within images, leading to significant improvements in both efficiency and precision. These advances have paved the way for the development of more sophisticated object detection algorithms, which are integral to automating the detection of tomato diseases. Among these algorithms, two-stage approaches such as R-CNN involve regional proposal and classification stages, and one-stage approaches, such as SSD [6-7] and YOLO [8], perform detection in a single step. The former includes algorithms, like R-CNN [9], Fast-RCNN [10], and Faster-RCNN [11], which exhibit high robustness with low error rates but require a long run time, making them unsuitable for real-time production. YOLO has undergone several iterations, namely YOLOv3 [10, 12], YOLOv4 [13], YOLOv5 [14-16], YOLOv7 [17], and

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

This study investigates the use of YOLO-based algorithms for tomato disease detection, specifically YOLOv5l and YOLOv8l, trained on a meticulously designed custom dataset. The most effective approach to this task can be identified by comparing the performance metrics of these models. This study goes beyond simply detecting diseases, as it strives to improve overall tomato quality and increase crop yields by promoting early and accurate diagnoses. This is achieved by harnessing the efficiency and precision of deep learning combined with cutting-edge object detection techniques. Furthermore, the results obtained from both YOLOv5l and YOLOv8l are analyzed and compared, providing valuable insight into their strengths and weaknesses in the context of tomato disease detection.

II. LITERATURE REVIEW

Many artificial intelligence techniques have been studied in agriculture [19]. In [20], CNN methods were used to categorize potato plant diseases into 15 classes. In [21], three prediction models (CNN, SVM, and KNN) were implemented to classify apple leaves as healthy or diseased. In [22], five deep-learning architectures, namely Vgg16, Resnet18, Resnet50, Resnet152, and InceptionV3, were employed to detect diseases in banana leaves. In [23], a CNN segmentation model was utilized to detect tomato leaves infected by the Tuta Absoluta pest. Many studies have engaged various YOLO versions for the detection of plant diseases. In [24], YOLOv7 was trained on a dataset of 4000 digital photos depicting five types of leaf disease obtained from tea gardens in Bangladesh. The performance of the YOLOv7 model was evaluated based on statistical measures including detection accuracy, precision, recall, mean Average Precision (mAP), and F1-score, achieving high values in all metrics. In [18], YOLOv8s was put into service for automated tomato detection to improve tomato harvesting and classification automation. Its main elements consist of DSConv, DPAG, and FEM to ameliorate accuracy and efficiency, achieving a mAP of 93.4%. In [25], YOLOv5m was combined with ResNet50, ResNet-101, and EfficientNet-B0 to classify vine tomato fruits into three categories: ripe, immature, and damaged. When combining YOLO5m with ResNet-101, the prediction accuracy for ripe and immature tomatoes was 100%. When using YOLOv5m with the Efficient-B0 architecture, the accuracy of predicting damaged tomatoes was 94%. ResNet-50, EfficientNet-B0, YOLOv5m, and ResNet-101 networks exhibited testing accuracies of 98, 98, 97, and 97%, respectively.

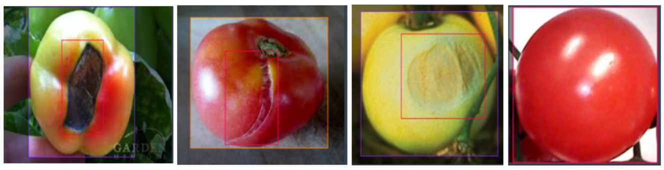
III. MATERIALS

A. Data Collection

A tomato fruit disease detection dataset was created by collecting images from two principal sources: online resources, carefully selecting high-quality tomato images from the internet, and existing datasets, leveraging relevant and publicly available datasets from previous research studies. Thus, the images were classified into two main categories, as portrayed in Figure 1:

- Healthy: This category contains images of tomatoes that do not show signs of disease.
- Diseased: This category includes tomato images that demonstrate various disease symptoms.

To ensure a balanced representation, approximately 1600 images were collected for each category, including both healthy and diseased tomatoes. This process was carried out in Roboflow.



(a) (b) (c) (d)

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

B. Image Annotation

This procedure involved the following steps:

- Image presentation: Each image was individually displayed within Roboflow.
- Manual object labeling: A rectangular shape was drawn manually around the tomato to precisely capture its location and size within an image. This precise labeling ensures accurate detection and classification later.
- Specific label definitions: Each tomato received different labels reflecting its condition. These labels included "Healthy," "Blossom end rot rotation," "Splitting rotation," and "Sun-scaled rotation," indicating both the presence or absence of disease and its specific type.
- Visualization of annotations: Figure 2 provides a visual representation of the annotated data, offering information on the distribution of different labels and the location of the bounding boxes around each tomato.

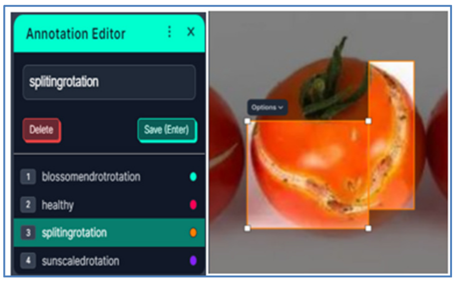


Fig. 2. Annotated image.

C. Data Resize

Image resizing was performed in Roboflow, changing all images in the collected dataset from various sizes to stretch 640×640.

D. Data Preprocessing

The collected dataset was carefully divided into three distinct subsets for effective training and evaluation of tomato fruit disease detection models.

- The training set (70%) serves as the backbone for model training. During this process, the model learns to recognize patterns and identify key features crucial for distinguishing between healthy and diseased fruit.
- The validation set (20%) plays a vital role in validating the model's performance during training. It is used to fine-tune hyperparameters and avoid overfitting, ensuring the model generalizes well to unseen data.
- The test set (10%) remains untouched throughout training and is utilized for the final evaluation of the model's performance on unseen data. This unbiased evaluation offers a reliable indicator of how well the model would perform in real-world scenarios.

E. Data Augmentation

After meticulously labeling the images in Roboflow, data-augmentation techniques were applied. This approach aims to enrich the dataset by increasing its volume and diversifying its content. Data augmentation helps mitigate the risk of overfitting, a common challenge when training models with limited data.

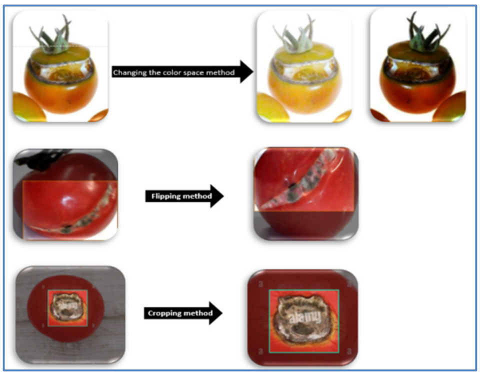


Fig. 3. Data augmentation example.

Figure 3 depicts various data augmentation methods deployed, including flipping, cropping, rotating, and changing the color space. These methods generate additional images employing the initial dataset, increasing its variability and offering the model a wider array of situations to train on. The amount of images increased to more than 3000 photos after resizing and data augmentation. The collected dataset [26], was separated into three separate subsets for training, evaluation, and testing, as observed in Figure 4.

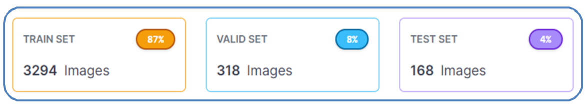


Fig. 4. Dataset split after resizing and data augmentation.

IV. PROPOSED METHODS

Using Roboflow, a dataset was generated to access and utilize the curated and augmented images. Google Colab was deployed to accelerate training and achieve superior results, as it provides free access to powerful GPUs. All training and testing procedures were carried out on a 12GB NVIDIA Tesla T4 GPU. This cloud-based environment helped to efficiently train the models in roughly two hours with 50 epochs.

To evaluate the dataset's suitability for both the YOLOv5l and YOLOv8l algorithms, separate models were trained engaging each method. This comparative approach was adopted to evaluate the precision and effectiveness of each algorithm in this specific dataset. Both YOLOv8 and YOLOv5 are powerful object detection models but differ in their underlying architecture, leading to unique strengths and weaknesses. Table I presents an outline of their fundamental distinctions, and Table II displays the similarity of the proposed methods in YOLO.

TABLE I. DIFFERENCE BETWEEN YOLOV5 AND YOLOV8

Architecture	**YOLOv8l**	**YOLOv5l**
Backbone	Relies on EfficientNet-Lite for feature extraction, offering a balance between accuracy and speed.	Employs various backbones, such as CSPDarknet53 and EfficientNet, allowing for customization based on desired speed and accuracy trade-offs.
Neck	Uses a simple Path Aggregation Network (PANet) for feature fusion, focusing on efficiency.	Leverages a more complex Spatial Attention Module (SAM) and Path Aggregation Network (PANet) combination for richer feature representations.
Head	Employs an anchor-free approach, directly predicting bounding box centers and sizes, which reduces training complexity.	Utilizes an anchor-based approach, requiring pre-defined anchors for bounding box prediction, offering potentially higher accuracy with careful anchor design.
Overall	Aims for simplicity and efficiency, achieving good performance with reduced training complexity and potentially faster inference speeds.	Prioritizes accuracy and flexibility, offering various backbone and neck options for customization but potentially requiring more training resources and slower inference.

TABLE II. SIMILARITY BETWEEN YOLOV5 AND YOLOV8

Architecture	**Use YOLOv5l and YOLOv8l**
Backbone	CSPDarknet53 backbone
Anchor Boxes	To improve object detection accuracy.
Non-Maximum Suppression (NMS)	To suppress multiple detections of the same object.
Post-processing	To improve the accuracy of object detection.
Optimizer	Adam optimizer for training the model.
Activation Function	Mish activation function in their architecture.

YOLOv8l could be a better fit for real-time applications where speed is crucial because of its lighter architecture and potentially faster inference times. YOLOv5l might be preferred for tasks that require high accuracy and are not resource-constrained due to its flexible backbone choices and potentially higher accuracy, especially with well-designed anchors.

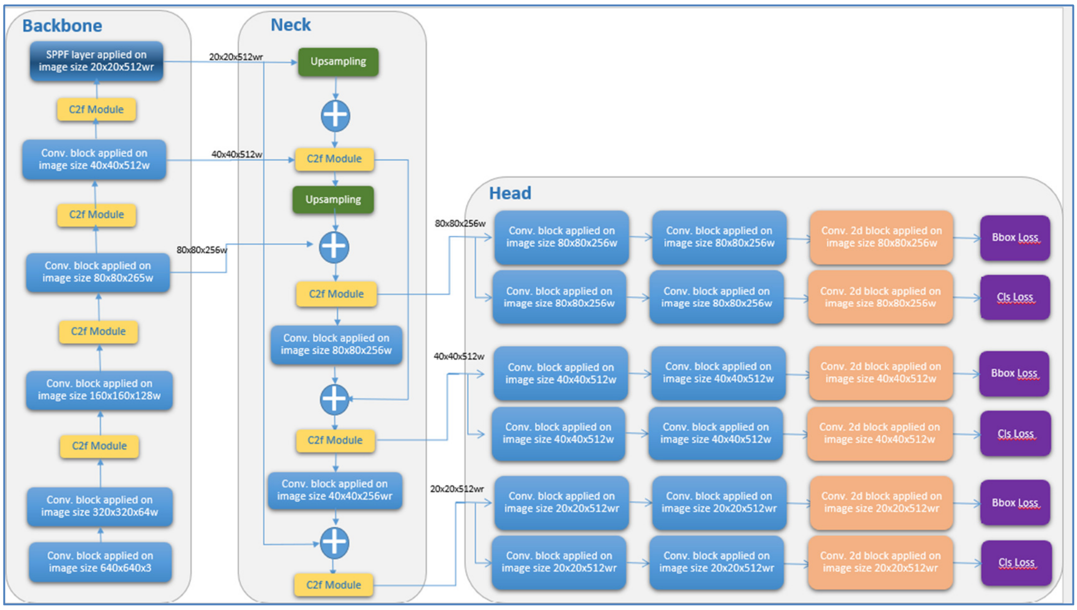


Fig. 5. YOLOv8 architecture.

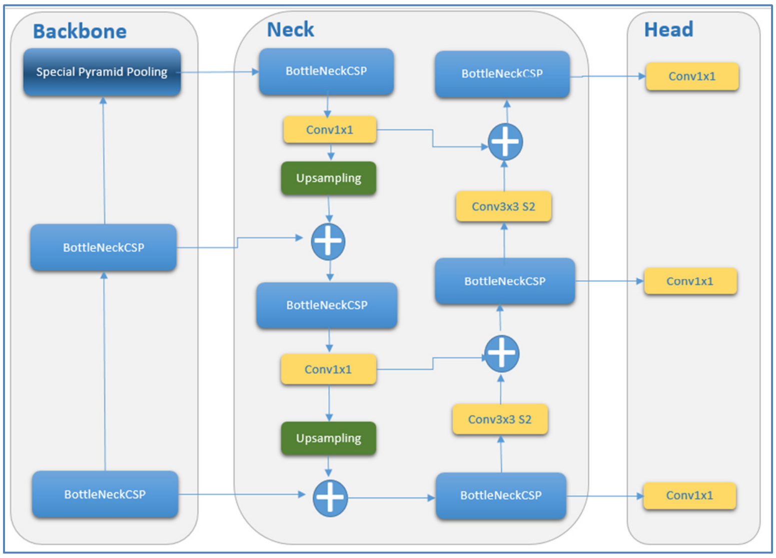


Fig. 6. YOLOv5 architecture.

V. EXPERIMENTAL RESULT ANALYSIS

A. Results

In the beginning, this study trained both YOLOv5l and YOLOv8l for disease detection but without data augmentation. Table III illustrates the performance metrics (recall, precision, and mAP) for both models. As noticed in Table III, YOLOv8l was significantly more accurate, but these values were lower than expected. Therefore, data augmentation was implemented to improve overall performance. Following data augmentation, the proposed dataset equipped with diverse images fueled both YOLOv5l and YOLOv8l models, leading to outstanding results, as spotted in Table IV. This remarkable performance enhancement marks this approach as the preferred method for accurate tomato disease detection in this specific context.

TABLE III. PERFORMANCE USING YOLOV5 AND YOLOV8 WITHOUT DATA AUGMENTATION

	Precision	Recall	mAP50	mAP50-95
YOLOv5l	69.8%	62.3%	67.9%	49.5%
YOLOv8l	79.2%	70.1%	78.9%	55%

TABLE IV. PERFORMANCE USING YOLOV5 AND YOLOV8 WITH DATA AUGMENTATION

	Precision	Recall	mAP50	mAP50-95
YOLOv5l	89.3%	74.4%	85.2%	58.5%
YOLOv8l	91.6%	83.1%	88.5%	60%

B. Evaluation Metrics

Figures 7 and 8 show the confusion matrices for the models trained on non-augmented and augmented datasets, respectively, revealing interesting findings:

- Increased True Positives (TP): Both YOLOv5l and YOLOv8l exhibit higher TP for classes, such as blossom end rot rotation, splitting rotation, and sun-scaled rotation after data augmentation, indicating an improved ability to correctly identify these diseases.
- Reduced False Positives (FP): YOLOv8l demonstrates lower FP in most classes compared to YOLOv5l, suggesting better precision in disease classification after augmentation.

These observations indicate that YOLOv8l exhibits superior performance in terms of both TP and FP, signifying improved overall accuracy in disease detection. To quantitatively assess this change, the accuracy of the models was calculated before and after data augmentation using the confusion matrices. Accuracy was chosen because of its ability to provide a comprehensive view of the model's behavior across all disease classes. Equation (1) was employed to calculate the accuracy in these different scenarios:

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

where TN denotes True Negatives, and FN denotes False Negatives.

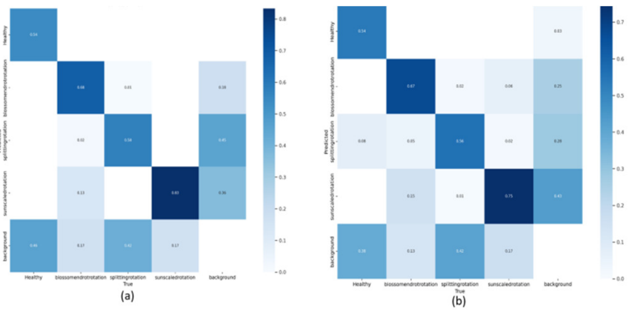


Fig. 7. Confusion matrices on the custom dataset before augmentation on (a) YOLOv8, (b) YOLOv5.

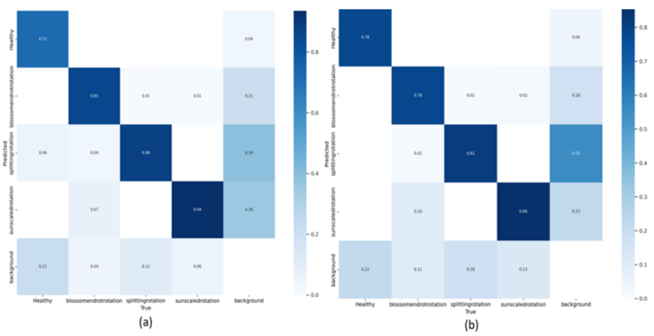


Fig. 8. Confusion matrices on the custom dataset after augmentation on (a) YOLOv8, (b) YOLOv5.

Analyzing the confusion matrix, a notable presence of background images was identified, devoid of relevant objects. These images can potentially skew the accuracy of the calculation. While a small number of background images might have minimal impact, a significant amount of them can lead to the underestimation of the model's true performance. Thus, accuracy was calculated in two ways to address this matter:

- Excluding background images: This approach disregards background images, potentially providing a more accurate representation of the model's ability to identify actual disease instances.
- Including background images: This method takes into account all images, involving those without disease, offering a more holistic view of the model's overall performance.

Table V portrays the accuracy values calculated using both approaches. YOLOv8l also outperforms YOLOv5l by significant margins, achieving accuracy gains of 3.67% and 9.25% before and after data augmentation, respectively. Figure 9 depicts the significant impact of augmented data on disease detection accuracy. YOLOv5 starts with a modest accuracy of 86.6% in the dataset without data augmentation and reaches a remarkable accuracy of 95.85% after augmentation. The accuracy of YOLOv8 increases from 94.3% to 97.9% utilizing data augmentation. These significant gains highlight the effectiveness of augmented data in improving the ability of models to accurately identify various tomato diseases.

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%

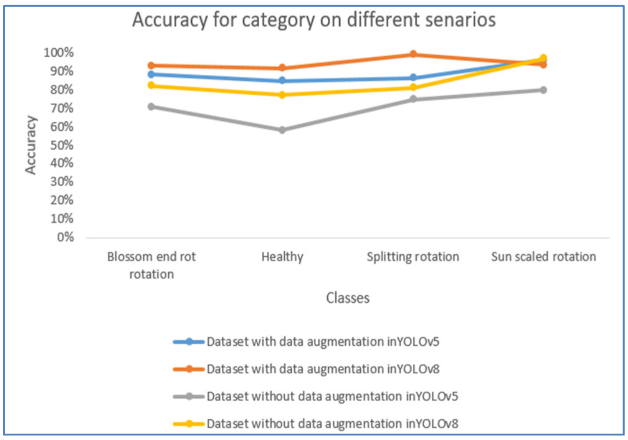


Fig. 9. Accuracy for each category in different scenarios.

VI. CONCLUSION

This study addresses the challenge of tomato disease detection by taking advantage of the power of deep learning techniques and data augmentation. The current study meticulously designed and collected a dataset in Roboflow that included over 1600 images naturally divided into training and validation sets. YOLOv5l and YOLOv8l were trained and validated on this dataset, and their initial performance achieved promising accuracy of 86.6% and 94.3%, correspondingly. To further improve performance, various data augmentation techniques, such as flipping, cropping, rotating, and altering the color space, were implemented to expand the dataset and enrich its content. Retraining both models on the augmented dataset revealed a remarkable impact, as both YOLOv5l and YOLOv8l models achieved significantly higher accuracy compared to the pre-augmented version. Specifically, YOLOv5l jumped from 86.6 to 95.85% and YOLOv8l climbed from 94.3 to a staggering 97.9%. Additionally, confusion matrix analysis demonstrated a decrease in FPs, showing increased precision in disease classification. Therefore, YOLOv8l showcased superior performance on various metrics, including TP, FP, and overall accuracy, especially when excluding background images. These significant findings disclose the power of data augmentation in boosting the effectiveness of deep-learning models for tomato disease detection. By expanding the dataset with diverse variations and addressing background noise, highly accurate and precise identification performance was achieved for various tomato diseases. Future research efforts will focus on enhancing this tomato disease detection system by improving background image training and augmenting the YOLO architecture. In addition, the creation of realistic disease-free backgrounds and focused enhancements will be investigated to reduce disease effects. Finally, attention mechanisms and multiscale feature fusion will be incorporated into the YOLO architecture to improve disease-specific focus and capture important patterns.

ACKNOWLEDGMENT

The authors extend their appreciation to the Deanship of Scientific Research at Northern Border University, Arar, KSA, for funding this research work through project number NBU-FFR-2024-1596-01.

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. Rajamohanan 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.

[19] A. Abbas, U. Maqsood, S. U. Rehman, K. Mahmood, T. AlSaedi, and 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.

[20] 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, pp. 8335–8341, Apr. 2022, https://doi.org/10.48084/etasr.4721.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] Q. H. Phan, V. T. Nguyen, C. H. Lien, T. P. Duong, M. T. K. Hou, and 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.

[26] "deseaseTomato Image Dataset." Feb. 17, 2024, [Online]. Available: https://universe.roboflow.com/datasetsnbu/deseasetomato/dataset/8.