Detection of tomato leaf diseases

Miloš Zeljko
Faculty of tehnical sciences
University of Novi Sad
Trg Dositeja Obradovića 6
21000 Novi Sad
miloszeljko00@gmail.com

This research addresses the early detection of tomato leaf diseases using convolutional neural networks (CNNs). A unique dataset combining controlled laboratory (PlantVillage) and real-world images was utilized to improve model robustness in practical settings. The combined dataset consists of 10 balanced tomato leaf disease classes for model training. The real-world image dataset includes 200 images with natural variations (lighting, background, disease severity), collected and labeled by the researcher. Several CNN architectures (ToLeD-inspired, VGG-19, InceptionV3, Xception, ResNet152V2) were evaluated. ResNet152V2 architecture demonstrated superior performance, achieving an accuracy of 93.50% on the PlantVillage dataset, likely due to its depth and effective use of ImageNet pre-training. However, a significant decrease to 70% accuracy was observed on the real-world dataset. Analysis revealed that complex backgrounds in real-world images were a primary factor leading to misclassifications. This research highlights the need for diverse datasets to bridge this performance gap. This work contributes to the development of practical AI driven tools for tomato disease detection, aiding farmers in timely interventions.

Keywords—tomato; disease detection; convolutional neural networks; PlantVillage; image classification

I. Introduction

Tomato crops play a vital role in global agriculture, providing essential nutrients and a source of income for farmers worldwide [1]. However, tomato plants are susceptible to a wide range of diseases that can significantly impact yield and quality. Early and accurate detection of these diseases is crucial for implementing effective control measures and minimizing losses [2]. Traditional methods of disease diagnosis often rely on visual inspection by experts, which can be time-consuming, subjective, and prone to error, especially in large-scale production systems. [3]

Recent advancements in computer vision and deep learning, particularly convolutional neural networks (CNNs), have opened up new possibilities for automated and reliable plant disease detection from images [4]. CNNs are exceptionally skilled at image analysis tasks, and capable of extracting complex features and patterns that are difficult for humans to recognize [5]. The application of CNNs in agriculture has the potential to revolutionize crop disease management, enabling faster and more precise disease identification [6].

This research aims to develop a CNN-based system for detecting various tomato leaf diseases. The proposed system utilizes a dataset combining controlled laboratory images and real-world field images to enhance its robustness in practical settings. We investigate the performance of several CNN architectures, including a custom ToLeD-inspired [7] model and pre-trained models like VGG-19 [8], InceptionV3 [9], Xception [10], and ResNet152V2 [11]. This research aims to provide tomato growers with a valuable tool for early disease detection, facilitating timely interventions to protect crop health.

II. Related research

This chapter explores the existing body of research on the application of CNNs for automated tomato leaf disease detection. It highlights key datasets, architectural choices, and the significance of real-world image inclusion, paving the way for a discussion of the present research contribution.

A. CNNs for Tomato Disease Identification

Several studies have successfully used CNNs to classify tomato leaf diseases. Agarwal et al. (2019) [7] employed a custom CNN on the PlantVillage [12] dataset, achieving an average accuracy of 91.2%, demonstrating the efficacy of this approach. Similarly, Ahmad et al. (2020) [13] investigated various CNN architectures for tomato disease detection, providing insights into their suitability.

B. Datasets and Real-World Images

The importance of including real-world images alongside laboratory-sourced datasets is well-recognized for improving model robustness and practical applicability. Chohan et al. (2020) [14] solved this by supplementing the PlantVillage dataset with field-captured images, leading to better performance in real-world settings. Ahmad et al. (2020) [13] also collected a smaller set of field images for testing purposes, further confirming the importance of real-world images in the dataset.

C. Architectural Comparisons

Performance comparison across various CNN architectures offers valuable guidance for model selection. Chohan et al. (2020) [14] found that InceptionV3 outperformed simpler CNNs and VGG architectures for tomato and other plant

disease detection. Ahmad et al. (2020) [13] similarly explored VGG, ResNet, and InceptionV3, offering a broader comparison with the same conclusion that InceptionV3 outperforms other architectures.

D. Addressing Gaps and Current Research

While the surveyed research highlights the potential of CNNs for tomato leaf disease detection, there remains scope for improvement in the following areas:

1) Robustness on Real-World Images

Existing datasets often contain a higher proportion of controlled laboratory images compared to challenging field-captured ones.

The present research addresses this gap by utilizing a dataset with an emphasis on real-world images to increase model generalizability to practical scenarios.

2) Architectural Exploration

Further investigation into less commonly studied architectures may yield promising results.

The present research addresses this gap by expanding the architectural comparison to include the Xception architecture.

III. DATASET

This dataset was constructed to provide a balanced foundation for training and evaluating a CNN model for tomato leaf disease detection. It strategically combines the advantages of a curated laboratory dataset with the challenges of practical image variability.

A. Dataset Sources

A subset of 11,000 images was extracted from the PlantVillage dataset (TomatoLeaf subset [15]). To ensure a balanced representation of 10 tomato disease classes, 1,100 images were included per class. This provides a strong initial base of accurately labeled images for model training. An example of images from this dataset is displayed in Figure 1.

To enhance model robustness, 200 real-world images (20 per class) were sourced from online platforms and carefully labeled by the researcher. This edition introduces the challenges of diverse backgrounds, lighting conditions, and natural variations that the model is likely to encounter in real-world applications. An example of images from this dataset is displayed in Figure 2.

B. Data Preprocessing

All images were resized to a uniform resolution of 256x256 pixels for computational efficiency and to ensure a standardized input format for the CNN.

Data augmentation techniques were applied to expand the dataset, minimizing overfitting and improving generalization. These techniques included rescaling, shearing, zooming, rotation, horizontal and vertical shifts, brightness adjustments, channel shifting, vertical flipping, and filling with the nearest pixel value.

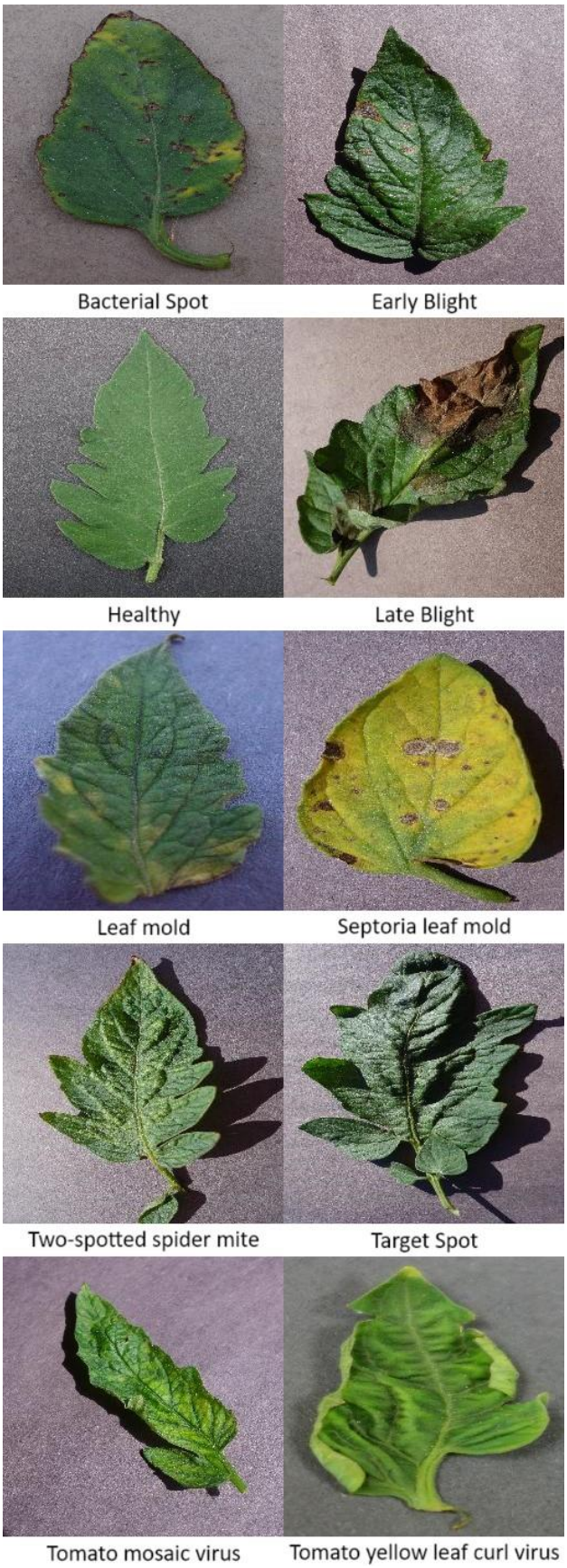

Figure 1 – Example of PlantVillage dataset

Figure 2 - Example of real-world images dataset

C. Data Split

The combined dataset was randomly split into the following proportions:

- Training: 50%
- Validation: 30%
- Testing: 20%

This division of dataset aims to provide sufficient data for training, ongoing model tuning based on validation performance, and a final unbiased performance assessment on the test set.

D. Class Distribution

The dataset maintains a balanced class distribution with 1,120 images per class (including both PlantVillage and real-world images) across 10 classes.

IV. Methodology

This research investigates the performance of several CNN architectures for automated tomato leaf disease detection. The efficacy of these architectures is assessed through an extensive comparison of accuracy and F_{macro} scores.

A. CNN Architectures

The following CNN architectures were explored: ToLeD-inspired, InceptionV3, Xception, VGG-19, and ResNet152V2.

1) ToLeD-inspired Architecture

Employed a custom CNN architecture inspired by the ToLeD network proposed in a paper by Agarwal et al. (2019) [8] and displayed in Figure 1.

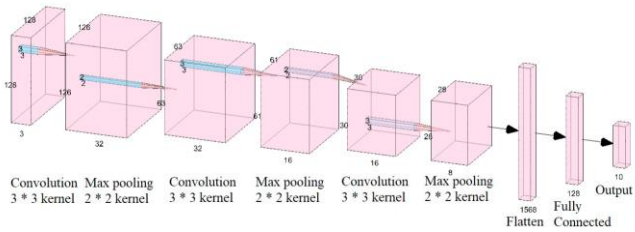

Figure 3 – ToLeD architecture

ToLeD-inspired architecture consists of the following layers:

- Three convolutional layers with 32, 16, and 8 filters respectively, each using a 3x3 kernel size and ReLU activation.
- Three max-pooling layers with a 2x2 pool size following each convolutional layer.
- One fully connected hidden layer with 128 neurons and ReLU activation.
- Regularization: Dropout (rate = 0.5) applied in the hidden layer to prevent overfitting.
- Output layer with 10 neurons and softmax activation.

2) InceptionV3 Architecture

A pre-trained model known for its efficient use of parameters and multi-scale feature extraction capabilities [16]. InceptionV3 architecture is displayed in Figure 4.

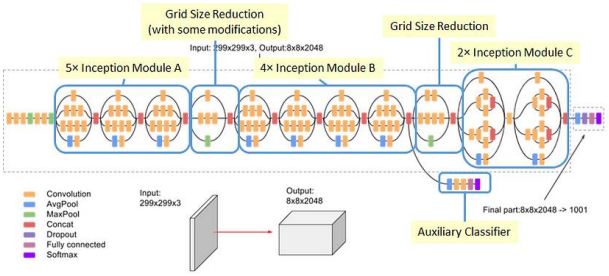

Figure 4 - InceptionV3 architecture

3) Xception Architecture

A pre-trained model building upon Inception, focusing on deep separable convolutions for improved parameter efficiency [17]. Xception architecture is displayed in Figure 5.

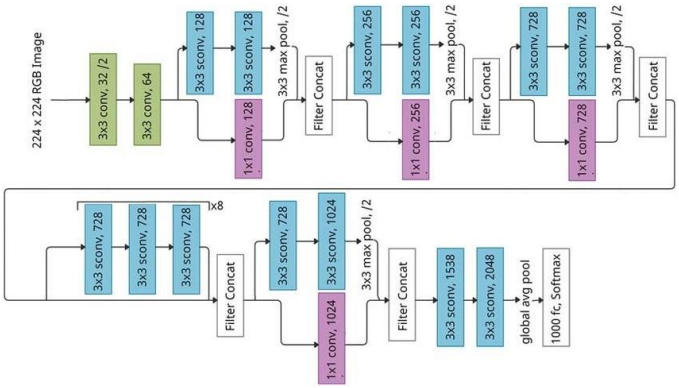

Figure 5 - Xception architecture

4) VGG-19 Architecture

A pre-trained model known for its depth and use of small (3x3) convolutional filters throughout the architecture [18]. VGG-19 architecture is displayed in Figure 6.

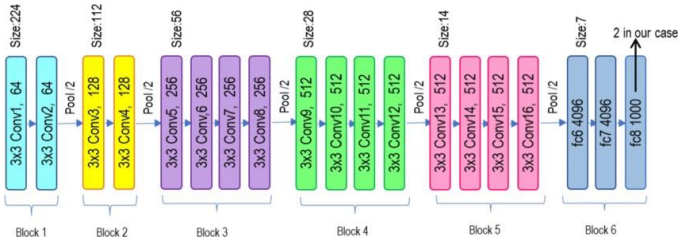

Figure 6 - VGG-19 architecture

5) ResNet152V2 Architecture

A pre-trained model with deep architecture and residual connections to combat the vanishing gradient problem [19]. ResNet152V2 architecture is displayed in Figure 7.

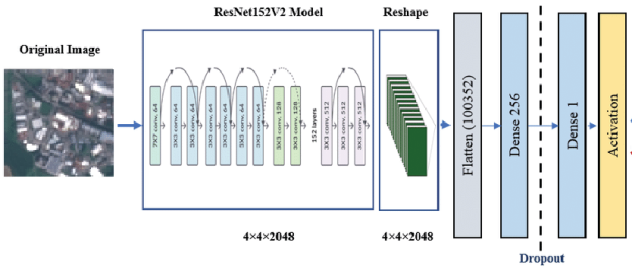

Figure 7 - ResNet152V2 architecture

B. Fine-Tuning

All pre-trained models were fine-tuned for the tomato leaf disease classification task. The following fine-tuning strategies were employed:

- The pre-trained models (weights from ImageNet [20]) were used as a base, excluding the original top layers.
- Initial pre-trained layers were frozen (weights not updated during training).
- Custom layers were added to adapt the model to the tomato leaf disease task:
  1. Average pooling to reduce spatial dimensions.
  2. Flatten layer to prepare the features for dense layers.
  3. Dense layer (256 neurons, ReLU activation) for intermediate processing.
  4. Dropout (rate = 0.5) to prevent overfitting.
  5. Output layer (10 neurons, softmax activation) for final classification.

Initial experiments revealed that the ResNet152V2 architecture achieved the most promising results even without extensive parameter optimization. Subsequent efforts were focused on further refinement of this model.

C. Model Training

1) Optimizer

Adam [21] - This optimizer is a popular choice for deep learning due to its efficient updates and ability to handle sparse gradients.

2) Loss Function

Categorical Cross-entropy [22] - This function is suitable for multi-class classification problems.

3) Hyperparameters

Employed a ReduceLROnPlateau [23] learning rate reduction scheduler to adjust the learning rate during training. This scheduler was configured to monitor validation loss, reducing the learning rate with the patience of 10 consecutive epochs without improvements in validation loss, by a factor of 0,5. The minimum learning rate was set to 1e-6 in order to prevent the learning rate from becoming too small and hindering the training process.

A batch size of 128 was selected after preliminary experimentation. This value demonstrated a good compromise between training speed, convergence, and final model performance.

To prevent overfitting and ensure optimal model generalization, early stopping was implemented with patience of 30 epochs (monitoring validation loss). This halts training when validation performance stops improving for 30 consecutive epochs and restores the model weights to their best state. A maximum of 1000 epochs was set to allow sufficient training time for the early stopping mechanism to take effect, typically triggering around epoch 150. This approach helps avoid unnecessarily long training cycles while achieving the best possible performance on unseen data.

4) Evaluation Metrics

Evaluation metrics used to evaluate trained models are accuracy [24] and F_{macro} [25] score.

a) Accuracy

Accuracy is the percentage of correctly classified images. It is calculated by formula 1, where TP represents a number of true positives, TN represents a number of true negatives, FP represents a number of false positives and a FN represents a number of false positives.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

Formula 1 - Accuracy

b) F_{macro}

F_{macro} is the harmonic mean of precision and recall, averaged across all classes. It is calculated by formula 2, where F_{1i} represents F_1 [26] score on a single class.

$$F_{macro} = \frac{1}{N}\sum_{i=1}^{n} F_{1i}$$

Formula 2 - F_{macro}

F_1 is the harmonic mean of precision and recall [27] on a single class. It is calculated by formula 3, where precision represents the ratio of correctly classified positive samples (true positive) to a total number of classified positive samples and recall represents a score that measures how often a model correctly identifies positive instances (true positives) from all the actual positive samples in the dataset.

$$F_1 = 2 \times \frac{precision \times recall}{precision \times recall}$$

Formula 3 – F_1

Precision is calculated by formula 4, where TP represents a number of true positives and FP represents a number of false positives.

$$precision = \frac{TP}{TP + FP}$$

Formula 4 - Precision

Recall is calculated by formula 5, where TP represents a number of true positives and FN represents a number of false negatives.

$$recall = \frac{TP}{TP + FN}$$

Formula 5 - Recall

V. Results

This section presents the experimental results obtained using the different CNN architectures for tomato leaf disease classification. Performance was evaluated using accuracy and F_1 scores explained earlier in the text.

A. Extensive Results Overview

Table 1 displays the results of ToLeD-inspired architecture on lab images from the PlantVillage dataset, while Table 2 displays the results on real-world images.

Class	Precision	Recall	F_1
Bacterial spot	0.91	0.91	0.91
Early blight	0.82	0.66	0.74
Late blight	0.90	0.77	0.83
Leaf mold	0.95	0.81	0.88
Septoria leaf mold	0.87	0.61	0.72
Two-spotted spider mites	0.64	0.79	0.71
Target spot	0.46	0.77	0.57
Yellow leaf curl virus	0.98	0.80	0.88
Mosaic virus	0.93	0.62	0.74
Healthy	0.72	1.00	0.84
	Accuracy	77.45 %	
	F_{macro}	0.7813	

Table 1 – ToLeD-inspired architecture results on PlantVillage dataset images

Class	Precision	Recall	F_1
Bacterial spot	1.00	0.25	0.40
Early blight	0.12	0.25	0.17
Late blight	0.33	0.50	0.40
Leaf mold	0.00	0.00	0.00
Septoria leaf mold	0.38	0.75	0.50
Two-spotted spider mites	0.33	0.25	0.29
Target spot	0.33	0.25	0.29
Yellow leaf curl virus	0.50	0.25	0.33
Mosaic virus	0.43	0.75	0.55
Healthy	0.50	0.25	0.33
	Accuracy	35.00 %	
	F_{macro}	0.3250	

Table 2 – ToLeD-inspired architecture results on real-world dataset images

Table 3 displays the results of InceptionV3 architecture on lab images from the PlantVillage dataset, while Table 4 displays the results on real-world images.

Class	Precision	Recall	F_1
Bacterial spot	0.90	0.90	0.90
Early blight	0.84	0.81	0.82
Late blight	0.84	0.86	0.85
Leaf mold	0.87	0.90	0.88
Septoria leaf mold	0.87	0.87	0.87
Two-spotted spider mites	0.89	0.80	0.84
Target spot	0.75	0.85	0.80
Yellow leaf curl virus	0.96	0.95	0.96
Mosaic virus	0.99	0.96	0.98
Healthy	0.94	0.89	0.91
	Accuracy	88.05 %	
	F_{macro}	0.8810	

Table 3 – InceptionV3 architecture results on PlantVillage dataset images

Class	Precision	Recall	F_1
Bacterial spot	0.50	0.25	0.33
Early blight	0.33	0.75	0.46
Late blight	0.50	0.50	0.50
Leaf mold	1.00	0.50	0.67
Septoria leaf mold	0.50	0.25	0.33
Two-spotted spider mites	1.00	1.00	1.00
Target spot	0.43	0.75	0.55
Yellow leaf curl virus	0.75	0.75	0.75
Mosaic virus	1.00	0.75	0.86
Healthy	1.00	0.75	0.86
	Accuracy	62.50 %	
	F_{macro}	0.6305	

Table 4 – InceptionV3 architecture results on real-world dataset images

Table 5 displays the results of Xception architecture on lab images from the PlantVillage dataset, while Table 6 displays the results on real-world images.

Class	Precision	Recall	F_1
Bacterial spot	0.91	0.90	0.91
Early blight	0.88	0.70	0.78
Late blight	0.83	0.88	0.86
Leaf mold	0.88	0.90	0.89
Septoria leaf mold	0.89	0.87	0.88
Two-spotted spider mites	0.81	0.86	0.83
Target spot	0.76	0.84	0.79
Yellow leaf curl virus	0.99	0.95	0.97
Mosaic virus	0.93	0.96	0.95
Healthy	0.91	0.91	0.91
	Accuracy	87.68 %	
	F_{macro}	0.8764	

Table 5 – Xception architecture results on PlantVillage dataset images

Class	Precision	Recall	F_1
Bacterial spot	0.43	0.75	0.55
Early blight	0.60	0.75	0.67
Late blight	0.33	0.25	0.29
Leaf mold	0.75	0.75	0.75
Septoria leaf mold	0.00	0.00	0.00
Two-spotted spider mites	0.75	0.75	0.75
Target spot	0.60	0.75	0.67
Yellow leaf curl virus	0.67	0.50	0.57
Mosaic virus	0.60	0.75	0.67
Healthy	1.00	0.75	0.86
	Accuracy	60.00 %	
	F_{macro}	0.5760	

Table 6 – Xception architecture results on real-world dataset images

Table 7 displays the results of VGG-19 architecture on lab images from the PlantVillage dataset, while Table 8 displays the results on real-world images.

Class	Precision	Recall	F_1
Bacterial spot	0.82	0.93	0.87
Early blight	0.80	0.67	0.73
Late blight	0.83	0.66	0.73
Leaf mold	0.89	0.80	0.84
Septoria leaf mold	0.71	0.80	0.75
Two-spotted spider mites	0.72	0.89	0.79
Target spot	0.79	0.70	0.74
Yellow leaf curl virus	0.96	0.90	0.93
Mosaic virus	0.89	1.00	0.94
Healthy	0.91	0.96	0.93
	Accuracy	82.91 %	
	F_{macro}	0.8267	

Table 7 – VGG-19 architecture results on PlantVillage dataset images

Class	Precision	Recall	F_1
Bacterial spot	0.25	0.25	0.25
Early blight	0.18	0.50	0.27
Late blight	0.20	0.25	0.22
Leaf mold	0.00	0.00	0.00
Septoria leaf mold	1.00	0.25	0.40
Two-spotted spider mites	0.50	0.25	0.33
Target spot	0.14	0.25	0.18
Yellow leaf curl virus	0.38	0.75	0.50
Mosaic virus	1.00	0.50	0.67
Healthy	0.00	0.00	0.00
	Accuracy	30.00 %	
	F_{macro}	0.2821	

Table 8 – VGG-19 architecture results on real-world dataset images

Table 9 displays the results of ResNet152V2 architecture on lab images from the PlantVillage dataset, while Table 10 displays the results on real-world images.

Class	Precision	Recall	F_1
Bacterial spot	0.96	0.95	0.95
Early blight	0.93	0.86	0.89
Late blight	0.92	0.96	0.94
Leaf mold	0.98	0.93	0.93
Septoria leaf mold	0.91	0.92	0.91
Two-spotted spider mites	0.92	0.90	0.91
Target spot	0.84	0.90	0.87
Yellow leaf curl virus	0.99	0.98	0.99
Mosaic virus	0.97	0.99	0.98
Healthy	0.94	0.97	0.96
	Accuracy	93.50 %	
	F_{macro}	0.9351	

Table 9 – ResNet152V2 architecture results on PlantVillage dataset images

Class	Precision	Recall	F_1
Bacterial spot	0.80	1.00	0.89
Early blight	0.60	0.75	0.67
Late blight	1.00	0.25	0.40
Leaf mold	0.50	0.50	0.50
Septoria leaf mold	0.67	0.50	0.57
Two-spotted spider mites	0.50	0.50	0.50
Target spot	0.67	1.00	0.80
Yellow leaf curl virus	0.75	0.75	0.75
Mosaic virus	0.80	1.00	0.89
Healthy	1.00	0.75	0.86
	Accuracy	70.00 %	
	F_{macro}	0.6823	

Table 10 – ResNet152V2 architecture results on real-world dataset images

B. Results Comparison

Table 11 presents a performance comparison of all tested architectures based on their accuracy and F_{macro} scores on PlantVillage dataset images, while Table 12 presents a performance comparison based on accuracy and F_{macro} scores on real-world dataset images.

Architecture	Accuracy	F_{macro}
ToLeD-inspired	77.45 %	0.7813
VGG-19	82.91 %	0.8267
Xception	87.68 %	0.8764
InceptionV3	88.05 %	0.8810
ResNet152V2	93.50 %	0.9351

Table 11 - Comparison of all tested architectures on PlantVillage dataset images

Architecture	Accuracy	F_{macro}
VGG-19	30.00 %	0.2821
ToLeD-inspired	35.00 %	0.3250
Xception	60.00 %	0.5760
InceptionV3	62.50 %	0.6305
ResNet152V2	70.00 %	0.6823

Table 12 - Comparison of all tested architectures on real-world dataset images

On PlantVillage images, ResNet152V2 demonstrates the best performance with an accuracy of 93.50% and a F_{macro} score of 0.9351. This is followed by InceptionV3, Xception, VGG-19, and the ToLeD-inspired architecture.

However, a significant decline in performance is observed across all architectures when tested on the real-world dataset. Notably, ResNet152V2's accuracy drops to 70%, and its F_{macro} score decreases to 0.6823. This substantial decrease of 23,5% in accuracy and 0,2528 in F_{macro} score is visualized in Figure 8 and Figure 9.

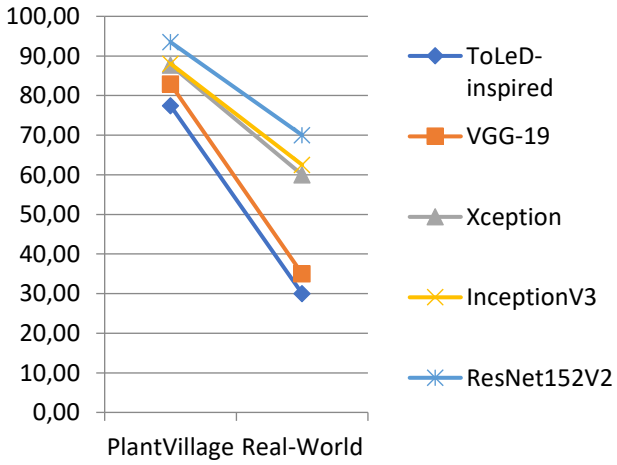

Figure 8 - Accuracy comparison between PlantVillage and real-world dataset images

Figure 9 - F_{macro} comparison between PlantVillage and real-world dataset images

C. Key findings

The severe contrast in performance between the two datasets reveals a gap between the curated datasets like PlantVillage and the complexities of real-world images. This highlights the limitations of models trained exclusively on laboratory datasets and emphasizes the need for more diverse and realistic training data.

ResNet152V2's consistent superiority, even with the increased challenges of the real-world dataset, demonstrates the advantages of very deep architectures and the effectiveness of transfer learning from large-scale image datasets like ImageNet.

The performance decline on real-world images suggests that factors such as background noise, variations in lighting and disease severity, and potentially incorrect image labeling pose significant challenges for the models.

D. Grad-CAM Visualizations

To better understand why the performance declined by such a big margin it will be useful to take a look at the Grad-CAM visualizations [28] of some of the misclassified images and try to analyze them.

Grad-CAM computes gradients of predicted class scores concerning the activations in the last convolutional layer. These gradients signify the importance of each activation map for predicting specific classes.

Figures 10, 11, and 12 confirm that background noise is indeed one of the root causes of the performance decline on real-world data, compared to the PlantVillage dataset.

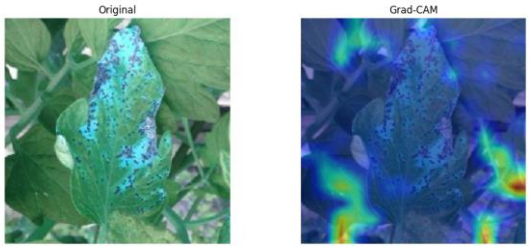

Figure 10 - Example 1 of background noise

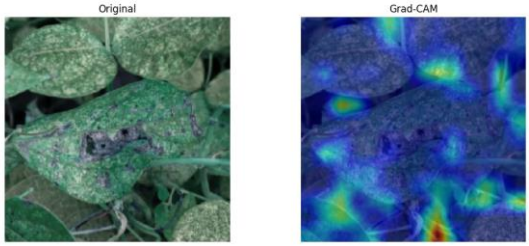

Figure 11 - Example 2 of background noise

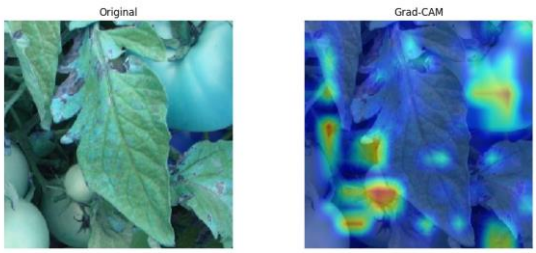

Figure 12 - Example 3 of background noise

Figure 13 and Figure 14 show that leaf mold is commonly misclassified as spider mites' disease. The reason for this misclassification could potentially be the grayish look on both leaf mold and spider mites' diseased leaves.

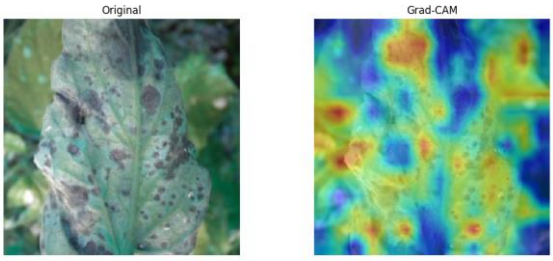

Figure 13 – Example 1 of leaf mold being mistaken for spider mites

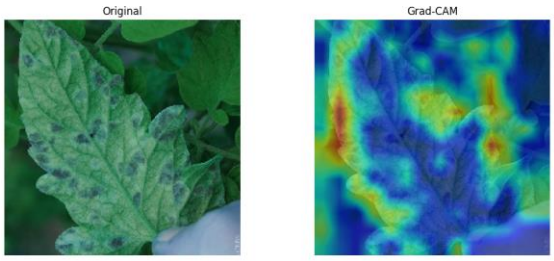

Figure 14 – Example 2 of leaf mold being mistaken for spider mites

Figure 15 shows that tomato leaf should be flat and fully visible for it to be classified correctly, as it can be seen that there is very little activation on the leaf itself when the image is taken from the side.

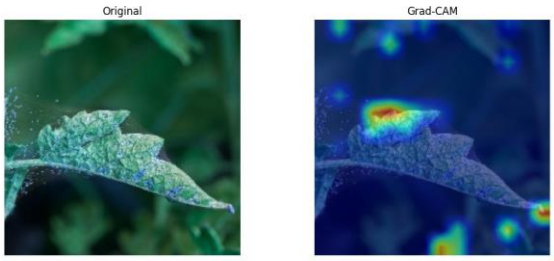

Figure 15 - Example of misclassification due to the leaf being captured from the side

E. Discussion

The results demonstrate the potential of CNNs, particularly ResNet152V2, for automated tomato leaf disease detection. However, to make it commercially useful, a much larger, carefully labeled dataset of real-world images should be gathered, as it would greatly help to improve the performance of CNN models.

VI. Conclusion

This research investigated the application of CNNs for tomato leaf disease detection, aiming to improve upon existing benchmarks while addressing the challenges of real-world image classification. With a carefully constructed dataset combining laboratory-style images and real-world examples, a range of CNN architectures were evaluated.

A. Key Findings

ResNet152V2 outperformed other architectures on both PlantVillage and real-world datasets.

A significant performance gap was observed between datasets, highlighting the complexities introduced by real-world images.

B. Comparison to Prior Work

On the PlantVillage dataset, the achieved accuracy of 93.50% surpasses the results reported by Agarwal et al. (2019). This improvement in results shows the effectiveness of the ResNet152V2 architecture and the well-balanced dataset used in this study.

While the present work did not achieve the same level of performance on the real-world dataset as Chohan et al. (2020) [14] and Ahmad et al. (2020) [13], it's important to note that this study tackled the greater challenge of classifying 10 disease classes.

C. Limitations and Future Directions

The significant performance decrease of the model on real-world image dataset indicates the need to substantially expand and diversify the real-world image dataset.

D. Contributions

This research provides valuable insights into architectural choices and the importance of realistic datasets for robust tomato leaf disease detection. It contributes to the ongoing development of practical AI-driven tools for agriculture.

References

[1] Dam, B.V., Goffau, M.D., Lidth de Jeude, J.V. and Naika, S., 2005. Cultivation of tomato: Production, processing and marketing. Agromisa.

[2] Panno, S., Davino, S., Caruso, A.G., Bertacca, S., Crnogorac, A., Mandić, A., Noris, E. and Matić, S., 2021. A review of the most common and economically important diseases that undermine the cultivation of tomato crop in the mediterranean basin. Agronomy, 11(11), p.2188.

[3] Smith, M.L., Smith, L.N. and Hansen, M.F., 2021. The quiet revolution in machine vision-a state-of-the-art survey paper, including historical review, perspectives, and future directions. Computers in Industry, 130, p.103472.

[4] Lu, J., Tan, L. and Jiang, H., 2021. Review on convolutional neural network (CNN) applied to plant leaf disease classification. Agriculture, 11(8), p.707.

[5] Ngugi, L.C., Abelwahab, M. and Abo-Zahhad, M., 2021. Recent advances in image processing techniques for automated leaf pest and disease recognition–A review. Information processing in agriculture, 8(1), pp.27-51.

[6] Sharma, V., Tripathi, A.K. and Mittal, H., 2022, June. Technological advancements in automated crop pest and disease detection: A review & ongoing research. In 2022 International Conference on Computing, Communication, Security and Intelligent Systems (IC3SIS) (pp. 1-6). IEEE.

[7] Agarwal, M., Singh, A., Arjaria, S., Sinha, A. and Gupta, S., 2020. ToLeD: Tomato leaf disease detection using convolution neural network. Procedia Computer Science, 167, pp.293-301.

[8] Sudha, V. & Ganeshbabu, Dr. (2020). A Convolutional Neural Network Classifier VGG-19 Architecture for Lesion Detection and Grading in Diabetic Retinopathy Based on Deep Learning. Computers, Materials & Continua. 66. 827-842. 10.32604/cmc.2020.012008.

[9] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2015). Rethinking the Inception Architecture for Computer Vision. *ArXiv*. /abs/1512.00567

[10] Chollet, F. (2016). Xception: Deep Learning with Depthwise Separable Convolutions. *ArXiv*. /abs/1610.02357

[11] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Identity Mappings in Deep Residual Networks. *ArXiv*. /abs/1603.05027

[12] Hughes, D. P., & Salathe, M. (2015). An open access repository of images on plant health to enable the development of mobile disease diagnostics. *ArXiv*. /abs/1511.08060

[13] Ahmad, Iftikhar & Hamid, Muhammad & Yousaf, Suhail & Shah, Syed & Ahmad, Muhammad Ovais. (2020). Optimizing Pretrained

Convolutional Neural Networks for Tomato Leaf Disease Detection. Complexity. 2020. 1-6. 10.1155/2020/8812019.

[14] Chohan, Murk & Khan, Adil & Chohan, Rozina & Katper, Saif & Mahar, Muhammad. (2020). Plant Disease Detection using Deep Learning. International Journal of Recent Technology and Engineering. 9. 909-914. 10.35940/ijrte.A2139.059120.

[15] kaustubhb999. (2022). Tomato Leaf Disease Detection [Dataset]. Kaggle. Retrieved November 20, 2023, from https://www.kaggle.com/datasets/kaustubhb999/tomatoleaf

[16] Lin, C., Li, L., Luo, W., Wang, K.C. and Guo, J., 2019. Transfer learning based traffic sign recognition using inception-v3 model. Periodica Polytechnica Transportation Engineering, 47(3), pp.242-250.

[17] Bakır, H., 2023. Evaluating the impact of tuned pre-trained architectures' feature maps on deep learning model performance for tomato disease detection. Multimedia Tools and Applications, pp.1-22.

[18] Ikechukwu, A.V., Murali, S., Deepu, R. and Shivamurthy, R.C., 2021. ResNet-50 vs VGG-19 vs training from scratch: A comparative analysis of the segmentation and classification of Pneumonia from chest X-ray images. Global Transitions Proceedings, 2(2), pp.375-381.

[19] Rachburee, N. and Punlumjeak, W., 2022. Lotus species classification using transfer learning based on VGG16, ResNet152V2, and MobileNetV2. IAES International Journal of Artificial Intelligence, 11(4), p.1344.

[20] J. Deng, W. Dong, R. Socher, L. -J. Li, Kai Li and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 2009, pp. 248-255, doi: 10.1109/CVPR.2009.5206848.

[21] Kingma, D. P., & Ba, J. (2014). Adam: A Method for Stochastic Optimization. *ArXiv*. /abs/1412.6980

[22] Zhang, Z., & Sabuncu, M. R. (2018). Generalized Cross Entropy Loss for Training Deep Neural Networks with Noisy Labels. *ArXiv*. /abs/1805.07836

[23] Bensaali, F., & Dakua, S. P. (2022). Scheduling Techniques for Liver Segmentation: ReduceLRonPlateau Vs OneCycleLR. *ArXiv*. /abs/2202.06373

[24] Google Developers. (2023). Classification: Accuracy [Web page]. Machine Learning Crash Course. Retrieved November 20, 2023, from https://developers.google.com/machine-learning/crash-course/classification/accuracy

[25] F-score. (2023, November 18). In Wikipedia. Retrieved November 20, 2023, from https://en.wikipedia.org/wiki/F-score

[26] Sokolova, Marina & Japkowicz, Nathalie & Szpakowicz, Stan. (2006). Beyond Accuracy, F-Score and ROC: A Family of Discriminant Measures for Performance Evaluation. AI 2006: Advances in Artificial Intelligence, Lecture Notes in Computer Science. Vol. 4304. 1015-1021. 10.1007/11941439_114.

[27] Precision and recall. (2023, November 19). In Wikipedia. Retrieved November 20, 2023, from https://en.wikipedia.org/wiki/Precision_and_recall

[28] Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2016). Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. *ArXiv*. https://doi.org/10.1007/s11263-019-01228-7