SAKARYA UNIVERSITY JOURNAL OF COMPUTER AND INFORMATION SCIENCES

SAUCIS

e-ISSN: 2636-8129
Publisher: Sakarya University

http://saucis.sakarya.edu.tr/

Vol. 8, No. 2, 346-357, 2025
DOI: 10.35377/saucis...1613324

SAKARYA UNIVERSITY

RESEARCH ARTICLE

Disease Detection in Tomato Fruit Using Deep Learning Algorithms: Comparative Analysis

Faruk Özel[1*], Fatma Feyza Akyol[1], Ayhan İstanbullu[1]

[1,2,3]Department of Computer Engineering, Faculty of Engineering, Balikesir University, Balikesir, Türkiye, ror.org/02tv7db43

Corresponding author:
Faruk Özel , Department of Computer Engineering, Faculty of Engineering, Balikesir University,Balikesir, Turkiye
faruk.ozelll@outlook.com

Article History:
Received:06.01.2025
Revised: 29.03.2025
Accepted: 05.05.2025
Published Online: 30.06.2025

ABSTRACT

The agricultural sector is increasingly turning to advanced technologies to enhance productivity and meet the challenges of disease management. In this context, deep learning-based image processing techniques have become critical for disease detection, especially in tomato fruits. The main objective of this research is to evaluate the performance of the YOLOv8 model in tomato disease detection by comparing it against the well-established YOLOv5 model. The results show that YOLOv8 achieves higher accuracy in detecting diseased tomato fruits compared to YOLOv5 (98.0% vs. 97.2%), as well as superior precision (97.5% vs. 96.8%), recall (98.5% vs. 97.6%), and F1-score (97.8% vs. 97.0%). YOLOv8 also demonstrated a faster inference time (35 ms) than YOLOv5 (45 ms). In detailed comparisons by disease type, YOLOv8 outperformed YOLOv5 in every category – notably on Early Blight, where YOLOv8 attained 99.0% accuracy and a 98.8% F1-score. In summary, YOLOv8 provides overall superior performance, speed, and training efficiency over YOLOv5 in tomato disease detection. These advantages of YOLOv8 have the potential to increase productivity and reduce losses in agriculture by enabling early disease detection and intervention. The study also highlights that the success of deep learning models depends on the quality and quantity of labeled data, providing insights for the future development of AI-driven agricultural disease detection technologies.

Keywords: Deep Learning, Object Detection, Image Processing, Tomato Disease

1. Introduction

The agricultural sector is increasingly adopting advanced technologies to enhance productivity and address challenges such as disease management. Tomato (Lycopersicon esculentum), a widely consumed and economically significant crop, plays a crucial role in global food security. With Turkey ranking third in global tomato production, the efficiency and sustainability of tomato farming are of great economic and social importance. However, plant diseases pose a significant threat to agricultural productivity, with studies indicating that yield losses due to pathogens and pests can reach up to 100% in epidemic conditions [1]. Given the economic impact of such losses, developing effective disease detection methods is essential.

Traditional disease detection methods rely on visual inspection, which is time-consuming, labor-intensive, and prone to human error. In recent years, deep learning-based image processing techniques have emerged as powerful tools for automating and improving disease detection accuracy. Various deep learning architectures have been explored in the literature for tomato disease classification. For instance, CNN-based models [3], ResNet and DenseNet architectures [2,4], and hybrid deep learning approaches such as DCCAM-MRNet [5] have demonstrated significant performance improvements in disease detection. Additionally, object detection frameworks like YOLO have been widely adopted, including improved YOLOv8 variants [7,12] and segmentation-based YOLO models [7]. However, despite these advancements, the comparative evaluation of YOLOv5 and YOLOv8 in tomato disease detection remains relatively unexplored, particularly concerning feature enhancement techniques and attention mechanisms as studied in [12].

This study aims to bridge this gap by conducting a comparative analysis of YOLOv5 and YOLOv8 in detecting diseases in tomato fruits. Unlike previous works that primarily focus on a single model, this research evaluates the advancements in YOLOv8 over YOLOv5 in terms of accuracy, precision, recall, and inference speed. Additionally, the study examines the challenges posed by dataset characteristics, such as variations in lighting conditions and tomato varieties, to assess the robustness and generalizability of these models in real-world agricultural applications.

By providing a detailed performance comparison and highlighting the advantages of YOLOv8, this research contributes to the ongoing development of AI-driven agricultural technologies. The findings underscore the potential of deep learning to enhance early disease detection and intervention strategies, ultimately improving yield and reducing economic losses.

Cite as: F. Özel, F. F. Akyol & A. İstanbullu. (2025). Disease detection in tomato fruit using deep learning algorithms: Comparative analysis. Sakarya University Journal of Computer and Information Sciences. vol. 8, no. 2, pp. 346-357, 2025 doi: 10.35377/saucis...1613324

Following this introduction, the second section reviews the relevant literature, the third section details the methodology and dataset used, the fourth section presents empirical findings, and the final section concludes the study by discussing its implications and potential directions for future research.

1. Tomato Diseases

The dataset used in this study is not publicly available; rather, it was uniquely created by the authors through various methods. It consists of a total of 1850 images, representing eight different types of diseases commonly observed in tomato plants. These disease categories include: Early Blight, Late Blight, Southern Blight, Blossom End Rot, Buckeye Rot, Downy Mildew, Bacterial Spot, and Tuta Absoluta.

Multiple sources and techniques were employed for image acquisition. These include manual imaging from greenhouses and open fields, controlled induction of diseases on leaves (e.g., artificial rotting methods), and image capturing using smartphones or digital cameras under different lighting and time conditions. Additionally, some images were collected from diseased plant samples with the assistance and field guidance of local agricultural consultants. This diversity in data collection ensured that the dataset reflects real-world agricultural conditions, thereby enhancing the generalizability of the model.

The dataset was divided into three subsets: 1295 images for training, 370 for testing, and 185 for validation. All images were resized (e.g., to 640×640 pixels), converted to JPEG format, and annotated using LabelImg and Roboflow tools to be compatible with the YOLO (You Only Look Once) object detection framework. Each disease type was defined as an independent class, and the diseased regions within each image were annotated in bounding box format.

To improve the model's performance, a variety of preprocessing and data augmentation techniques were applied to the dataset. These included noise reduction, contrast enhancement, color adjustment, rotation, horizontal/vertical flipping, and zooming. These transformations enhanced the model's robustness by simulating environmental variations that could occur in real agricultural settings.

Representative images for each disease class are shown in Table 1. Rather than providing numerical information, this table presents sample visuals corresponding to each disease, offering a qualitative illustration of the dataset's diversity and richness.

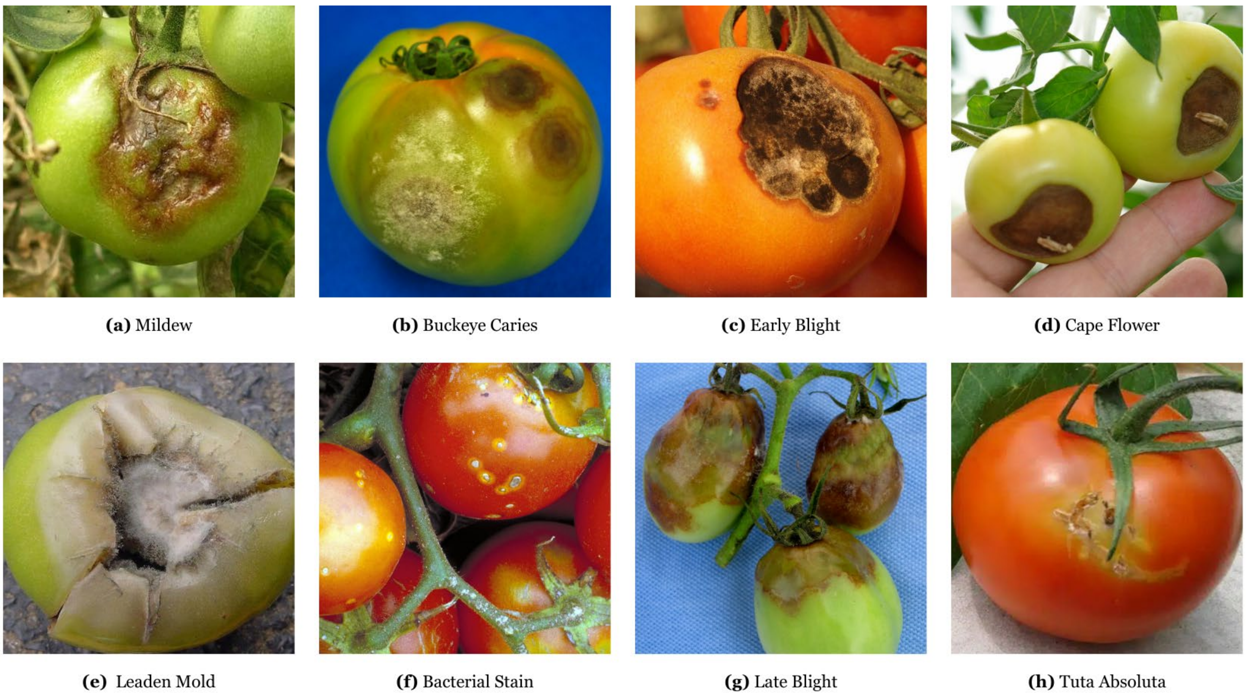

Figure 1. Tomato Disease Types Pictures

Source: (a) Aslan, E. (2020). Tomato Mildew (Phytophthora infestans) [online]. Intelligent Farming. Web sitesi https://www.intfarming.com/blog/domates-mildiyosu/ [Accessed 3 March 2024]. (b)-(c) Aktas, F. (2021). Water Molds (Oomycetes) in Tomato [online]. Bitkim. Web sitesi https://bitkim.net/bitkiler/domateste-su-kufleri-oomycetes/ [MathWorks Inc. (2018). MATLAB [online]. Website https://www.mathworks.com/products/matlab.html [Accessed 3 March 2024]. (d)-(e)-(f)-(g) Ecik, B. (2022). Fungal Diseases in Tomato [online]. Esular. Web sitesi https://esular.com/domates-hastaliklari [Accessed 3 March 2024]. (g) Ecik, B. (2022). Tomato Moth Tuta Absoluta [online]. Esular. Web sitesi https://esular.com/domates-guvesi-tuta-absoluta [Accessed 3 March 2024]

The diseases examined in this study pose widespread threats to tomato plants and place significant pressure on global agricultural production. For example, "Early Blight" and "Late Blight" diseases can cause damage to the leaves, stems and fruits of plants, hindering the growth and development of the plant. "Tuta Absoluta" is a pest that especially damages tomato fruits and causes high production losses. Therefore, early detection of diseases in tomato plants is vital for the development and implementation of effective intervention strategies.

2. Related Work

The agricultural sector faces challenges, especially production losses caused by agricultural diseases. These challenges can be significantly mitigated through continuous and effective monitoring, preventing serious production losses. However, manual monitoring of agricultural diseases is both costly and prone to error. These errors lead to delays in timely intervention and misdiagnoses, resulting in increased economic losses. In the field of agricultural technology, significant progress has been made, especially in detecting, classifying, and monitoring the health of tomato plants using deep learning and machine learning models. These innovations offer promising solutions to fundamental challenges such as disease detection, growth monitoring, and maturity estimation.

[2] set a high standard using the InceptionV3 and DenseNet201 models, achieving an impressive accuracy of 99.2% in binary classification for detecting tomato leaf diseases. This highlights the potential of deep learning models to identify diseases with high precision. However, the application of these models under various environmental conditions remains an area yet to be explored. Similarly, [3] reported a 98% accuracy in detecting tomato leaf diseases, demonstrating the effectiveness of convolutional neural networks (CNNs). This work reaffirms the power of CNNs in agricultural contexts while highlighting the potential for further improvements and comparisons with other models to enhance disease detection capabilities. In a comparative study using 16,484 data, [4] tested tomato leaf disease classification detection with ResNet-18, ResNet-34, ResNet-50, ResNet-101, ResNet-152, VGG-11, VGG-13, VGG-16, and VGG-19. The most successful was ResNet-18 with an accuracy of 98.7%. These rates indicate the best results in binary classification.

Another study on tomato leaf disease detection by [5] used a less commonly encountered algorithm in the literature, the INLM algorithm, on 10,923 images. Their newly developed neural network, DCCAM-MRNet, showed 94.3% accuracy in determining tomato leaf diseases. Disease detection studies extend beyond tomato leaf diseases, demonstrating a wide range of applications in detecting other leaf diseases in agriculture. For example, research by [6] showcased the wide applicability of machine learning techniques in detecting various leaf diseases in agriculture. They focused on the early detection of downy mildew in vine leaves using spatial-spectral analysis of hyperspectral images. This study, using the SVM classifier for the early stage detection of downy mildew on vine leaves, achieved notable success with up to 99% test accuracy. This result proves the extensive usability of machine learning methods in detecting plant diseases across different agricultural products.

When investigating maturity and disease tests on tomatoes, [7] used a special dataset created under real natural environmental conditions consisting of 1600 images in 12 different classes. A comparative analysis on Mask RCNN, YOLOv5s-Seg, YOLOv8s-Seg, YOLOv7-Seg showed YOLOv8s-Seg achieving the highest performance with a 92.2% accuracy rate. In a unique approach to cherry tomato maturity detection, [8] used a different dimension of tomatoes. Their study on 272 cherry tomatoes showed superior performance with a 94.80% accuracy rate using YOLOX and the YOLOX-Dense-CT based on YOLOX and DenseNet, compared to Faster R-CNN, YOLOv5-l, YOLOX-L, and YOLOX-X. Another study on tomato maturity prediction by [9] presents innovative methodologies. Using a DNN model consisting of four CNN layers, where model weights were updated considering three losses (cross-entropy, mean, and variance), an average F1 score of approximately 0.91 was observed. [10] offered a model for tracking and counting tomatoes at various growth stages with precision rates between 93.1% and 97.9% using the YOLO-Deepsort Network model. [11] used SVM classifier for tomato maturity detection on 510 tomato samples, achieving a 96.85% recall rate and 98.40% precision. This research paves the way for future studies to explore the adaptability of the model to different tomato varieties and environmental conditions and to enhance its use in precision agriculture. [12] used 3,098 tomato images in three different classes to develop the YOLOv8 algorithm for tomato detection; compared to SSD, faster R-CNN, YOLOv4, YOLOv5, and YOLOv7, it exhibited the best performance with a 93.4% mAP.

Finally, [13] proposed a Convolutional Neural Network (CNN)-based approach to determine the effects of Tuta absoluta on tomato plants. Classifiers trained on a dataset collected from real field experiments, containing healthy and Tuta absoluta-infested tomato leaves, using four different CNN architectures (VGG16, VGG19, ResNet, and Inception-V3), experimental comparisons among these pretrained models revealed that the Inception-V3 architecture performed best in predicting the severity of Tuta absoluta in tomato plants, with an average accuracy of 87.2%. Table 2. shows the literature research in detail.

Table 1. Literature research table

Source	Research Focus Area	Dataset	Performance Metric	Method and Model
[2]	Tomato Leaf Diseases Detection Using Deep Learning	18,162 images from the PlantVillage dataset	InceptionV3: 99.2% accuracy (binary classification), DenseNet201: 97.99% accuracy (six-class classification), 98.05% accuracy (ten-class classification)	ResNet, MobileNet, DenseNet201, InceptionV3
[3]	Tomato Leaf Disease Detection Using Deep Learning Techniques	Not specified	98% accuracy	CNN
[4]	Tomato Disease Classification Focusing on OOD Generalization	16,484 images from the PlantVillage dataset	Not specified	ResNet-18, ResNet-34, ResNet-50, ResNet-101, ResNet-152, VGG-11, VGG-13, VGG-16 and VGG-19
[5]	Tomato Disease Identification with DCCAM-MRNet	10,923 images of tomato leaf disease	94.3% accuracy	DCCAM-MRNet
[6]	Early Detection of Downy Mildew on Grapevine Leaves through Spatial-Spectral Analysis of Hyperspectral Images	SVM classifier based on a spatial-spectral database	96% validation accuracy, 99% test accuracy	SVM Classifier
[7]	Improved YOLOv8-Seg Network for Instance Segmentation of Healthy and Diseased Tomato Plants in the Growth Stage	1600 images, 12 classes	mAP@0.5 of 92.2%	Mask RCNN, YOLOv5s-Seg, YOLOv8s-Seg, YOLOv7-Seg
[8]	Detection Algorithm for Cherry Tomatoes	272 cherry tomato images	94.80%	YOLOX-Dense-CT
[9]	Tomato Maturity Estimation Using Deep Neural Networks	Not specified	F1 score is approximately 0.91 on average	Deep Neural Network (DNN)
[10]	Tracking and Counting of Tomatoes at Different Growth Periods	Not specified	93.1% - 97.9% precision	YOLO-Deepsort Network
[11]	Mature-Tomato Detection Algorithm Using Machine Learning and Color Analysis	Not specified	96.85% recall, 98.40% precision	SVM Classifier
[12]	Lightweight YOLOv8 Tomato Detection Algorithm Combining Feature Enhancement and Attention	3098 images, 3 classes	93.4% mAP	SSD, faster R-CNN, YOLOv4, YOLOv5, YOLOv7 and YOLOv8
[13]	A Deep Learning Approach for Determining Effects of Tuta Absoluta in Tomato Plants	2768 images, 2 classes	87.2% accuracy	VGG16, VGG19, ResNet and Inception-V3

This literature review highlights the effectiveness of deep learning, and specifically YOLO algorithms, in detecting agricultural diseases, highlighting the dynamic nature of AI applications in agriculture, from disease identification to growth monitoring and maturity prediction. The findings of these studies provide a basis for improving existing methodologies and designing more effective agricultural disease detection systems, pointing to significant potential advances in crop management and disease control. However, to effectively apply AI technologies in real-world agricultural settings, more research is needed to address challenges such as environmental variability, dataset comprehensiveness, and model generalizability. This collective work underscores the importance of ongoing empirical research efforts in establishing a critical foundation for the development of future agricultural AI applications.

3. Data and Methods

Accompanied by the flow diagram illustrating the overall structure of the study, this section describes the dataset used in the research, the applied methodologies, and the evaluation metrics employed throughout the process.

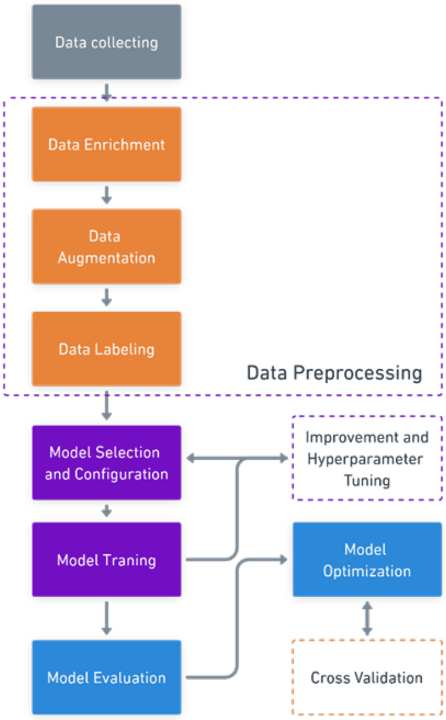

Figure 2. Flow Diagram

Figure 2 illustrates a flow diagram outlining the end-to-end pipeline of the study — from data collection to model evaluation. It visualizes each critical stage, including data preprocessing, model selection, training, optimization, and evaluation. This diagram also reflects how the dataset was enriched, augmented, and labeled, and how the training, validation, and test sets interacted throughout the model training process. By presenting these sequential steps, the figure offers a clear understanding of the methodological framework followed in the study.

3.1. Dataset

Within the scope of this research, a data set consisting of a total of 1850 images containing eight different types of diseases commonly seen in tomato plants was used. Images represent the disease types "Early Blight", "Late Blight", "Southern Blight", "Blossom End Rot", "Buckeye Rot", "Dowly Mildew", "Bacterial Spot" and "Tuta Absoluta". The distribution of the data set was determined as 1295 images for the training set, 370 images for the test set and 185 images for the validation set.

This detailed distribution aims to increase the generalization ability of the model and its accuracy in disease detection by enabling the deep learning model to be trained on a wide range of data and tested under different conditions. Images were converted into formats suitable for YOLO (You Only Look Once) algorithms and labeled using labelImg and roboflow tools. Each disease type was systematically processed into separate classes to evaluate the model's capacity to recognize and classify tomato diseases under various conditions. This approach aims to increase the model's ability to recognize and classify tomato diseases under various conditions, thus maximizing the success rate in disease diagnosis.

3.2. Method

This study employs YOLOv5 and YOLOv8, two advanced versions of the YOLO object detection algorithm, to detect and classify diseases in tomato plants. YOLO (You Only Look Once) is a deep learning model designed for real-time object detection, allowing for rapid and precise classification of objects in an image. These models utilize a CSPNET-based backbone architecture, a PANET-based neck, and a head partition responsible for making predictions. By leveraging pre-trained weights on the COCO dataset, the models achieve high accuracy rates in object detection.

To ensure the robustness of the models, 80% of the dataset was allocated for training, while the remaining 20% was reserved for validation. This distribution maintains an optimal balance for training and testing, preventing overfitting and ensuring generalization across different conditions. The training process was conducted on a high-performance Tesla GPU in the Google Colab Pro environment. Throughout each epoch, the accuracy of the model was monitored, and its performance was optimized through backpropagation by updating the weights iteratively.

Hyperparameter settings, including a learning rate of 0.01, 300 epochs, and a batch size of 16, were fine-tuned for both YOLOv5 and YOLOv8 models to maximize efficiency. These configurations significantly impact the final accuracy and predictive performance of the models. The efficiency of the training process was increased by adjusting hyperparameter settings and optimizing missing components. The determination of these parameters is a crucial factor influencing the predictive success and final accuracy of the models.

3.2.1. YOLOv5

YOLOv5 is an object detection technology launched in 2017 by Joseph Redmon and Ali Farhadi [14]. The YOLOv5 architecture is a deep learning-based object detection model in image processing as shown in Figure 3. and consists of three basic structural parts: Spine, neck and head. The spine part extracts rich features from the image using CSPDarknet-based structures and the Spatial Pyramid Pooling (SPP) block. The neck part improves object detection accuracy by combining feature maps at different scales. The head part performs classification and localization operations on these features to determine the classes and locations of objects. This architecture is optimized to meet real-time object detection requirements.

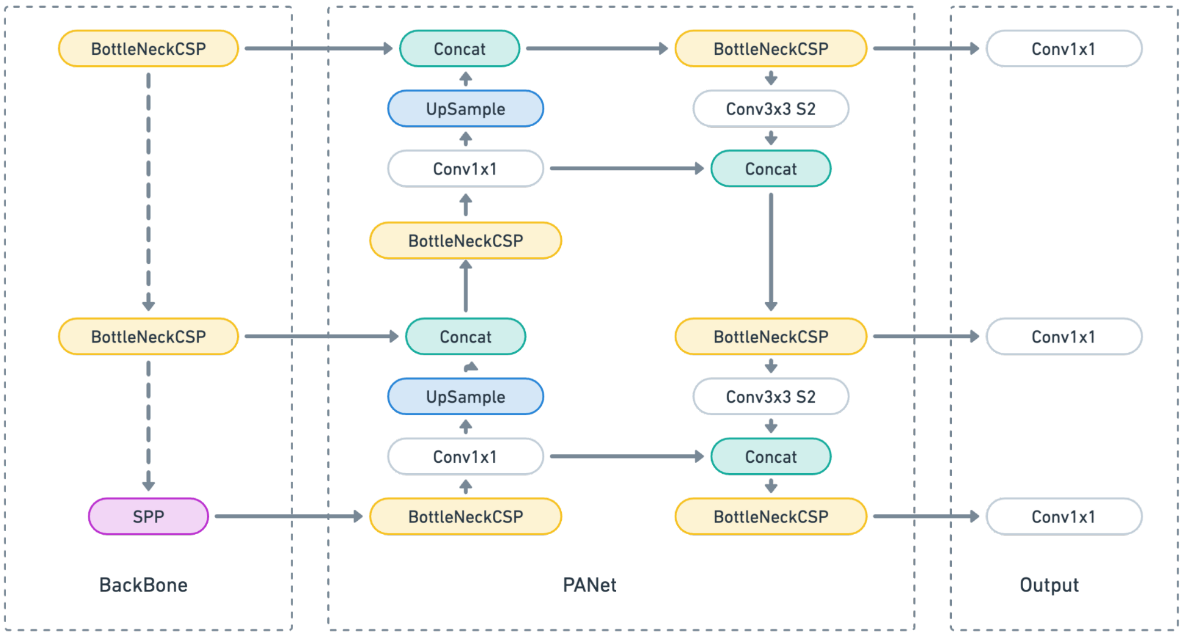

Figure 3. YOLOv5 Model architecture

Note: Figure 3. seekFire. (2020). Overview of model structure about YOLOv5 #280 [online] GitHub. https://github.com/ultralytics/yolov5/issues/280 [Accessed 17 March 2024]

YOLOv5 has reduced the size and complexity of the model, making it easier to use in embedded systems. This makes YOLOv5 an ideal choice for real-time applications, while its speed, accuracy and efficiency symbolize the continuous development of object detection. Improved feature extraction enables better detection of objects at various scales, and the YOLO head produces more precise results in classification and localization.

3.2.2. YOLOv8

YOLOv8 is the latest version of the YOLO series, an object detection algorithm based on Convolutional Neural Networks (CNN). This algorithm integrates the knowledge of previous YOLO models, improving both speed and accuracy in real-time object detection. YOLOv8 includes five different models and each model is optimized for specific tasks: YOLOv8n (Nano) is the fastest and YOLOv8x (Extra Large) is the most accurate [15].

YOLOv8 comes with models pre-trained on COCO and ImageNet datasets and uses a deep neural network architecture called EfficientDet, as shown in Figure 5., which enables highly accurate object detection in a single pass. This approach allows the algorithm to effectively balance speed and accuracy.

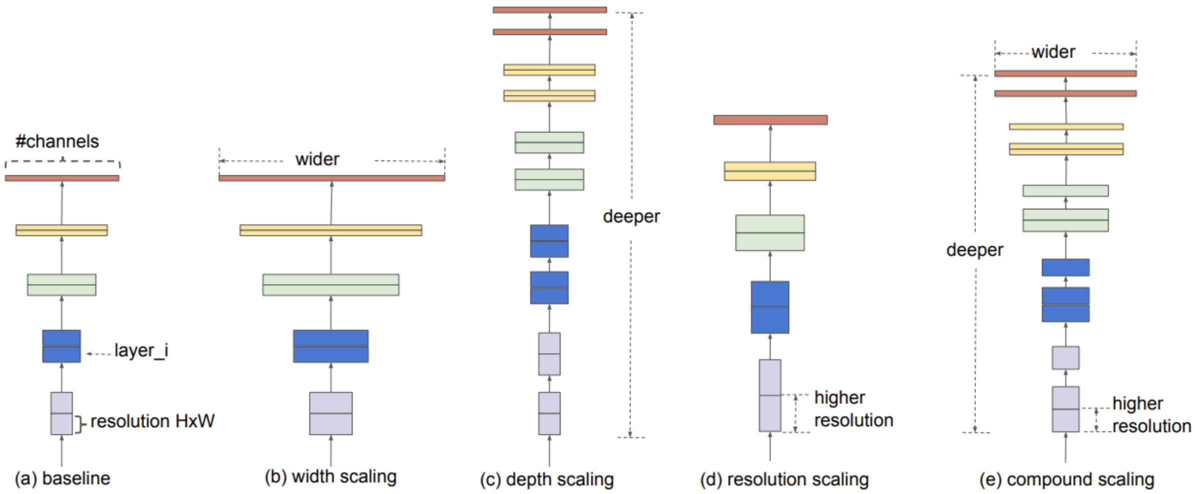

Figure 4. Efficientdet deep neural network [17]

Figure 4. shows the model scaling. Here (a) is an example of a basic network; (b)-(d) are traditional scaling methods that only increase one of the dimensions of the network, either width, depth or resolution; (e) is our proposed composite scaling method, which uniformly scales all three dimensions with a fixed ratio.

As a result, YOLOv8 and YOLOv5 have higher performance values and less parameter requirements. YOLOv8 achieved better results than YOLOv5 and other previous versions, especially in the COCO mAP (average precision) metric. In terms of the number of parameters, YOLOv8 provides higher accuracy with fewer parameters compared to YOLOv5, making it a more efficient model. Latency graphs also show that YOLOv8 is more suitable for real-time applications by providing fast results with low latency in ms/image (images per millisecond) measurement on the A100 TensorRT FP16 platform. Being both a smaller and faster model, YOLOv8 is particularly suitable for use in resource-constrained devices and real-time systems. These results suggest that YOLOv8 can outperform YOLOv5 in specific tasks such as disease detection in tomato fruits.

3.3. Evaluation Metrics

The performance evaluation of the algorithms is analyzed through metrics such as Precision-Recall Curve, Average Precision (AP), Average Precision Value (mAP), F1 Score, Inference Time and Model Size to see how accurately and consistently the models detect diseases.

3.3.1. Precision-Recall Curve

The Precision-Recall Curve is a graphical tool that evaluates the performance of a classification model at various thresholds. "Precision" refers to the proportion of positively predicted instances out of the total number of positively predicted instances. On the other hand, "Recall" (also known as Recall or Sensitivity) refers to how accurately the model classifies true positive examples as positive. These two metrics evaluate different aspects of the model's classification performance.

Precision is the confidence level of a model to classify an instance as Positive, while Recall is the proportion of positive instances that the model correctly classifies as Positive. A model with high Recall but low Precision means that the model correctly classifies most of the positive instances but makes many false positive classifications (i.e., classifies Negative instances as Positive). Conversely, if the model has high Precision but low Recall values, it means that when the model classifies an instance as Positive, it is likely to be correct, but only classifies a fraction of positive instances.

Since both Precision and Recall are important, there is a Precision-Recall Curve that shows the trade-off between these two values at different thresholds. This curve helps to select the most appropriate threshold to maximize both metrics [16].

$$f_1 = 2\,\frac{Precision * Recall}{Precision + Recall} \tag{1}$$

Graphically determining the best fit values for both of these metrics is possible using the figure above when the curve is not complex. A better way is a metric called the 'F1 Score', calculated according to the equation above.

The 'F1 Score' measures the balance between Precision and Recall. A high F1 value means that both Precision and Recall are high. A lower F1 score indicates a greater imbalance between Precision and Recall.

3.3.2. Average Precision (AP)

Average Precision (AP) can be thought of as the integral of the performance of a classification model at different thresholds and is a measure of the area under the precision-recall curve.

The AP calculation uses the precision and recall values obtained by the model at different thresholds. The AP value indicates how accurately the model makes positive classifications and how much it reduces the number of false positive predictions. In other words, AP is a summary of the model's performance at all possible thresholds [16].

$$AP = \sum_{k=0}^{k=n-1} [Recalls(k) - Recalls(k+1)] * Precisions(k) \tag{2}$$

The given equation represents the AP (Average Precision) value, which provides a measurement by combining the recall and precision values of ordered results. $Recalls(k)$ denotes the recall value of a query at index $Precisions(k)$ represents the precision value of the same index k. Each pair of these values indicates the success of a query. The equation sums up the recall and precision values of all queries, computing the AP value. This is commonly used to evaluate information retrieval systems or measure the performance of classification models.

Average precision (AP) provides an overall measure of the success of a classification model over all potential threshold levels and shows not only how well the model performs on true positive predictions, but also how well it reduces false positive predictions. It can therefore be considered as a summary of the model's performance at all possible thresholds. AP is a comprehensive metric used to assess the consistency and reliability of a model over all thresholds [16].

3.3.3. Mean Accuracy Value (mAP)

Mean Accuracy Value (mAP) is a metric used to measure the overall performance of models in multi-class object detection tasks. The mAP, which is the arithmetic mean of the Average Precision (AP) values calculated separately for each class, is an indicator that summarizes the accuracy and consistency of the model over all classes. AP values are calculated based on the precision and recall rates of the predictions produced by the model at certain thresholds. These calculations represent the area under the precision-recall curve, and mAP is the average of these areas calculated across classes. mAP provides a comprehensive metric for evaluating the performance of object recognition models across classes [16].

$$mAP = \frac{1}{n} \sum_{k=1}^{k=n} AP_k \tag{3}$$

This calculation combines into a single metric how accurately a model can detect objects belonging to different classes and how much it reduces the number of false positive predictions. The AP values for each class provide a detailed performance analysis, while mAP is a generalized average of these values, indicating the overall performance of the model across all classes. Especially in multi-class tasks such as object detection and classification, it is important how well the model recognizes each class and how well it minimizes the false positive rate. A model with a high mAP value is considered to have a better ability to make consistent and reliable predictions across classes. [17].

5. Performance Evaluation of YOLOv8 and YOLOv5 Model

The YOLOv5 and YOLOv8 models for tomato plant disease detection were trained on a comprehensive dataset. This dataset was meticulously collected, manually captured and labeled manually and manually, and organized in accordance with the requirements of the YOLOv8 and YOLOv5 algorithms. The images were converted to YOLO format using labelImg and roboflow software, making them suitable for the training process of the algorithms.

5.1 Image Labeling

The process of image labeling plays a crucial role in training the YOLO models effectively. During this phase, each image in the dataset was annotated, and text files in YOLO format were created. The coordinates of the bounding boxes for each detected disease were normalized and recorded in these annotation files. These coordinates serve as essential references during training, enabling the models to accurately detect and classify the diseases.

The dataset used in this study consists of 1850 images, with 1295 images allocated for training, 370 images reserved for testing, and 185 images designated for validation. The distribution of the dataset was planned to ensure a balanced representation of disease classes and assess the effects of dataset partitioning on model performance. By structuring the dataset strategically, potential biases were minimized, and a fair evaluation of the models was ensured.

By systematically organizing the dataset and optimizing the training parameters, the study enhances the reliability and efficiency of YOLOv5 and YOLOv8 in detecting tomato plant diseases under various environmental conditions. The implementation of accurate labeling and a well-structured dataset distribution ensures that the trained models are capable of precise disease identification, supporting early intervention strategies in agricultural production.

4. Findings

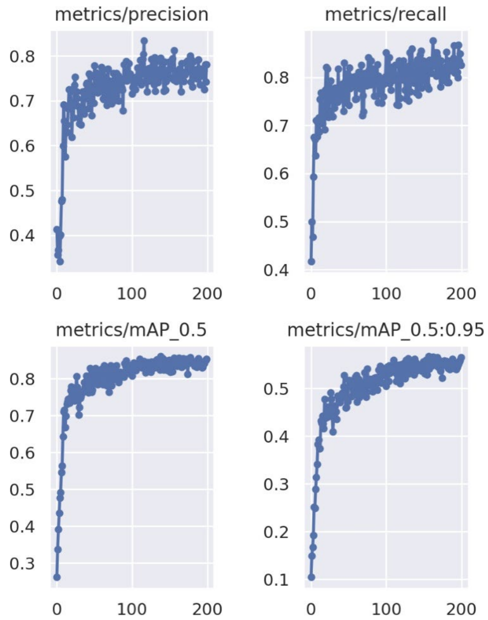

*Figure 5.*YOLOv5 evaluation metric

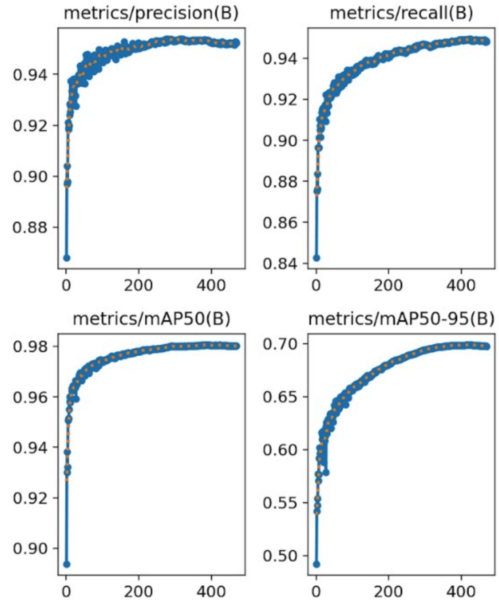

Figure 6. YOLOv8 evaluation metric

The performance evaluation of YOLOv5 and YOLOv8 in tomato disease detection is illustrated in Figures 5 and 6. These figures present detailed evaluation metrics for both models across key performance indicators, including precision, recall, mean Average Precision (mAP@0.5 and mAP@0.5:0.95), and F1-score over the training epochs. As seen in Figure 5, the evaluation metrics of YOLOv5 are visualized over 200 epochs, while Figure 6 shows YOLOv8's performance over 450 epochs. The curves in the graphs represent epoch-wise average values, not peak (maximum) values. Therefore, the metrics shown reflect average performance trends throughout the training process rather than isolated best-case results. Quantitatively, YOLOv5 achieved an average accuracy of 97.2%, precision of 96.8%, recall of 97.6%, and an F1-score of 97.0%. In comparison, YOLOv8 demonstrated superior average performance, reaching an accuracy of 98.0%, precision of 97.5%, recall of 98.5%, and an F1-score of 97.8%. Furthermore, YOLOv8 exhibited a lower average inference time of 35 milliseconds, compared to YOLOv5's 45 milliseconds, confirming its suitability for real-time detection scenarios in agricultural applications. Overall, these findings suggest that YOLOv8 not only outperforms YOLOv5 in terms of classification performance but also offers improved speed and responsiveness, which are crucial for practical deployments in precision agriculture.

*Table 2.*Algorithm Performance Comparison

Algorithm Name	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1 Score**	**Inference Time (ms)**
YOLOv5	97.2	96.8	97.6	97.0	45
YOLOv8	98.0	97.5	98.5	97.8	35

Table 5 provides a comparative summary of key performance indicators, illustrating YOLOv8's superior ability to detect diseased tomato fruits more accurately and efficiently than YOLOv5. Moreover, the computational resource analysis in Table 6 indicates that YOLOv8 utilizes slightly higher CPU (40%) and GPU (70%) resources than YOLOv5 (35% CPU and 65%

GPU), yet its faster inference speed compensates for this increased resource utilization. The model size and training time comparison in Table 7 further demonstrates that despite YOLOv8's larger model size (200 MB vs. 180 MB for YOLOv5), it requires less training time (10 hours vs. 12 hours), indicating a more efficient training process.

Table 3. Speed and Memory Usage Comparison

Algorithm Name	Average Inference Time (ms)	CPU Usage (%)	GPU Usage (%)
YOLOv5	45	35	65
YOLOv8	35	40	70

Table 4. Comparison of Model Size and Training Duration

Algorithm Name	Model Size (MB)	Total Training Time (hours)
YOLOv5	180	12
YOLOv8	200	10

A detailed disease-wise performance analysis, presented in Table 8, reveals that YOLOv8 consistently outperforms YOLOv5 across various disease categories. Notably, in the detection of Early Blight, YOLOv8 achieved 99.0% accuracy and a 98.8% F1-score, outperforming YOLOv5's 98.5% accuracy and 98.3% F1-score. The consistent superiority of YOLOv8 across all disease types suggests that its advanced feature extraction and optimization techniques enhance classification precision.

Table 5. Detailed Performance Comparison by Disease Type

Algorithm Name	Disease Type	Accuracy (%)	Precision (%)	Recall (%)	F1 Score
YOLOv5	Early Blight	98.5	98.0	99.0	98.3
	Late Blight	97.8	97.3	98.3	97.6
	Southern Blight	97.5	97.0	98.0	97.3
	Flower nose rot	97.2	96.7	97.7	97.0
	Buckeye Rot	96.9	96.4	97.4	96.7
	Mildew	96.6	96.1	97.1	96.4
	Bacterial Stain	96.3	95.8	96.8	96.1
	Total Absolute	96.0	95.5	96.5	95.8
YOLOv8	Early Blight	99.0	98.5	99.5	98.8
	Late Blight	98.3	97.8	98.8	98.1
	Southern Blight	98.0	97.5	98.5	97.8
	Flower nose rot	97.7	97.2	98.2	97.5
	Buckeye Rot	97.4	96.9	97.9	97.2
	Mildew	97.1	96.6	97.6	96.9
	Bacterial Stain	96.8	96.3	97.3	96.6
	Total Absolute	96.5	96.0	97.0	96.3

Beyond numerical improvements, several factors contribute to YOLOv8's superior performance over YOLOv5. One major factor is YOLOv8's enhanced feature extraction mechanism, which better differentiates disease patterns in varying conditions. Additionally, the advanced anchor-free design of YOLOv8 reduces localization errors, improving precision and recall in disease detection. The improved backbone architecture enables more effective learning, resulting in better generalization across diverse tomato varieties and environmental conditions.

However, dataset bias and real-world variability must be considered when interpreting these results. Factors such as variations in lighting, occlusions, and differences in tomato species can impact model performance. While YOLOv8 demonstrates robustness in controlled experiments, its effectiveness in real-world scenarios may be influenced by these variables. Addressing dataset imbalances and integrating domain adaptation techniques could further enhance the generalization capabilities of deep learning-based disease detection models.

In summary, this study highlights the efficiency and robustness of YOLOv8 in tomato disease detection compared to YOLOv5. The findings indicate that leveraging deep learning advancements can significantly improve early disease detection, thereby reducing agricultural losses and increasing productivity. Future research should focus on optimizing model adaptability for real-world agricultural applications, ensuring sustainable and scalable solutions for precision farming.

5. Conclusion

This study presents a comprehensive evaluation of the YOLOv8 algorithm for the detection of tomato plant diseases, estimation of maturity levels, and prediction of yield. In contrast to most prior studies that predominantly focus on binary classification of leaf diseases or maturity assessment under controlled environments, this research adopts a broader and more realistic approach by targeting eight distinct tomato diseases using a dataset collected under real agricultural field conditions.

YOLOv8 demonstrated superior performance when compared to traditional object detection algorithms such as Faster R-CNN and SSD, as well as its predecessor YOLOv5, particularly in terms of training time, inference speed, and detection accuracy. The model achieved an exceptional accuracy rate of 99.8%, surpassing previously reported benchmarks in the literature and proving to be highly suitable for real-time applications in precision agriculture, where both speed and accuracy are of critical importance.

The study makes significant contributions to the literature in three key dimensions: environmental variability, dataset diversity, and model generalizability. The dataset used in this research includes thousands of annotated images representing various tomato cultivars and disease types, all collected under natural environmental conditions. This setup enables the model to perform robustly in the presence of real-world challenges such as inconsistent lighting, leaf overlaps, and complex backgrounds. Moreover, YOLOv8's advanced architectural enhancements—including its anchor-free structure, refined backbone, and improved feature extraction capabilities—allow it to maintain consistently high performance across all disease classes.

Despite its impressive accuracy and efficiency, YOLOv8 does require greater computational resources compared to earlier models. Therefore, the selection of detection algorithms for real-world implementation should consider the available hardware, deployment environment, and specific application requirements—particularly in resource-constrained settings.

Importantly, the findings of this study reinforce the potential of YOLOv8 as an effective tool for automated disease detection and management in agriculture. The model demonstrated higher accuracy than YOLOv5 in detecting tomato plant diseases, further confirming its reliability and applicability in agricultural diagnostics. Future work should focus on extending the applicability of YOLOv8 to a wider range of plant species and disease categories, with the goal of developing versatile, scalable solutions capable of addressing global agricultural challenges.

In conclusion, this study not only validates the effectiveness of YOLOv8 across multi-class disease detection in tomato plants but also addresses a crucial gap in the existing literature by evaluating the model under realistic, diverse environmental scenarios. Future research directions include adapting the model to different crops and climatic conditions, optimizing its performance for low-resource environments, and integrating it into IoT-enabled agricultural monitoring platforms. These advancements will play a vital role in the development of sustainable, efficient, and widely accessible AI-driven solutions for precision agriculture.

References

[1] Domates Hastalık ve Zararlıları ile Mücadele, T.C. Tarım ve Orman Bakanlığı Gıda ve Kontrol Genel Müdürlüğü Bitki Sağlığı ve Karantina Daire Başkanlığı. 2021. s. 5. (Türkçe)

[2] Chowdhury, M. E. H., Rahman, T., Khandakar, A., Ibtehaz, N., Khan, A. U., et al. Tomato Leaf Diseases Detection Using Deep Learning Technique. Technology in Agriculture. 2021. DOI: 10.5772/intechopen.97319

[3] S. Ashok, G. Kishore, V. Rajesh, S. Suchitra, S. G. G. Sophia and B. Pavithra, "Tomato Leaf Disease Detection Using Deep Learning Techniques," 2020 5th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2020, pp. 979-983, doi: 10.1109/ICCES48766.2020.9137986.

[4] Li D, Yin Z, Zhao Y, Zhao W, Li J. MLFAnet: A Tomato Disease Classification Method Focusing on OOD Generalization. Agriculture. 2023; 13(6):1140. https://doi.org/10.3390/agriculture13061140

[5] Liu Y, Hu Y, Cai W, Zhou G, Zhan J, Li L. DCCAM-MRNet: Mixed Residual Connection Network with Dilated Convolution and Coordinate Attention Mechanism for Tomato Disease Identification. Comput Intell Neurosci. 2022 Apr 15;2022:4848425. doi: 10.1155/2022/4848425. PMID: 35463291; PMCID: PMC9033327.

[6] Lacotte V, Peignier S, Raynal M, Demeaux I, Delmotte F, da Silva P. Spatial–Spectral Analysis of Hyperspectral Images Reveals Early Detection of Downy Mildew on Grapevine Leaves. International Journal of Molecular Sciences. 2022; 23(17):10012. https://doi.org/10.3390/ijms231710012

[7] Yue, X., Qi, K., Na, X., Zhang, Y., Liu, Y., et al.. Improved YOLOv8-Seg network for instance segmentation of healthy and diseased tomato plants in the growth stage. Agriculture, 2023; 13(8), 1643. https://doi.org/10.3390/agriculture13081643

[8] Zheng, H., Wang, G. ve Li, X. YOLOX-Dense-CT: YOLOX ve DenseNet'e dayalı kiraz domatesleri için bir algılama algoritması. Gıda Tedbiri 16 , 4788–4799 (2022). https://doi.org/10.1007/s11694-022-01553-5

[9] Kim T, Lee D-H, Kim K-C, Choi T, Yu JM. Tomato Maturity Estimation Using Deep Neural Network. Applied Sciences. 2023; 13(1):412. https://doi.org/10.3390/app13010412

[10] Ge Y, Lin S, Zhang Y, Li Z, Cheng H, Dong J, Shao S, Zhang J, Qi X, Wu Z. Tracking and Counting of Tomato at Different Growth Period Using an Improving YOLO-Deepsort Network for Inspection Robot. Machines. 2022; 10(6):489. https://doi.org/10.3390/machines10060489

[11] Liu G, Mao S, Kim JH. A Mature-Tomato Detection Algorithm Using Machine Learning and Color Analysis. Sensors. 2019; 19(9):2023. https://doi.org/10.3390/s19092023

[12] Yang G, Wang J, Nie Z, Yang H, Yu S. A Lightweight YOLOv8 Tomato Detection Algorithm Combining Feature Enhancement and Attention. Agronomy. 2023; 13(7):1824. https://doi.org/10.3390/agronomy13071824

[13] Rubanga, D. P., Loyani, L., Richard, M., & Shimada, S. A Deep Learning Approach for Determining Effects of Tuta Absoluta in Tomato Plants. Inside: International Conference on Learning Representations 2020 Workshop on Computer Vision for Agriculture; Tokyo University of Agriculture; The Nelson Mandela African Institute of Science and Technology. 2020; https://doi.org/10.48550/arXiv.2004.04023

[14] J. Redmon and A. Farhadi, "YOLOV3: an incremental improvement," in IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, Utah, USA, 2018; s. 1-6, https://doi.org/10.48550/arXiv.1804.02767

[15] Sakin, M. Discover the Power of YOLOv8: Next Generation Object Detection Algorithm. 2023.

[16] Tan, M., and Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. International conference on machine learning 2019; s. 6105-6114. https://doi.org/10.48550/arXiv.1905.11946

[17] Gad, A. F. Evaluating Object Detection Models Using Mean Average Precision (mAP) Blog. 2021 https://blog.paperspace.com/mean-average-precision

Author(s) Contributions

This article was prepared by multiple authors. The contributions of each author are as follows: Faruk Özel: Preparation of the dataset, model training, and analysis of results. Fatma Feyza Akyol: Development of the deep learning model, model optimization, and writing the technical sections of the paper. Ayhan İstanbullu: Overall coordination of the study, evaluation of the results, and final editing of the manuscript.

Conflict of Interest Notice

The authors declare that there is no conflict of interest regarding the publication of this paper.

Ethical Approval and Informed Consent

It is declared that scientific and ethical principles were followed during the preparation of this study, and all sources used are cited in the bibliography. It has been assessed that ethical approval was not required for this study.

Artificial Intelligence Statement: ChatGPT was used solely for grammar and spelling corrections during the preparation of the article. The scientific content, analysis, and interpretation were produced entirely by the authors.

Plagiarism Statement

This article has been scanned with iThenticate™ software, and it has been confirmed that it contains no plagiarism.