International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2022): 7.942

AI-Driven Plant Disease Detection: Leveraging Deep Learning for Accurate Plant Disease Detection from Leaf Images

Shanukumar Singh[1], Zek Furtado[2], Apurv Patil[3]

[1]Email: *shanusingh2409[at]gmail.com*

[2]Email: *zekfurtado2303[at]gmail.com*

[3]Email: *patilapurv18[at]gmail.com*

Abstract: ***Plant diseases are a critical threat to global food security and agricultural sustainability because of the crop losses they cause. 195 million tons of crops are lost to fungal diseases each year and alone in India, more than 5 million metric tonnes go waste annually from it[1]. Also, from FAO "Globally up to 16% of harvests worth about US$220 billion are lost due to plant pests every year" [2] The urgency of the situation is clear, as wrapped up in these figures are reasons why long-term growth requires early bite detection services to prevent plant disease. In this paper, deep learning algorithms were used to detect diseases in plants by taking image of the leaves as input. Our research is limited to the detection of these 6 plant diseases, Aphid infestation, Bacterial leaf spot disease Black apple scab Early blight Septoria leaf spot on tomato Grape powdery mildew. In order to do this, we construct and train a machine learning model using two different raw image datasets. These were meticulously curated and enriched datasets, geared towards improving the model's generalisability across extensive variety of conditions. Our approach not only facilitates the early detection of these diseases but also demonstrates the potential for scalable, real-time applications in agricultural settings. The results highlight the effectiveness of deep learning in identifying and classifying plant diseases, offering a promising solution for reducing crop losses and improving agricultural productivity.***

Keywords: Plant Disease Detection, YOLOv8, Deep Learning, Image Processing, Data Augmentation, Agriculture Technology, Machine Learning, Leaf Image Classification, Precision Agriculture, Object Detection Model, AI in Agriculture, Crop Health Monitoring, Real-time Disease Detection

1. Introduction

Agriculture remains the backbone of the global economy, providing food, raw materials, and employment for a significant portion of the population. The health of crops is paramount to ensuring food security and sustaining economic stability, particularly in regions heavily dependent on agriculture. However, plants are constantly threatened by a wide range of diseases caused by fungi, bacteria, viruses, and pests. These diseases can lead to substantial crop losses, with severe economic repercussions. In India, for example, fungal diseases alone are estimated to cause losses exceeding 5 million tonnes of crops annually. On a global scale, the Food and Agriculture Organization of the United Nations (FAO) reports that plant pests account for 10-16% of global harvest losses, amounting to approximately US$220 billion each year.

Traditionally, plant disease detection has relied on manual inspection by farmers or agricultural experts, often requiring significant time and expertise. This process is not only labor-intensive but also prone to human error, leading to misdiagnosis or late detection of diseases. Early detection is critical, as it allows for timely intervention, preventing the spread of disease and minimizing crop damage. However, the manual methods currently in use often fail to provide the rapid, accurate diagnosis needed to protect crops effectively. The lack of scalable, automated solutions for early disease detection presents a significant challenge to the agricultural sector, limiting its ability to respond promptly to emerging threats.

To address these challenges, this paper proposes the development of an AI-driven application that leverages deep learning, specifically the YOLOv8 model, for real-time plant disease detection using leaf images. Our objective is to create a tool that can accurately identify and diagnose plant diseases at an early stage, enabling farmers to take immediate action and reduce crop losses. The application focuses on detecting six specific plant diseases: Aphid infestation, Bacterial leaf spot, Black spot apple, Early blight, Septoria leaf spot on tomato, and Powdery mildew on grape. By training two models on distinct datasets created from raw images, we aim to assess the effectiveness of this approach and its potential for broader agricultural applications.

This research explores the potential of deep learning models in revolutionizing plant health monitoring, offering a scalable, automated solution for real-time disease detection. Through this study, we seek to contribute to the growing body of knowledge on AI applications in agriculture and provide a practical tool for enhancing crop management practices.

2. Literature Review

- In one approach, as described in [16], the authors categorized the intensity of infection into different percentages: 20%, 40%, and 75%. Using these categories, they proposed a solution achieved through the Canny edge detection technique and the Gaussian Mixture Model (GMM). In a different study [14], the researchers identified plant diseases using an image processing technique involving K-means clustering, the Random

Forest algorithm, and the Gray Level Co-occurrence Matrix (GLCM). Although the authors did not specify the accuracy, they noted that the solution was notably fast.

- Another research effort [3] developed a system to identify plant diseases using image processing, employing various techniques to demonstrate overall precision, accuracy, recall, and F-measure. The accuracy of three algorithms was found to exceed 90%, with SVM (Linear Kernel) achieving 95.63%, SVM (RBF Kernel) 94.23%, and SVM (Polynomial Kernel) 95.87%. The SVM with a polynomial kernel provided the best result. In the proposed work detailed in [17], the experiment achieved an accuracy of 98.60%. The accuracy of the algorithm was tested using simple threshold and triangle thresholding methods.
- The authors in [4] suggested using MATLAB for detecting plant diseases during image processing. Although the authors claimed that the system would provide high accuracy, they did not mention the specific accuracy percentage. The system utilized K-means clustering and Support Vector Machine (SVM). In another study [9], the authors used image processing techniques, including K-means clustering, Artificial Neural Networks (ANN), and fuzzy classification, to identify and quantify signs of paddy leaf disease. Despite not providing the accuracy rate, the authors stated that the solution was highly accurate.
- For their research, the authors in [12] utilized a Convolutional Neural Network (CNN), achieving an excellent accuracy of 98%. Similarly, in [5], the authors proposed a system that detects unhealthy regions of plant leaves using texture features. The proposed algorithm was effective, with a 94% accuracy in detecting and classifying the studied diseases, achieved using Support Vector Machine and Minimum Distance Criterion.
- In another work [18], the authors presented a system for studying and evaluating the use of image processing in cotton leaf disease identification. Segmentation was conducted using the K-means clustering approach, while classification was done using neural networks, resulting in an accuracy of 89.56%. In [15], the authors suggested identifying and classifying plant leaf diseases using color transformation, Content-Based Image Retrieval (CBIR) methods, and K-means clustering. The accuracy of this study was reported to be 90.98%.
- Another approach [11] employed two cascaded classifiers. The authors proposed local statistical features and hue and luminance from the HSV color space. The KNN classifier was used, achieving an accuracy of 82.50%. Similarly, in [13], the authors used a convolutional neural network (CNN) innovatively to create a model for plant disease recognition based on leaf disease classification. A deep learning framework called Caffe was used to execute the CNN, resulting in an accuracy of 96.3%.
- In [19], the authors proposed a method for segmenting images to differentiate between two categories of orchid leaf diseases. The study used MATLAB to examine orchid leaves using boundary segmentation techniques, and morphological processing techniques were applied to classify the images, achieving an accuracy of 86.36%, which is considered moderate. Another study [6] described a web-based application that helps farmers identify fruit diseases by uploading photographs of the fruit. The system compared the trained dataset with the input dataset, utilizing the K-means clustering approach for clustering and Support Vector Machine (SVM) for classification. The approach was effective, achieving an accuracy of 82%.
- In [7], the authors proposed an SVM classifier to automatically detect rice leaf disease using image processing techniques. Features were extracted using the Scale-Invariant Feature Transform (SIFT), and both SVM and KNN classifiers were used to analyze the results. The accuracy results were 95.5% for SVM and 92.2% for KNN. A similar study [8] used digital image processing methods to examine and identify plant leaf diseases. Segmentation was carried out using K-means clustering, and features were extracted using GLCM and Local Binary Patterns (LBP). The classification was done using three classifiers: KNN, SVM, and Ensemble. Among these, the best accuracy was achieved with SVM using a cubic kernel, which was 98.2%.
- eFinally, in [10], a technique was proposed to detect and classify leaf diseases using artificial neural networks (ANN). The features were extracted using HSV, and the ANN classifier was employed to classify the input images, resulting in an accuracy of 80%, which is moderate. In another study [20], the authors aimed to identify leaf diseases using artificial neural networks and image processing. K-means clustering was used for segmentation, GLCM was employed for feature extraction, and the Back Propagation Neural Network (BPNN) was used for classification. Remarkably, the accuracy was found to be 100%.

3. Methodology

You Only Look Once (YOLO) is a revolutionary object detection model that addresses the challenge of real-time image analysis. Unlike traditional object detection methods that involve a multi-stage process, YOLO simplifies the task by framing it as a single regression problem. It divides the image into a grid and predicts bounding boxes and class probabilities for each grid cell simultaneously. YOLO models have evolved through various versions, each offering improvements in speed and accuracy. YOLOv8, the latest iteration, leverages advanced techniques in neural network architecture to enhance performance. It uses a convolutional neural network (CNN) to process images and employs anchor boxes to predict multiple bounding boxes per grid cell. This approach allows YOLO to detect objects quickly and accurately, making it suitable for applications requiring real-time analysis, such as plant disease detection.

3.1 Data Collection

To train and evaluate our plant disease detection model, we curated a comprehensive dataset of leaf images representing various plant conditions. The dataset includes images of leaves affected by five distinct types of plant diseases, as well as healthy leaves, to ensure a diverse and balanced training set. The breakdown of the dataset is as follows:

- Aphid Infestation: 100 images
- Bacterial Leaf Spot: 173 images
- Black Spot on Apple: 94 images
- Early Blight: 126 images

- Powdery Mildew on Grape: 33 images
- Healthy Leaves: 85 images

In total, the dataset comprises 611 images. This diverse collection was designed to cover a range of common plant diseases and healthy conditions to improve the model's ability to generalize across different scenarios. The images were collected from various sources, ensuring variability in lighting conditions, backgrounds, and leaf appearances.

3.2 Data Preprocessing

Data preprocessing is a crucial step in preparing raw images for model training. It involves cleaning and transforming the data to enhance model performance. For our study, we applied different preprocessing techniques to two datasets:

Dataset 1: No preprocessing techniques were applied to this dataset. It served as a baseline to evaluate the model's performance on raw, unaltered images.

Dataset 2: To enhance the quality and consistency of the images, we applied the following preprocessing techniques:
- Static Crop: Extracted 25-75% of the horizontal region and 5-75% of the vertical region of the images. This cropping helped to focus on relevant areas of the leaf while removing extraneous background.
- Grayscale Conversion: Converted images to grayscale to reduce the complexity of the data. This conversion helps the model to focus on structural features rather than color variations, which can be beneficial when color information is less relevant to the detection task.

3.3 Data Augmentation

Data augmentation techniques were employed to artificially expand the dataset and improve the model's robustness. Augmentation helps simulate variations that the model might encounter in real-world scenarios, thus enhancing its generalization capabilities. For one of the datasets, the following augmentation techniques were applied:
- **Flip:** Applied horizontal and vertical flips to create mirror images of the original leaf images. This technique helps the model learn to detect diseases from different orientations.
- **Rotation:** Rotated images by 90° clockwise, counter-clockwise, and upside down to simulate different viewing angles and perspectives.
- **Grayscale Application:** Applied to 19% of images to further diversify the dataset and assist the model in learning from both color and grayscale inputs.
- **Hue Adjustment:** Modified the hue between -180° and +180° to account for color variations and lighting conditions.
- **Brightness Adjustment:** Adjusted brightness levels between -49% and +49% to simulate different lighting environments.
- **Blur:** Introduced blur up to 15.8px to simulate out-of-focus or low-quality images.
- **Noise:** Added noise to up to 1.13% of pixels to mimic image artifacts and variations in real-world conditions.

3.4 Model Architecture

The YOLOv8 architecture is designed for high efficiency and accuracy in object detection tasks. The architecture operates as follows:
- **Image Resizing:** The input image is resized to 448x448 pixels to standardize the input size and ensure consistent processing across the network.
- **Convolutional Layers:** Initially, a 1x1 convolution is applied to reduce the number of channels, followed by a 3x3 convolution to generate a feature map with spatial and semantic information.
- **Activation Functions:** ReLU (Rectified Linear Unit) is used as the activation function in the intermediate layers, providing non-linearity and allowing the network to learn complex patterns. The final layer utilizes a linear activation function to produce continuous outputs for bounding box coordinates and class probabilities.
- **Regularization Techniques:** Batch normalization is applied to normalize activations and gradients, improving training stability and convergence. Dropout is used to prevent overfitting by randomly dropping units during training, which encourages the model to generalize better.

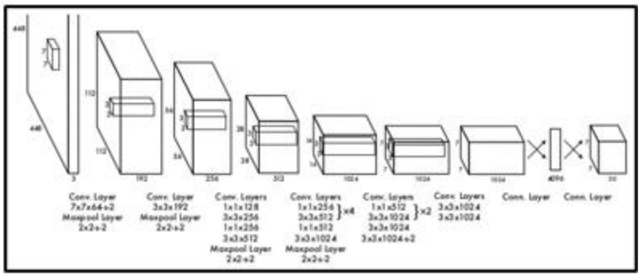

Figure 1: YOLO Model Architecture

3.5 Training Process:

The model training was conducted using JupyterLab with the YOLOv8n (nano) model. The YOLOv8n model is a lightweight variant of the YOLOv8 architecture, designed to deliver high performance with reduced computational requirements. It is well-suited for applications where resource constraints are a concern, such as real-time detection on edge devices.

The dataset was split into three parts:
- Training Set: 70% of the images were used for training the model. This large portion of the dataset allowed the model to learn from a diverse range of examples and improve its accuracy.
- Validation Set: 20% of the images were reserved for validation. This subset was used to tune hyperparameters and monitor the model's performance during training, helping to prevent overfitting.
- Test Set: 10% of the images were used for final testing. This set provided an unbiased evaluation of the model's performance on unseen data, ensuring that the results were reflective of its real-world applicability.

The training involved the following configurations:
- Libraries Used: Ultralytics (for YOLOv8 implementation), PyTorch (for deep learning framework), and Roboflow (for data labeling and management).

- Training Configuration:

```
model.train(
    data='leafs-dataset/data.yaml',
    epochs=150,
    save=True,
    batch=32,
)
```

3.6 Evaluation Metrics:

To evaluate the performance of the trained models, several key metrics were used:

- Intersection over Union (IoU): IoU quantifies the overlap between predicted and ground truth bounding boxes. It is essential for assessing the accuracy of object localization and determining how well the model identifies disease-affected areas.
- Average Precision (AP): AP calculates the area under the precision-recall curve for each class, providing a measure of the model's precision and recall performance.
- Mean Average Precision (mAP): mAP aggregates the AP values across all classes, offering a comprehensive evaluation of the model's performance in multi-class object detection tasks.
- Precision and Recall: Precision measures the proportion of true positives among all positive predictions, while recall assesses the proportion of true positives among all actual positives. These metrics help evaluate the model's ability to detect diseases accurately and consistently.
- F1 Score: The F1 Score is the harmonic mean of precision and recall, providing a balanced assessment that accounts for both false positives and false negatives.

4. Results

4.1 Model Performance on Raw Labeled Data:

The performance of the YOLOv8n model on raw labeled data reveals how well it can detect and classify plant diseases and healthy leaves based solely on unprocessed, original images.

a) Aphid Infestation: 64% accuracy
The model demonstrated moderate success in identifying aphid infestation. While it achieved a reasonable accuracy, there is room for improvement, particularly in distinguishing this condition from other similar-looking diseases or variations.
Bacterial Leaf Spot: 55% accuracy

The accuracy for detecting bacterial leaf spot was relatively low. This lower performance may be attributed to the variability in the appearance of the spots or similarities with other types of leaf damage, making it challenging for the model to differentiate effectively.

b) Black Spot on Apple: 86% accuracy
The model performed well in identifying black spot on apple leaves, achieving high accuracy. This suggests that the black spot disease has distinct features that the YOLOv8n model can recognize effectively, likely due to clear visual patterns or well-defined symptoms.

c) Early Blight: 53% accuracy
The model's accuracy for early blight was also on the lower side. The challenges here may involve the early stage of the disease, which might present less distinctive symptoms compared to more advanced stages or other diseases.

d) Powdery Mildew on Grape: 71% accuracy
The model showed moderate accuracy in detecting powdery mildew on grape leaves. While it performed better than some other conditions, the accuracy suggests that there may be overlapping features with other diseases or variability in the visual presentation of powdery mildew.

e) Healthy Leaves: 100% accuracy
The model achieved perfect accuracy in classifying healthy leaves. This indicates that healthy leaves are visually distinct and easily recognizable, allowing the model to perform flawlessly in this category.

Overall, while the model showed strong performance in certain categories like healthy leaves and black spot on apple, there is variability in accuracy across different disease types. This suggests that the raw labeled data alone might not fully capture the complexity and diversity of plant diseases, impacting the model's ability to generalize effectively.

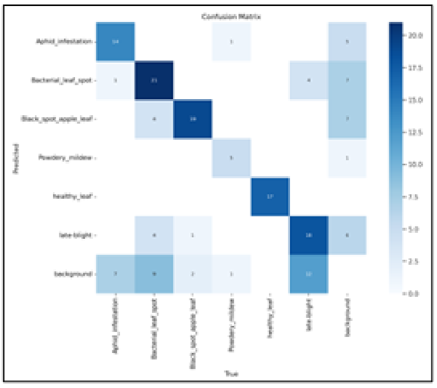

Figure 2: Confusion Matrix (Raw Dataset)

4.2 Model Performance on Augmented and Processed Labeled Data:

When the model was trained on augmented and preprocessed data, the performance metrics changed as follows:

a) Aphid Infestation: 68% accuracy
The model's accuracy improved with the augmented and processed data, indicating that data augmentation techniques such as rotation, flipping, and hue adjustment helped the model better recognize aphid infestation. The increase in accuracy suggests that these techniques made the disease features more distinct and varied, aiding the model's learning.

b) Bacterial Leaf Spot: 63% accuracy
Accuracy increased for bacterial leaf spot with the augmented and processed data. This improvement may be due to preprocessing techniques like static cropping and grayscale conversion, which helped the model focus on relevant features and reduce background noise, leading to better identification of bacterial leaf spot.

c) Black Spot on Apple: 64% accuracy
There was a decrease in accuracy for black spot on apple leaves with the augmented data. This might suggest that some augmentation techniques inadvertently introduced variations that made it harder for the model to recognize the disease consistently. It highlights the need for careful selection of augmentation methods to avoid compromising the model's ability to identify specific conditions.

d) Early Blight: 67% accuracy
The accuracy for early blight improved with the augmented and processed data. This suggests that the preprocessing and augmentation techniques helped highlight the disease's features better, making it easier for the model to learn and detect early blight effectively.

e) Powdery Mildew on Grape: 57% accuracy
The accuracy for powdery mildew on grape leaves decreased with the augmented data. This reduction might indicate that the augmentation techniques introduced variations that were not representative of the disease's typical appearance, leading to decreased model performance.

f) Healthy Leaves: 100% accuracy
The accuracy for healthy leaves remained unchanged at 100%. This consistency indicates that the preprocessing and augmentation methods did not affect the model's ability to correctly classify healthy leaves, which are likely easily distinguishable from diseased conditions.

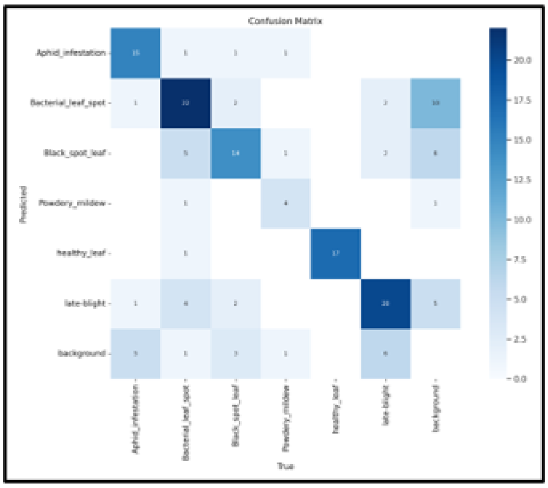

Figure 3: Confusion Matrix (Augmented and Processed Dataset)

5. Comparative Analysis

5.1 Analysis of Performance Differences

The performance differences between the models trained on raw labeled data and those trained on augmented and processed labeled data can be attributed to several factors:

a) Feature Representation and Complexity:
Raw Labeled Data: The model's performance on raw data often reflects its ability to learn and generalize from the original, unaltered images. Variations in lighting, backgrounds, and leaf appearances can make it challenging for the model to identify subtle disease features, leading to lower accuracy in some categories.

Augmented and Processed Data: Data augmentation techniques, such as flipping, rotating, and adjusting hue, help the model learn from a broader range of variations. These methods enhance the model's ability to recognize diseases under different conditions, often leading to improved accuracy. However, if the augmentations introduce unrealistic variations, they may also confuse the model, leading to decreased performance in certain cases.

b) Disease Specificity:
Aphid Infestation and Early Blight: The improvements in accuracy for these diseases with augmented data suggest that the model benefits from enhanced feature diversity. Augmentation likely made it easier for the model to recognize these diseases by providing varied examples. However, the complexity of disease features and their similarity to other conditions might still pose challenges.

Black Spot on Apple and Powdery Mildew on Grape: The decrease in accuracy for black spot on apple and powdery mildew on grape when using augmented data could be due to the specific characteristics of these diseases being altered or distorted by the augmentation techniques, making them harder for the model to identify consistently.

c) Preprocessing Techniques:
Static Crop and Grayscale Conversion: The preprocessing techniques applied to one of the datasets helped improve the model's ability to focus on relevant features and reduce noise. This likely contributed to better performance for some diseases, such as bacterial leaf spot, by highlighting disease-specific patterns and reducing background variability.

5.2 Impact of Data Augmentation on Model Generalization and Accuracy

Data augmentation generally helps improve model generalization by exposing it to a wider range of variations and scenarios. This allows the model to become more robust to different conditions and enhances its ability to generalize from the training data to real-world situations. The benefits of augmentation are evident in the increased accuracy for diseases like aphid infestation and bacterial leaf spot.

However, excessive or poorly chosen augmentation techniques can introduce distortions that do not accurately represent real-world variations. For instance, unrealistic rotations or hue adjustments might lead to reduced performance if the model encounters variations that are not representative of the actual disease features. Thus, while data augmentation is a powerful tool for improving generalization, it requires careful implementation to balance the benefits and avoid introducing misleading variations.

6. Discussion on Overfitting/Underfitting

6.1 Overfitting:

Signs of Overfitting: Overfitting occurs when the model performs exceptionally well on the training data but poorly on unseen validation or test data. In this case, if the model trained on raw labeled data shows significantly higher accuracy on the training set but lower accuracy on the validation or test sets, it may indicate overfitting. The model might have learned to memorize specific details of the training images rather than generalizing the disease features.

Detection: Overfitting can be detected through a high discrepancy between training and validation/test accuracies. If the model's performance on the validation/test sets is significantly worse compared to the training set, it suggests overfitting.

6.2 Underfitting:

Signs of Underfitting: Underfitting occurs when the model fails to capture the underlying patterns in the data, leading to poor performance on both training and validation/test sets. If the model trained on augmented data consistently shows lower accuracy across all datasets, it might indicate underfitting. The model may not be complex enough to learn the disease features or may have been trained on augmented data that did not effectively represent real-world variations.

Detection: Underfitting can be identified by consistently low accuracy across training, validation, and test sets. If the model struggles to perform well on any subset of data, it suggests that it is not learning the essential features of the diseases.

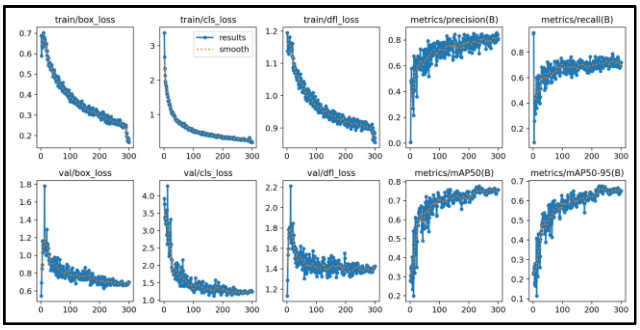

Figure 4: Analysis Results (Raw Dataset)

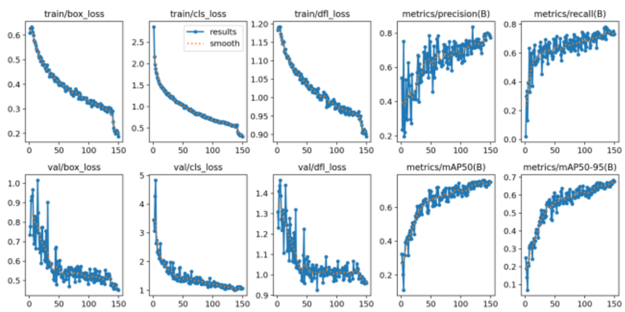

Figure 5: Analysis Results (Augmented and Processed Dataset)

7. Comparative Analysis

7.1 Key Findings

The YOLOv8 model has demonstrated substantial potential for plant disease detection using leaf images. The model's ability to effectively identify various plant diseases and healthy leaves underscores its applicability in agricultural settings. The comparative analysis of performance on raw labeled data versus augmented and processed labeled data revealed important insights:

- Effectiveness: The YOLOv8 model generally showed strong performance in identifying distinct conditions such as healthy leaves and black spot on apple. Data augmentation and preprocessing improved the model's accuracy for several diseases, including aphid infestation and bacterial leaf spot, by diversifying the training data and enhancing feature representation.

- Comparative Results: The analysis highlighted both improvements and challenges associated with data augmentation. While augmentation techniques often led to better performance in recognizing certain diseases, they also introduced variations that affected the accuracy for some conditions. This underscores the need for careful implementation of augmentation strategies to optimize model performance.

8. Limitations

Several limitations were identified in the study:

- Dataset Size and Diversity: The relatively small size of the dataset and the limited number of images for certain diseases, such as powdery mildew on grape, may have constrained the model's ability to generalize effectively. A larger and more diverse dataset could enhance the model's accuracy and robustness.
- Augmentation Effects: While data augmentation generally improved performance, some techniques led to decreased accuracy for specific diseases. This indicates that not all augmentations are equally beneficial and that careful selection is crucial to avoid introducing misleading variations.
- Model Complexity: The YOLOv8n (nano) model, being a lightweight variant, may have limitations in capturing complex features compared to larger models. This could impact the model's ability to detect subtle or early-stage symptoms of plant diseases.

9. Results

Detection Images: The results of the model's performance on raw labeled data and augmented data can be visualized through the detection images provided. These images illustrate the model's ability to identify and classify different diseases and healthy leaves accurately, highlighting both successful detections and areas where improvements are needed.

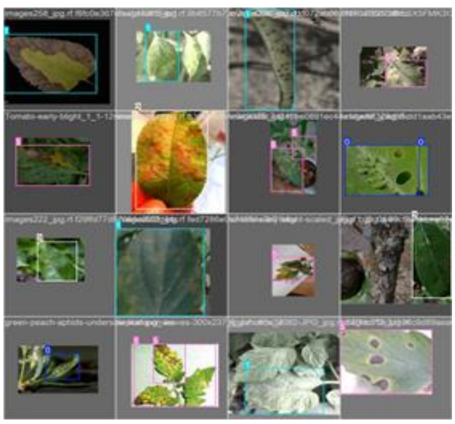

Figure 6: Detection Results (Raw Dataset)

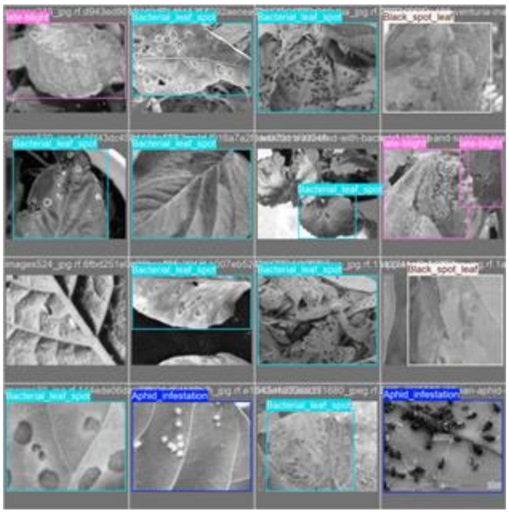

Figure 7: Detection Results (Augmented and Processed Dataset)

10. Future Work

To enhance the model and extend the research, several directions are suggested:

- Dataset Expansion: Increasing the size and diversity of the dataset by including more images of each disease and healthy leaves can improve the model's generalization and accuracy. Collecting data from different sources and conditions will help the model learn from a broader range of variations.
- Enhanced Data Augmentation: Refining the data augmentation techniques to better simulate realistic variations without introducing excessive noise or distortions can further improve model performance. Experimenting with different augmentation strategies and their impact on specific diseases will be beneficial.
- Model Optimization: Exploring more complex variants of YOLO or alternative architectures may yield better results, especially for detecting subtle or early-stage symptoms. Experimenting with different model configurations and hyperparameters can help find the optimal balance between accuracy and computational efficiency.
- Expansion to Other Diseases: Extending the research to include additional types of plant diseases and pest infestations will broaden the model's applicability. Incorporating a wider range of conditions can provide a more comprehensive tool for plant health monitoring.
- Integration of Multimodal Data: Combining image data with other types of information, such as environmental conditions or sensor data, could enhance the model's ability to diagnose plant health issues more accurately and holistically.

References

[1] https://www.researchgate.net/publication/361064240_Challenges_Faced_by_Farmers_in_Crops_Production_Due_to_Fungal_Pathogens_and_Their_Effect_on_Indian_Economy

[2] https://www.safefood.net/food-safety/news/impact-plant-diseases

[3] K. Rajesh Babu, "Plant Disease Identification and Classification using Image Processing Depth Resolution in Non Stationary Thermal Wave Imaging Using Wavelet Transform View project," 2019. [Online]. Available: https://www.researchgate.net/publication/337023535

[4] S. Raut, A. Fulsunge, and P. G. Student, "Plant Disease Detection inImage Processing Using MATLAB," International Journal of Innovative Research in Science, Engineering and Technology (An ISO, vol. 3297, 2007, doi: 10.15680/IJIRSET.2017.0606034.

[5] S. Arivazhagan, R. N. Shebiah, S. Ananthi, and S. V. Varthini, "Detection of unhealthy region of plant leaves and classification of plant leaf diseases using texture features," 2013. [Online]. Available: http://www.cigrjournal.org

[6] M. Bhange and H. A. Hingoliwala, "Smart Farming: Pomegranate Disease Detection Using Image Processing," in Procedia Computer Science, 2015, vol. 58, pp. 280–288. doi: 10.1016/j.procs.2015.08.022.

[7] S. Pavithra, A. Priyadharshini, V. Praveena, and T. Monika, "Paddy Leaf Disease Detection Using Svm Classifier," International Journal of communication and computer Technologies, vol. 3, no. 1, Jan. 2019, doi: 10.31838/ijccts/03.01.04.

[8] Y. M. Oo and N. C. Htun, "Plant Leaf Disease Detection and Classification using Image Processing," International Journal of Research and Engineering, vol. 5, no. 9, pp. 516–523, Nov. 2018, doi: 10.21276/ijre.2018.5.9.4.

[9] R. P. Narmadha and G. Arulvadivu, "Detection and measurement of paddy leaf disease symptoms using image processing," in 2017 International Conference on Computer Communication and Informatics (ICCCI), Jan. 2017, pp. 1–4. doi: 10.1109/ICCCI.2017.8117730.

[10] M. Ranjan, M. Rajiv Weginwar, N. Joshi, and A. Ingole, "Detection and Classification of Leaf Disease Using Artificial Neural Network," 2015. [Online]. Available: www.ijtra.com

[11] A. Parikh, M. S. Raval, C. Parmar, and S. Chaudhary, "Disease detection and severity estimation in cotton plant from unconstrained images," in Proceedings - 3rd IEEE International Conference on Data Science and Advanced Analytics, DSAA 2016, Dec. 2016, pp. 594–601. doi: 10.1109/DSAA.2016.81.

[12] M. A. Jasim and J. M. AL-Tuwaijari, "Plant Leaf Diseases Detection and Classification Using Image Processing and Deep Learning Techniques," in 2020 International Conference on Computer Science and Software Engineering (CSASE), Apr. 2020, pp. 259–265. doi: 10.1109/CSASE48920.2020.9142097.

[13] S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, and D. Stefanovic, "Deep Neural Networks Based Recognition of Plant Diseases by Leaf Image Classification," Comput Intell Neurosci, vol. 2016, pp. 1–11, 2016, doi: 10.1155/2016/3289801.

[14] A. Devaraj, K. Rathan, S. Jaahnavi, and K. Indira, "Identification of Plant Disease using Image Processing Technique," in 2019 International Conference on Communication and Signal Processing (ICCSP), Apr. 2019, pp. 0749–0753. doi: 10.1109/ICCSP.2019.8698056.

[15] Mr. A. N. Patil and Miss. V. Pawar, "Detection and Classification of Plant Leaf Disease," IARJSET, vol. 4, no. 4, pp. 72–75, Jan. 2017, doi: 10.17148/IARJSET/NCIARCSE.2017.20.
[16] S. D.M., Akhilesh, S. A. Kumar, R. M.G., and P. C., "Image based Plant Disease Detection in Pomegranate Plant for Bacterial Blight," in 2019 International Conference on Communication and Signal Processing (ICCSP), Apr. 2019, pp. 0645–0649. doi: 10.1109/ICCSP.2019.8698007.
[17] S. B. Patil and S. K. Bodhe, "Leaf Disease Severity Measurement Using Image Processing," 2011.
[18] V. SBhong, "Study and Analysis of Cotton Leaf Disease Detection Using Image Processing," Int J Adv Res Sci Eng Technol, vol. 3, no. 2, 2016, [Online]. Available: www.ijarset.com
[19] IEEE Control Systems Society, IEEE Control Systems Society. Chapter Malaysia., and Institute of Electrical and Electronics Engineers, Orchid Leaf Disease Detection using Border Segmentation Techniques. 2014.
[20] S. S. Sannakki, V. S. Rajpurohit, V. B. Nargund, and P. Kulkarni, "Diagnosis and classification of grape leaf diseases using neural networks" in 2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT), Jul. 2013, pp. 1–5. doi: 10.1109/ICCCNT.2013.6726616.