International Journal of Computational and Experimental Science and ENgineering
(IJCESEN)
Vol. 11-No.3 (2025) pp. 5396-5404
http://www.ijcesen.com
Copyright © IJCESEN
ISSN: 2149-9144

Research Article

Comparative Analysis of Deep Learning Models for Tomato Leaf Disease Classification: Insights and Opportunities

Mamatha. G[1*], G T Raju[2]

[1]Research Scholar, Department of Computer Science and Engineering, SJCIT, Chickballapur- 562101
***Corresponding Author Email:** mamathagowda812@gmail.com **ORCID** : 0000-0002-2654-567X

[2]Professor, Department of Information Science and Engineering, SJCIT, Chickballapur- 562101
Email gtraju1990@yahoo.com - **ORCID**: 0000-0003-2446-6596

Article Info:

DOI: 10.22399/ijcesen.3483
Received : 21 May 2025
Accepted : 13 July 2025

Keywords

Comparative Analysis
Deep Learning Models
Tomato Leaf Disease

Abstract:

This study evaluates the performance of four deep learning models—CNN, ResNet50, VGG16, and MobileNetV2—on the classification of tomato disease images into seven distinct categories: Septoria Leaf Spot, Early Blight, Mosaic Virus, Spider Mites, Target Spot, Leaf Mold, and Healthy Leaf. Using a dataset of 1,100 images, equally distributed across categories, the models were trained on 1,000 images and tested on 100 images to ensure standardized performance evaluation. ResNet50 demonstrated superior performance with an accuracy of 86.8%, precision of 87.72%, recall of 86.8%, and F1 score of 86.74%. VGG16 followed with an accuracy of 78.7% and F1 score of 78.7%, showcasing competitive but slightly lower efficacy compared to ResNet50. The custom CNN model achieved moderate results with an accuracy of 73.9% and an F1 score of 73.57%. Computationally efficient but with lowest performance metrics of 69.4% and 69.52% accuracy and F1 score respectively, MobileNetV2 was an underperformer. Data visualization showed a balanced dataset distribution for unbiased training, and we used data augmentation to improve model generalizability and reduce overfitting. Deep architecture and residual connection demonstrated to play a crucial role in feature extraction and classification. Future work could focus on hyperparameter tuning, more sophisticated architectures (such as EfficientNet) and combining the different architectures in order to maximize performance. It may also help to expand the dataset and to use transfer learning. ResNet50's efficacy for complex image classification tasks is evident from these findings and the potentials for improvement in deep learning based agricultural disease diagnosis are also shown.

1. Introduction

1.1. Overview of the Domain

Tomatoes are one of the most widely cultivated crops, and their production is among the most important for agro-economies and food supply chains. Nevertheless, tomato plants are susceptible to diseases caused by fungi, bacteria, viruses, etc which make it difficult to cultivate them. These diseases not only decrease yield, but they also compromise fruit quality, causing economic losses and perhaps supply chain disruption. Early and accurate detection of tomato leaf diseases is required for timely intervention, and sustainable crop management.

Challenges in Traditional Disease Detection

Manual inspection by experts or farmers is typical of conventional methods for diagnosing tomato plant diseases. While effective in certain contexts, these methods have several limitations:

1. Labor-Intensive Processes: Large scale farming operations are very time and labour intensive and require manual inspection [10].
2. Human Error and Subjectivity: Inconsistent or inaccurate diagnoses result [11] from such factors as fatigue, differing levels of expertise, and environmental conditions.
3. Late Detection: In many cases disease is only discovered after visible damage has occurred, by which point the infection has already spread, making management more difficult [12].

4. Resource Constraints: In remote or resource constrained areas, disease detection is further exacerbated by limited access to agricultural experts.

Emergence of Automated Solutions

Technology has made rapid advances, most importantly in AI, ML and DL, with its influence being felt most strongly in aspects of the agricultural sector. Plant disease detection through automated process has been becoming more and more feasible with the integration of computer vision and deep learning. However, CNNs are particularly good at processing what is complex visual data, and therefore are a good choice for analyzing images of diseased tomato leaves.

1.2. Role of Deep Learning in Tomato Disease Detection

Neural network architectures are used by deep learning to extract hierarchical features of raw input data, automatically. These models, in the plant disease detection context, take a visual pattern of leaf discoloration, spots, or texture changes and classify and predict diseases.

Existing Research and Bridging the Gap

Plant disease detection using CNNs, and pre trained architectures has been successfully demonstrated by several studies. These models have proven their promise but their performance usually relies on the quality of the dataset, preprocessing techniques and model’s generalization ability. Despite various challenges like imbalance, overfitting and not applicable in real world still have not been solved. Furthermore, the scalability and computational efficiency of these models are crucial for their deployment in real world agricultural settings, as the size of the system is vast and computations must be performed quickly.

By evaluating four models on a publicly available tomato leaf disease dataset, this work adds to the ongoing research. The uniformity of the study is achieved by the application of advanced preprocessing techniques such as resizing, normalization, and data augmentation. This research also looks at the feasibility of fine tuning and transfer learning of pre-trained models and compares it to a custom CNN.

In this work, we propose to improve the inclusion of deep learning into agricultural practices and provide a scalable, accurate and low-cost solution to the management of tomato plant diseases.

1.3. Objectives

The goal of this study is to evaluate and compare the performance of four deep learning architectures: CNN, VGG16, ResNet50 and MobileNetV2, for detecting and classifying tomato leaf diseases using a publicly available dataset. The primary objectives include:

- Dataset Preparation: Preparing a robust dataset by resizing, normalizing, and augmenting images to ensure compatibility with the models and simulate real-world variability.
- Model Training and Fine-Tuning: Training a custom CNN alongside pre-trained models (VGG16, ResNet50, MobileNetV2), with fine-tuning to tailor their feature extraction capabilities to the dataset.
- Performance Comparison: Using evaluation metrics such as accuracy, precision, recall, F1-score, and confusion matrices to assess and compare the models' classification capabilities.
- Real-World Applicability: Leveraging augmented datasets and transfer learning to optimize models for practical deployment in disease management systems.

2. Literature Review

Several studies looking into advanced methods of tomato plant disease detection using deep learning techniques, including accuracy, computational efficiency and feasibility in real world settings, have been conducted.

In [1], they proposed a DL-based system to detect tomato leaf diseases using AlexNet and SqueezeNet architectures, and deployed it on the real time basis on Nvidia Jetson TX1. The authors used the PlantVillage dataset to show potential of autonomous monitoring with RGB cameras and fabricated greenhouses. The study in [2] used a custom convolutional neural network (CNN) architecture with three convolution and max pooling layers based on which the tomato diseases can be detected and classified. The proposed model outperformed pre trained architectures like VGG16 and InceptionV3 with an average accuracy of 91.2% on 10 classes. Classifying nine types of tomato diseases was achieved using transfer learning in [3] to optimize training efficiency. The authors compared five architectures: DenseNet_Xception, ShuffleNet, ResNet50, etc. The authors found that DenseNet_Xception showed the highest accuracy of 97.1% and has good potential for integration by intelligent diagnostic systems on mobile platforms.

A LeNet CNN model variation was used in [4] to detect tomato leaf diseases with low computational resources. It was shown that the approach reaches an average accuracy of 94 - 95 %, and thus remains competitive in resource constrained settings. In [5] a review of deep learning and machine learning techniques for tomato disease detection was conducted. The study pointed out the weakness of the traditional image processing methods and pointed out the necessity of robust framework using public and private datasets to boost the prediction accuracy. In [6], the researchers used image processing techniques such as image segmentation and clustering to develop a reliable and accurate tomato leaf disease detection system specifically for India's agricultural needs

In [7], the authors introduced two deep learning architectures: The first one is with residual learning and the other one has an attention mechanism on top of the residual networks. With these proposed models, using the PlantVillage dataset, we achieve an accuracy of 98% in the detection of early blight, late blight, and leaf-mold, indicating the efficacy of hierarchical feature learning. A CNN based classification system on top of Raspberry Pi hardware was developed in [8] to classify common tomato diseases, like late blight and bacterial-canker. Feature extraction using image processing techniques was used and focused on practical applicability to farmers. A novel method to generate synthetic images using Conditional Generative Adversarial Networks (C-GAN) was proposed in [9] to overcome the problem of having limited labelled data. They used synthetic and real images to train DenseNet121 and achieved accuracy rates of 99.51% (5 classes), 98.65% (7 classes) and 97.11% (10 classes), showing the virtues of synthetic data generation combined with transfer learning.

Together, these studies demonstrate the importance of architecture selection, data augmentation and computational efficiency in the development of practical deep learning solutions for agricultural challenges and highlight the development of these methodologies for tomato disease detection.

3. Methodology

The dataset preparation, model architectures, training processes and evaluation metrics in this study are described in detail in this section. Four models were utilized for comparative analysis: Each among CNN, VGG16, ResNet50, InceptionV3 is chosen based on its distinctive architecture and talent to extract detailed attributes for image classification. The model as in Fig 3.1 was broadly followed for implementation.

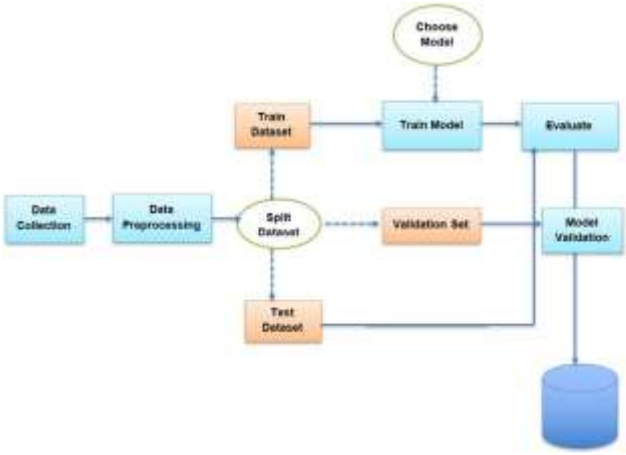

Figure 1. *Implementation flow*

2.1 Dataset and Preprocessing

Source and Structure: The dataset was derived from a public dataset having images of tomato diseases. The different images represented different scenarios that the models needed to be robust at classifying variations.

Preprocessing Workflow

1. Resizing**:** To ensure uniformity across inputs and make these inputs compatible with the input layers of all four models, each of the images was resized to fixed dimension of 112×112 pixels. The resizing step achieves computational efficiency and feature resolution balance.

2. Normalization**:** Pixel intensity values were scaled to the range [0, 1] to stabilize training. Standardizing input distributions on this adjustment reduced computational overhead and faster convergence.

3. Data Augmentation: We applied augmentation techniques to combat overfitting and improve model generalizability. Random rotations (up to ±30°), horizontal and vertical flips, random zooming (10–20%), cropping and contrast adjustments were used for these. The dataset was augmented with augmented samples, introducing diversity to the dataset, simulating real world conditions like image distortions and different orientations.

4. Dataset Partitioning**:** The dataset was split into three subsets; training 70%, validation 20% and testing 10%. This stratified division maintained each subset proportionate to all classes so that class distribution and bias was minimized.

2.2 Model Architectures

Four models were employed to examine their performance on the given dataset:

1. Custom Convolutional Neural Network (CNN)

Comprised multiple convolutional layers with 3×3 kernels for spatial feature extraction, each followed by ReLU activation to introduce non-linearity. Max-pooling layers reduced spatial dimensions while retaining critical features, thereby minimizing computational complexity. Fully connected dense layers transformed extracted features into a class probability distribution, culminating in a softmax output layer for classification.
A typical CNN model has been shown in figure 2.

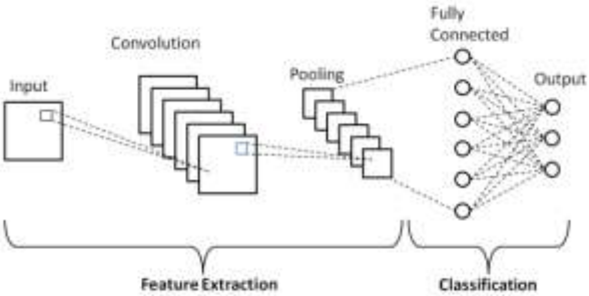

***Figure 2**. A typical CNN Model*

The deployed model has the following parameters specifically, as in figure 3.

Layer (type)	Output Shape	Param #
data_augmentation (Sequential)	(None, None, None, 3)	0
conv2d (Conv2D)	(None, None, None, 32)	896
max_pooling2d (MaxPooling2D)	(None, None, None, 32)	0
conv2d_1 (Conv2D)	(None, None, None, 16)	4,624
max_pooling2d_1 (MaxPooling2D)	(None, None, None, 16)	0
conv2d_2 (Conv2D)	(None, None, None, 32)	4,640
max_pooling2d_2 (MaxPooling2D)	(None, None, None, 32)	0
global_average_pooling2d (GlobalAveragePooling2D)	(None, 32)	0
output_layers (Dense)	(None, 10)	330

***Figure 3.** Deployed CNN Model parameters (code output snippet)*

2. VGG16

A 16-layer deep pre-trained model renowned for its structured arrangement of convolutional layers followed by pooling layers. Fine-tuning of the model's weights was performed, focusing on the later layers to adapt its feature representations to the specific dataset.
A typical VGG16 model has been shown in figure 4.

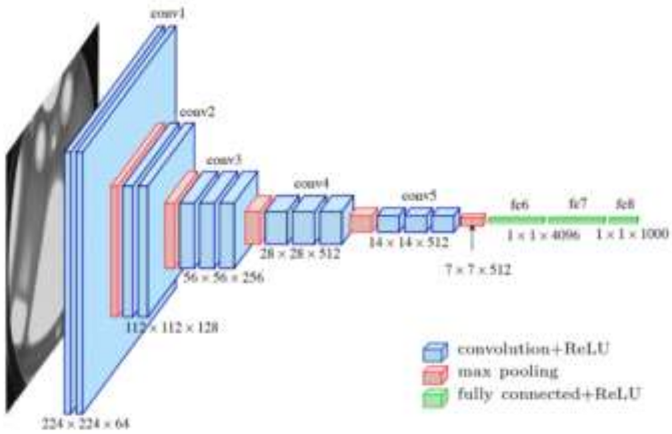

***Figure 4.** A typical VGG16 Model*

3. ResNet50

A deep residual network with 50 layers, utilizing residual connections to address vanishing gradient issues. The skip connections allowed gradients to flow uninterrupted, improving learning efficiency even in deeper layers. Its modular design effectively captures both low-level and high-level features.

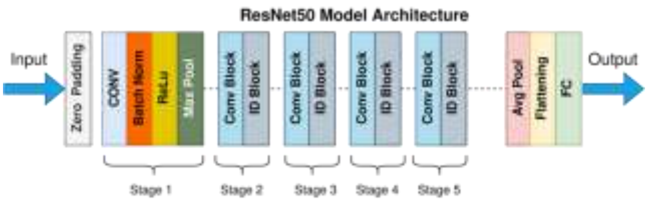

***Figure 5.** ResNet-50 Architecture used*

4. MobileNetV2

A sophisticated model employing parallel convolutional operations with varying kernel sizes within its Inception modules. This architecture enabled effective multi-scale feature extraction, capturing fine-grained and coarse features simultaneously. Fine-tuning was conducted on this model to leverage its robust feature extraction capabilities while tailoring it for the given classification task.

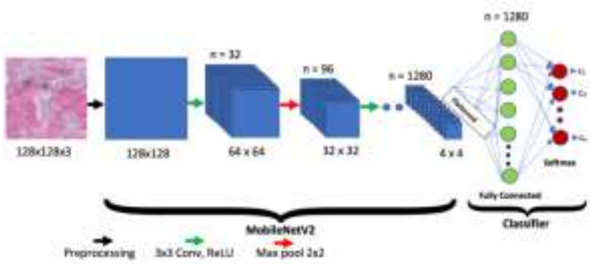

***Figure 6.** MobileNetv2 Architecture*

Role of Data Augmentation

All models leveraged the augmented dataset to improve their generalizability. While the custom CNN benefited from augmentation to prevent overfitting, the pre-trained models (VGG16, ResNet50, and MobileNetV2) used augmentation to simulate real-world conditions, further refining their transfer learning performance.

2.3 Training and Validation

The training process emphasized optimizing model performance while maintaining generalization to unseen data.

- Optimization: It was performed for backpropagation and weight updates, ensuring smooth convergence.
- Loss Function: Categorical cross-entropy was employed to measure the divergence between predicted and true class probabilities, providing a gradient for backpropagation.
- Validation Monitoring: Model performance was monitored on the validation set during training to identify potential overfitting or underfitting trends.

2.4 Evaluation Metrics

Performance evaluation was conducted using multiple metrics to provide a holistic view of each model's classification capabilities:

1. Accuracy**:** It measures the proportion (out of total) of the samples correctly classified.
2. Precision**:** It evaluates how frequently the true positives were among all samples predicted as positive, allowing for the installation of false positives.
3. Recall (Sensitivity): It shows how well the model can find all actual positives with as few false negatives as possible.
4. F1-Score: In datasets that have imbalanced class distributions, the harmonic-mean of precision and recall, balancing their trade-offs, is especially important.

Performance Visualization:

- *Confusion Matrix:* Provided detailed class-wise predictions, categorized correctly classified samples and misclassifications for error analysis.
- *Model Comparisons:* The classification task was then performed on using CNN, VGG16, ResNet50, InceptionV3 and then compared on accuracy, precision, recall and F1 score to determine most effective architecture for this task.

4. Results

This section presents the results of the study in two parts: Data Visualization, which provides insights into the dataset's distribution and representative images, and Model Results, which highlights the comparative performance of the four models.

4.1. Representative Images from Dataset Categories

A representative image from each category is shown to give an idea of the dataset composition. We use these images to demonstrate the distinguishing features within each category and provide context for the classification task. The categories represented in the dataset correspond to the following classes:

1. Septoria Leaf Spot: This category is a collection of images of leaves with small, dark spots with yellow halos (a symptom of the fungal infection Septoria lycopersici). They normally occur as a scattered spots on the leaf surface and cause premature defoliation.
2. Early Blight: Leaves with concentric rings or 'target like' lesions caused by the Alternaria solani fungus are included in this category. The areas are brown to black, and often surrounded by yellowing tissue.
3. Mosaic Virus: These are leaves with light and dark green mottles, a characteristic of viral infection, and they fall into this category. The virus inhibits chlorophyll production, causing spots, resulting in irregular patches that can greatly impact photosynthesis and plant growth.
4. Spider Mites: The leaves under this category were damaged due to the infestations of Tetranychus urticae. Stippling, discoloration, and even fine webbing can be seen, and in severe cases, symptoms. The damage, which typically leaves the foliage dusty or bronzed, can also cause leaves to take on a sickly purple or brown appearance.
5. Target Spot: This category describes symptoms caused by Corynespora cassiicola, characterised by round, necrotic spots with concentric rings. The lesions often coalesce to produce extensive leaf damage that can drastically reduce crop yield.
6. Leaf Mold: Leaves in this category have yellow upper surface and olive green to brown mold-like growth on the underside; symptoms are caused by Cladosporium fulvum infection. Humid environments are perfect for the disease, and severe defoliation can occur.
7. Healthy Leaf: The images in this category are of leaves that are healthy (uniform green and no visible symptoms of disease or pest damage) and unaffected by disease or pest problems. These are

leaves, which serves as a control group: baseline for model classification accuracy.

The dataset underlines the variability in visual characteristics by showing a representative image of each category. The visual diversity of these examples shows the difficulties and possibilities in designing models that can make fine distinctions between categories. Accurate classification of these categories requires these distinctions, in categories where symptoms overlap or symptoms present in reduced severity.

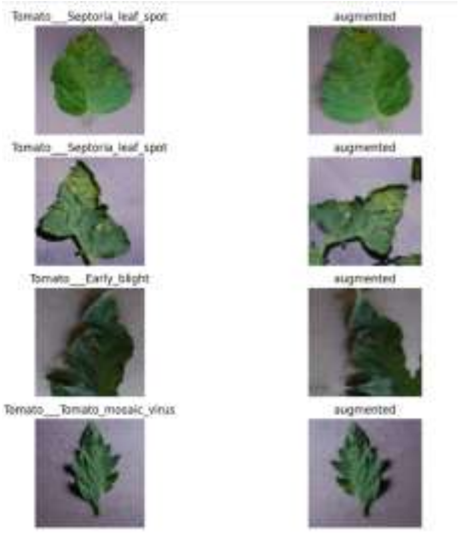

Figure 7. *Sampel images from the dataset*

4.2. Dataset Visualization

The dataset visualization provides an overview of the data distribution across categories, aiding in understanding the dataset's structure and balance. The number of images in each category is shown on bar plots to concentrate on the organization of the dataset. As shown in Figure 8., the dataset consists of a total of 1,100 images and 1,000 for training and 100 for testing. This division guarantees that the data is enough to train while keeping the model performance evaluated the same way. Figure 9., also displays a pie chart indicating the proportion of images from each category with 10% for each category. This uniform distribution guarantees a balanced dataset, as we don't want biases in the dataset that cause the model to be trained in an unbalanced manner and to not produce consistent classification accuracy per category.

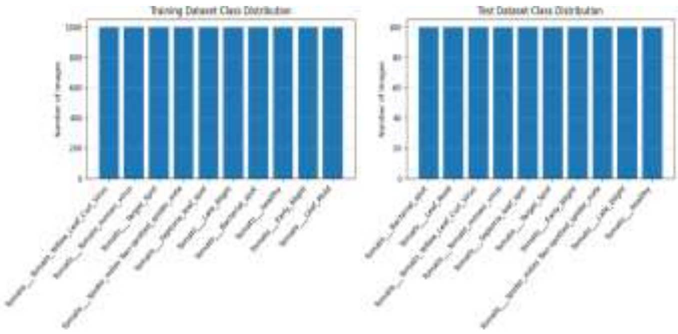

Figure 8. *Dataset visualization bar plots*

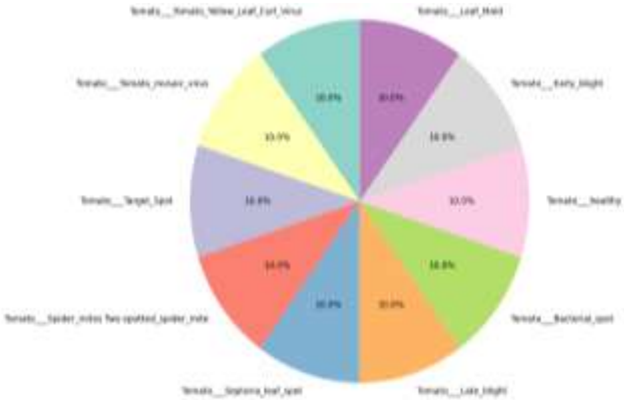

Figure 9. *Dataset visualization pie-chart*

4.3. Confusion Matrix and Model Training results

A. Confusion Matrix

The confusion matrix is a performance measurement tool used to evaluate the effectiveness of a classification model. It provides a detailed breakdown of the model's correct and incorrect classifications across different categories. The confusion matrix for the model ResNet50 is plotted as follows:

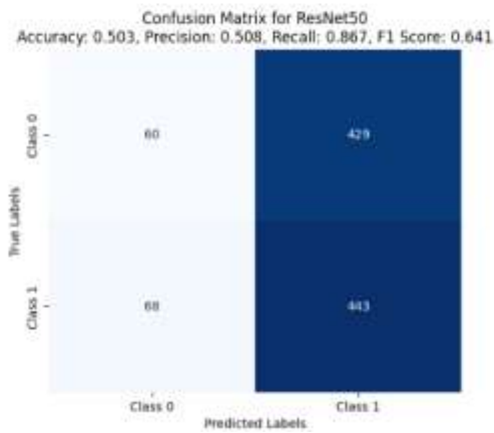

Figure 10. *Confusion Matrix*

The results are both tabulated and visualized using bar plots for a more comprehensive understanding.

Comparative Performance Metrics

The following table summarizes the performance of each model across the selected metrics:

Table 1. Results Table

Metric	CNN	ResNet50	VGG16	MobileNetV2
Precision	0.755826	0.877287	0.803540	0.731699
Recall	0.739000	0.868000	0.787000	0.694000
F1 Score	0.735780	0.867413	0.787356	0.695278
Accuracy	0.739000	0.868000	0.787000	0.694000

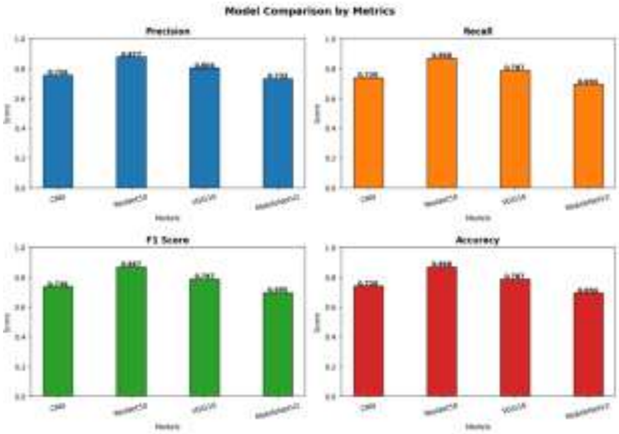

Figure 11. *Model's Comparison in bar-plot*

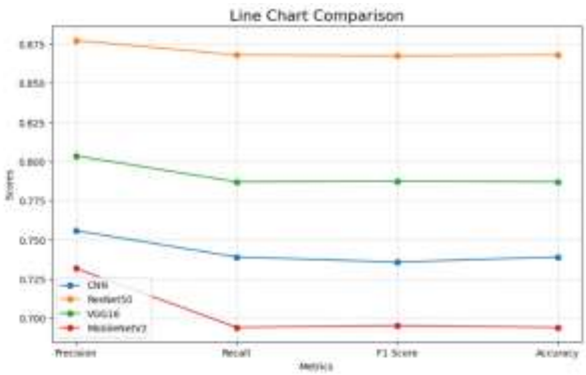

Figure 12. *Model's Comparison in Line-graph*

Interpretation of Results

1. CNN Performance:

- The CNN model achieved an accuracy of 73.9%, which indicates a moderate ability to classify the images correctly.
- Its F1 score of 73.57% demonstrates a balance between precision (75.58%) and recall (73.9%). However, the performance is outshined by deeper architectures such as ResNet50 and VGG16.

2. ResNet50 Performance:

- ResNet50 emerged as the best-performing model, with an accuracy of 86.8% and an F1 score of 86.74%.
- Its precision (87.72%) and recall (86.8%) are also the highest among the models, highlighting its superior ability to identify and correctly classify images across all categories.

3. VGG16 Performance:

- The VGG16 model delivered an accuracy of 78.7%, with a precision of 80.35% and a recall of 78.7%.
- While not as robust as ResNet50, VGG16 outperforms CNN and MobileNetV2, showcasing its effectiveness in handling image classification tasks.

4. MobileNetV2 Performance:

- MobileNetV2 achieved the lowest scores among the models, with an accuracy of 69.4% and an F1 score of 69.52%.
- While it is lightweight and computationally efficient, its relatively lower precision (73.17%) and recall (69.4%) suggest it may not be the best choice for datasets requiring higher accuracy.

Visualization and Insights

- ResNet50 consistently outperforms the other models across all metrics, reflecting its architectural depth and ability to extract complex features from the dataset.
- VGG16 demonstrates competitive performance but falls short of ResNet50, likely due to its comparatively older design and lack of residual connections.
- CNN and MobileNetV2 show limitations in accuracy and F1 scores, suggesting these models might not be ideal for datasets of similar complexity.

These visualizations provide a clear comparative perspective and underscore the importance of choosing architectures aligned with the complexity and size of the dataset.

Some of the key advantages of using deep learning models include:

1. High Accuracy*:* Due to their capacity to learn nonlinear patterns at subtle levels it is common to see deep learning models outperforming traditional image processing techniques in the classification tasks [13].

2. Scalability**:** The models can be trained once, and will process large datasets in real time, making them suitable for use within automated monitoring systems [14].
3. Robustness to Variations: These models can tolerate a number of real-world conditions with proper preprocessing and data augmentation, such as varying lighting, angles, or background noise [15].

5. Discussion

1. Summary of Findings

For classifying tomato disease images, we evaluated four deep learning models: CNN, ResNet50, VGG16 and MobileNetV2 and found huge differences in their performance. Finally, the most accurate model we found was ResNet50, with a precision of 87.72%, recall of 86.8%, F1 score of 86.74% and accuracy of 86.8%. We attribute its superior performance to its deep network architecture and residual connections that reduce the vanishing gradient problem and improve feature extraction.
Although not as good as ResNet50, VGG16 still gave performance competitive with an accuracy of 78.7% and an F1 score of 78.7%. While not able to classify as effectively as it could have based on the more traditional architectures, this model was able to classify images and the only failings seen pertained to the more dated nature of its design, lacking the sophisticated features seen in more contemporary architectures, such as ResNet50.
Accuracy of 73.9% and F1 score of 73.57% was achieved by the custom CNN model. The CNN model was not able to perform as well as the pre trained models, but was effective for simple image classification tasks. We showed that despite being deep, basic CNNs lack the ability to handle complex datasets as well as deeper architectures, such as VGG16 and ResNet50.
The models were MobileNetV2, with the lowest performance that achieved the accuracy of 69.4% and F1 score of 69.52%. The implication of this result is that, while intended for speed and efficiency, MobileNetV2 may not be the appropriate choice for applications where higher accuracy is also required, especially for intricate image classification workloads.

2. Future Scope

The best model in this work was ResNet50, but there is still room for improvement and exploration in future work. One area for improvement in models fine tuning hyper parameters like learning rate, batch size and dropout rates for better results. It also enables us to study the effect of other state of the art architectures such as EfficientNet or DenseNet and further improve the accuracy and efficiency.
In future studies one could also explore combining different models in an ensemble model in order to take advantage of the strengths of each architecture. If one could couple MobileNetV2's light weight nature with ResNet50's high accuracy, it should result into low computation and high accuracy, which could be deployed on any edge device.
Additionally, the dataset for this study could be enlarged with more categories and images from different environments to improve the robustness and generalize ability of models to new data. Other types of augmentation, such as color jittering, or adding noise, may also be incorporated to further simulate real world conditions.
Finally, further research can be conducted to exploit the transfer learning techniques to build on the pre trained models on larger datasets. This would allow domain specific fine tuning on tasks to improve classification performance especially when there were imbalanced or underrepresented classes.

6. Conclusion

On the task of classifying tomato disease images, this study evaluated the performance of four deep learning models, CNN, ResNet50, VGG16, and MobileNetV2. The results show that ResNet50 is the most accurate, precise, recall, and F1 score in all the models for the tasks of complex image classification. Older in design, but still having shown competitive results, VGG16 shows that it is still a viable option for image classification tasks with moderately complex datasets. At the same time, although the custom CNN model worked fine, it did not perform as well as the pre trained models, favouring the use of more advanced architectures for complex tasks. Although being lightweight and efficient, MobileNetV2 was not very effective in this context.
To prevent overfitting and achieve more robust performance on real data, data augmentation techniques were critical in improving model generalisation. This helped us achieve fair and accurate results across all classes, as the balanced dataset.
As we look ahead, there are a number of opportunities to improve model performance further. By fine tuning hyperparameters, exploring newer architectures, and using ensemble models we can even obtain more accurate and efficient results. Increasing the models' generalizability and robustness may lie in expanding the dataset to include more images and categories. It should be

continued to refine deep learning models for their application in agriculture, improving accuracy and optimizing models for real time deployment on edge devices.

Finally, this study shows how deep learning models can be used for detecting diseases in agriculture. With appropriate selection and optimization of the appropriate models, these technologies can provide important contributions to the timely and accurate identification of disease, and thus provide important tools for improving crop management and agricultural sustainability.

Author Statements:

- **Ethical approval:** The conducted research is not related to either human or animal use.
- **Conflict of interest:** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper
- **Acknowledgement:** The authors declare that they have nobody or no-company to acknowledge.
- **Author contributions:** The authors declare that they have equal right on this paper.
- **Funding information:** The authors declare that there is no funding to be acknowledged.
- **Data availability statement:** The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

References

[1] Durmuş, Halil, Ece Olcay Güneş, and Mürvet Kırcı, (2017). Disease detection on the leaves of the tomato plants by using deep learning. *2017 6th International conference on agro-geoinformatics*. IEEE.

[2] Agarwal, Mohit, et al., (2020). ToLeD: Tomato leaf disease detection using convolution neural network. *Procedia Computer Science* 167: 293-301.

[3] Hong, Huiqun, Jinfa Lin, and Fenghua Huang, (2020). Tomato disease detection and classification by deep learning. *2020 International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE)*. IEEE.

[4] Tm, Prajwala, et al., (2018). Tomato leaf disease detection using convolutional neural networks. *2018 eleventh international conference on contemporary computing (IC3)*. IEEE.

[5] Thangaraj, Rajasekaran, et al., (2022). Artificial intelligence in tomato leaf disease detection: a comprehensive review and discussion. *Journal of Plant Diseases and Protection* 129.3: 469-488.4

[6] Ashok, Surampalli, et al., (2020). Tomato leaf disease detection using deep learning techniques. *2020 5th International Conference on Communication and Electronics Systems (ICCES)*. IEEE.

[7] Karthik, R., et al., (2020). Attention embedded residual CNN for disease detection in tomato leaves. *Applied Soft Computing* 86: 105933.

[8] Adhikari, Santosh, et al., (2018). Tomato plant diseases detection system. *1st KEC Conference Proceedings*. Vol. 1.

[9] Abbas, Amreen, et al., (2021). Tomato plant disease detection using transfer learning with C-GAN synthetic images. *Computers and Electronics in Agriculture* 187: 106279.

[10] Mehra, Tanvi, Vinay Kumar, and Pragya Gupta, (2016). Maturity and disease detection in tomato using computer vision. *2016 Fourth international conference on parallel, distributed and grid computing (PDGC)*. IEEE.

[11] De Luna, Robert G., Elmer P. Dadios, and Argel A. Bandala, (2018). Automated image capturing system for deep learning-based tomato plant leaf disease detection and recognition. *TENCON 2018-2018 IEEE Region 10 Conference*. IEEE.

[12] David, Hepzibah Elizabeth, et al., (2021). Literature review of disease detection in tomato leaf using deep learning techniques. *2021 7th International conference on advanced computing and communication systems (ICACCS)*. Vol. 1. IEEE.

[13] Mim, Tahmina Tashrif, et al., (2019). Leaves diseases detection of tomato using image processing. *2019 8th international conference system modeling and advancement in research trends (SMART)*. IEEE.

[14] Chen, Hsing-Chung, et al., (2022). AlexNet convolutional neural network for disease detection and classification of tomato leaf. *Electronics* 11.6: 951.

[15] Vetal, Sagar, and R. S. Khule, (2017). Tomato plant disease detection using image processing. *International Journal of Advanced Research in Computer and Communication Engineering* 6.6: 293-297.