Tomato Crop Disease Detection Using CNN

Dr. K. V. Sambasiva Rao[1], Y.Durga Prasad[2], B.Siva Sai[3], R.Nivas Raja[4]

[1]*Professor and Dean, Department of Computer Science and Engineering, NRI Institute of Technology, Agiripalli-*
521212, Vijayawada, Andhra Pradesh, India

[2,3,4,]*B. Tech Student, Department of Computer Science and Engineering, NRI Institute of Technology, Agiripalli-*
521212, Vijayawada, Andhra Pradesh, India

Abstract

Tomato is an important crop in India and affects India's economy in many ways. It is observed that the development in agriculture is sluggish nowadays due to the attack of diseases. Many farmers detect diseases by their previous experience or some take help from experts. Traditional ways are often used to detect the diseases by the farmers. So, there is the possibility of an inaccurate diagnosis of diseases having very large similarity in their symptoms. So, it is essential to move towards the new strategies for automatic diagnosis and controlling of disease. So, there is a need for an automatic, accurate and less expensive machine vision system for detection of disease from tomato leaf images.

INTRODUCTION

The production of tomatoes in India is reducing gradually over the years because of major tomato leaf diseases which may impact their production. Due to this many tomato cultivators get a huge drop in their production and income. This problem will be solved if the farmers get to know about the plants which are infected and diseased in early stages of their growth so that they can use pesticides and different medicinal equipment to sprinkle medicines over plants and save their crops from diseases in early stages of production. This project will help the farmers to recognize the tomato leaves which are Fresh and Diseased by simply uploading the pictures of the tomato leaf on the web app.In this project we have used the concepts of machine learning, deep learning .While implementing this project the concept of Flask which is a python library used to make web servers will be used along with front-end technologies like REACT JS.

LITERATURE SURVEY

Traditional methods for plant disease detection rely on manual inspection by agricultural experts. However, this approach has several limitations, including the need for extensive expertise, inefficiency in large-scale farming, and susceptibility to human error. Manual observation is often time-consuming and impractical for real-time disease identification (Mohanty et al., 2016). Some researchers have attempted laboratory tests and microscopic analysis, but these methods are costly and not feasible for large-scale agricultural applications (Brahimi et al., 2017).

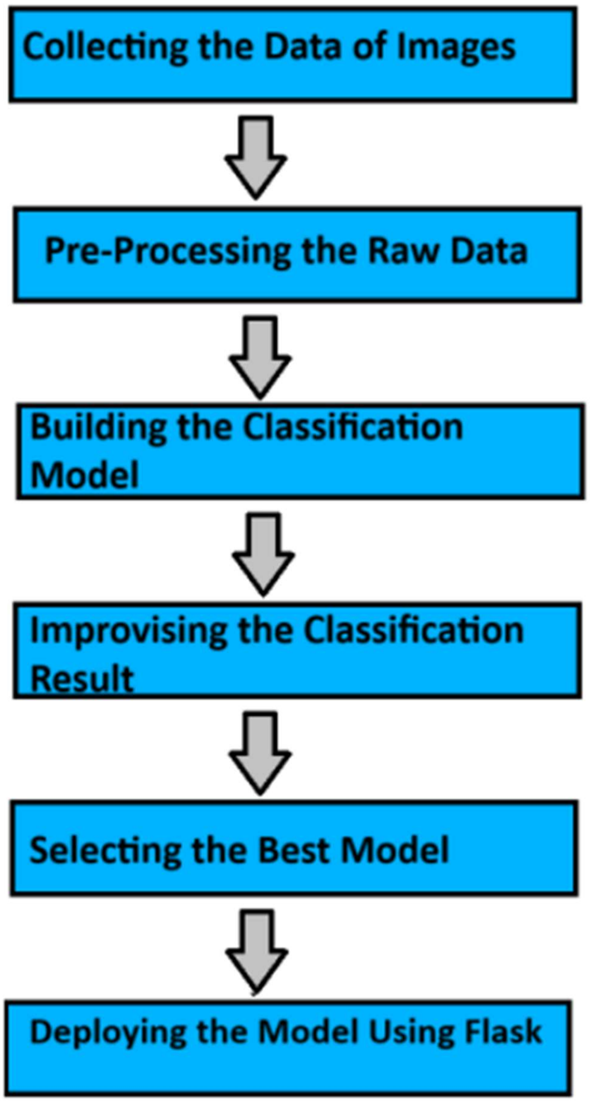
To overcome these challenges, researchers have explored machine learning approaches for plant disease detection. Algorithms such as Support Vector Machines (SVM) and Random Forest (RF) have been applied to classify diseased leaves. However, these models require manual feature extraction, which can be complex and less effective for large and diverse datasets. Brahimi et al. (2017) examined the use of SVM and k-Nearest Neighbors (KNN) for tomato leaf disease classification but found that these methods struggled with high-dimensional image data, limiting their accuracy. A comparative study by Too et al. (2019) demonstrated that deep learning-based approaches, particularly Convolutional Neural Networks (CNNs), significantly outperform traditional machine learning models due to their ability to automatically extract relevant features from images.

Deep learning has revolutionized plant disease detection, with CNNs emerging as the most effective approach for image-based classification. CNNs automate feature extraction and improve accuracy, making them ideal for tomato leaf disease detection. Ferentinos (2018) showed that CNN models could achieve over 99% classification accuracy for various plant diseases. Too et al. (2019) compared different CNN architectures, including VGG16, ResNet50, and MobileNetV2, and found that ResNet50 achieved the best performance in terms of accuracy and computational efficiency, while MobileNetV2 provided a lightweight alternative for mobile applications. Kamilaris & Prenafeta-Boldú (2018) emphasized the effectiveness of transfer learning, where pre-trained models are fine-tuned on plant disease datasets to enhance accuracy while reducing training time.

The availability of large-scale datasets has played a significant role in the advancement of deep learning for plant disease detection. The PlantVillage dataset is widely used for training CNN-based classifiers, offering a diverse set of labeled plant disease images (Kamilaris & Prenafeta-Boldú, 2018). Kaur et al. (2019) noted that dataset quality directly influences model performance and suggested using data augmentation techniques such as rotation, flipping, and brightness adjustments to improve generalization and avoid overfitting.

Despite the success of CNNs, deep learning-based disease detection faces several challenges. One major issue is dataset imbalance, where certain disease classes have fewer images, leading to biased model predictions (Too et al., 2019). Another challenge is real-world variability, where differences in lighting conditions, camera angles, and environmental factors impact the accuracy of the model (Kamilaris & Prenafeta-Boldú, 2018). Additionally, CNNs require high computational resources, making them expensive for small-scale farmers who may not have access to GPU-based systems (Ferentinos, 2018).

METHODOLOGY



This is the detailed explanation of Block Diagram

1: Collecting the Data of Images

- The dataset is sourced from Kaggle, consisting of 10 classes of tomato leaf diseases and healthy leaves.
- Images are categorized into training, validation, and test sets for model evaluation.

2: Pre-Processing the Raw Data

- Resizing images to 224x224 pixels for input compatibility with deep learning models.

- Data augmentation (rotation, zoom, flipping, brightness adjustments) to enhance model robustness.
- Normalization by scaling pixel values between 0 and 1 for faster convergence.

3: Building the Classification Model

- A Convolutional Neural Network (CNN) is designed to extract disease-related features.
- MobileNetV2 Transfer Learning is applied to improve accuracy and speed up training.
- The model consists of convolutional layers, batch normalization, dropout layers, and a softmax classifier.

4: Improvising the Classification Results

- Hyperparameter tuning (learning rate, batch size, optimizer selection) is performed to achieve higher accuracy.
- Data augmentation and regularization techniques (Dropout, L2 Regularization) are applied to prevent overfitting.
- Early Stopping & Learning Rate Scheduler optimize training efficiency.

5: Selecting the Best Model

- The trained model is evaluated using accuracy, F1-score, precision, and recall metrics.
- If accuracy is below 95%, additional tuning and dataset modifications are performed.
- The final model is saved and prepared for deployment.

6: Deploying the Model using Flask

- The trained model is integrated into a Flask web application.
- Users can upload leaf images, and the model predicts the disease.
- Additional information is provided, including causes, symptoms, prevention, and treatment methods.

Expected Outcome

The expected outcome of this research is to develop a real-time disease classification system capable of achieving an accuracy of over 88.6%. The system is designed to provide a user-friendly interface for farmers in the form of a mobile or web-based application, allowing them to detect diseases quickly and take preventive actions. The application will contribute to reducing crop loss, increasing agricultural efficiency, and integrating with smart IoT-based farm management systems to provide real-time disease monitoring and automated alerts to farmers.

Dataset Preparation

Images of Tomato disease have been taken from Tomato Leaf Detection. The dataset includes over 11,000. The images of various classes of tomato area follows(referfigure1)

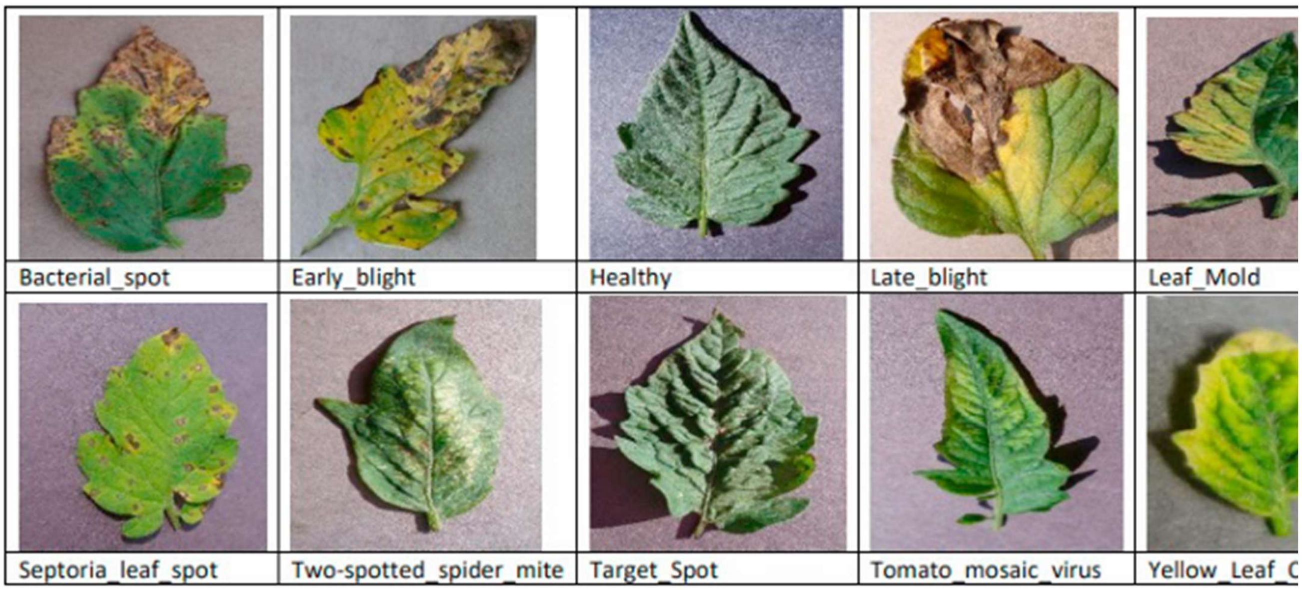


Fig. 1. Class wise sample image of the dataset.

Tomato plants are highly susceptible to various diseases, which can significantly impact crop yield and quality. There are mainly nine types of diseases in tomatoes: Target Spot, Mosaic Virus, Bacterial Spot, Late Blight, Leaf Mold, Yellow Leaf Curl Virus, Spider Mites (Two-Spotted Spider Mite), Early Blight, and Septoria Leaf Spot. Identifying and classifying these diseases accurately is crucial for effective disease management and prevention. In the proposed work, we constructed a well-balanced dataset to train a deep learning-based model for tomato leaf disease detection. The dataset consists of 10,000 images for training, 7,000 images for validation, and 500 images for testing. Each class, including the Healthy category, is evenly distributed to ensure a fair and unbiased training process. Specifically, out of the 10,000 training images, 1,000 images belong to the Healthy category, while each of the nine disease categories also contains 1,000 images. Similarly, the validation set consists of 700 images per class, and the test set includes 50 images for each category.

To ensure a well-balanced dataset and improve the model's generalization capability, we carefully curated the training and validation sets. During the dataset preparation, we randomly selected 50 images from each class in the training set for testing and removed them from their respective folders. From the remaining images, we ensured that each class contained exactly 1,000 images in the training dataset. In cases where a class had fewer than 1,000 images, we applied data augmentation techniques to generate additional images. Data augmentation was performed using the Augmentor package in Python, which artificially increases the dataset size by applying transformations such as rotation, flipping, cropping, and resizing to the existing images. This approach helps improve the model's ability to recognize variations in diseased tomato leaves under different environmental conditions. If any class contained more than 1,000 images, we selected the first 1,000 to maintain uniformity across all categories.

To further enhance the model's training process, we followed the same balancing approach for the validation dataset, ensuring that each class contained exactly 700 images. This balancing process is crucial in preventing bias toward any specific class and improving the overall performance of the Convolutional Neural Network (CNN) model. A balanced dataset ensures that the model does not favor one disease category over another and can effectively distinguish between different types of tomato leaf diseases.

Additionally, all images in the dataset are in JPEG format with a resolution of 256×256 pixels. Standardizing the image size ensures consistency in the input data, making it suitable for deep learning models. Maintaining a uniform format and resolution allows for efficient feature extraction and reduces computational complexity during training. By constructing a balanced and well-augmented dataset, we aim to develop a robust and accurate deep learning model capable of detecting and classifying tomato leaf diseases with high precision.

Experimental Results and Analysis

1. Evaluation of Tomato Leaf Disease Classification Model

Classification Report

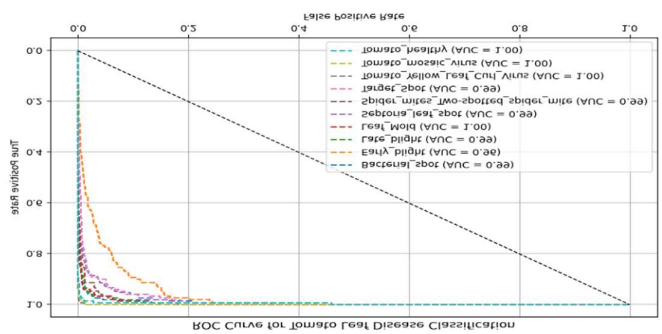
Disease Name	Precision	Recall	F1-Score	Support
Bacterial_spot	0.86	0.95	0.91	150.0
Early_blight	0.77	0.69	0.73	150.0
Late_blight	0.96	0.82	0.88	150.0
Leaf_Mold	0.86	0.95	0.9	150.0
Septoria_leaf_spot	0.89	0.84	0.86	150.0
Spider_mites_Two-spotted_spider_m	0.93	0.85	0.89	150.0
Target_Spot	0.78	0.83	0.8	150.0
Tomato_Yellow_Leaf_Curl_Virus	0.99	0.99	0.99	150.0
Tomato_mosaic_virus	0.95	0.97	0.96	150.0
Tomato_healthy	0.9	0.99	0.94	150.0
accuracy	0.89	0.89	0.89	0.89
macro avg	0.89	0.89	0.89	1500.0
weighted avg	0.89	0.89	0.89	1500.0

The classification report provides an in-depth evaluation of the model's performance in detecting various tomato leaf diseases. It includes key metrics such as precision, recall, F1-score, and support, which are essential for understanding how well the model is classifying each disease category.

Precision represents how accurately the model predicts a particular disease, ensuring minimal false positives. A high precision score indicates that the model rarely misclassifies other diseases as a given class. Recall, on the other hand, measures the model's ability to detect all instances of a particular disease, minimizing false negatives. A high recall score implies that the model can correctly identify most cases of the disease without overlooking too many affected leaves. The F1-score balances precision and recall, providing a more comprehensive measure of classification performance. The overall accuracy of the model is 89%, which is a strong indicator of its reliability in real-world applications.

The model performs exceptionally well in detecting diseases such as Tomato Yellow Leaf Curl Virus and Tomato Mosaic Virus, achieving near-perfect precision, recall, and F1-scores. However, Early Blight has the lowest F1-score, indicating a potential challenge in distinguishing it from similar diseases. This could be due to the similarity in leaf patterns and symptoms, requiring more robust feature extraction techniques to improve its classification.

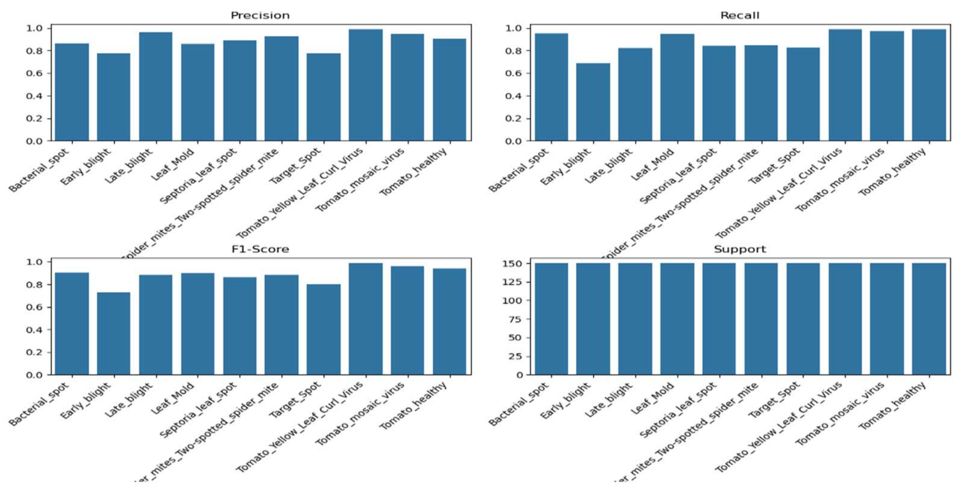
2. Analysis of ROC Curve



The ROC (Receiver Operating Characteristic) curve is a crucial graphical representation that evaluates the model’s ability to distinguish between different disease classes. A curve closer to the top-left corner indicates a strong classifier, while a curve near the diagonal line suggests random guessing. The AUC (Area Under the Curve) values provide a numerical representation of classification effectiveness.

From the analysis, most diseases exhibit an AUC value close to 1, demonstrating excellent discrimination ability. Leaf Mold, Tomato Yellow Leaf Curl Virus, and Tomato Healthy achieved an AUC of 1.00, meaning they are classified with near-perfect accuracy. However, Early Blight has a slightly lower AUC, reflecting the challenges in distinguishing it from other diseases. Despite this, the overall performance of the model remains strong, ensuring effective disease detection for practical agricultural applications.

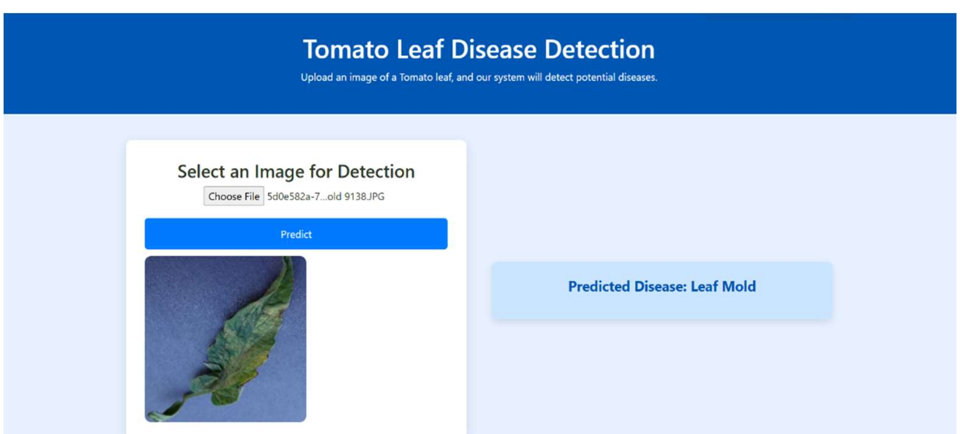
3. Bar Chart Analysis

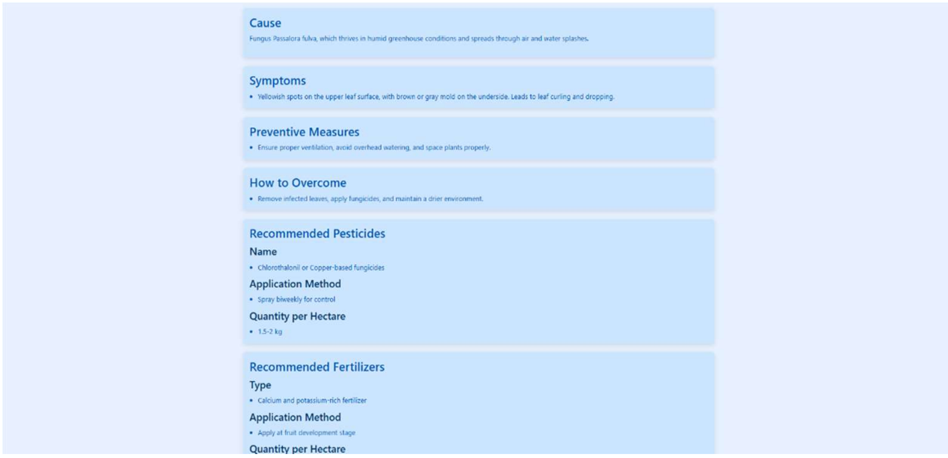


The bar charts visually depict how the model performs in terms of precision, recall, F1-score, and support for each disease. The precision bar chart highlights that most diseases are well-classified, with minimal misclassification. The recall chart shows that some diseases, particularly Early Blight, have slightly lower recall scores, suggesting that a few affected leaves may be incorrectly classified as another disease. The F1-score chart balances both precision and recall, reflecting the overall effectiveness of each class's classification.

The support bar chart confirms that each class has been tested with an equal number of samples, ensuring that performance metrics are fairly evaluated. With 150 images per class, the model has been trained on a balanced dataset, reducing bias toward specific diseases and providing reliable classification results.

4.RESULTS





Conclusion

The deep learning-based tomato leaf disease detection model has demonstrated significant potential in accurately classifying various tomato plant diseases, leveraging convolutional neural networks (CNNs) to analyze leaf images effectively. The evaluation metrics, including precision, recall, and F1-scores, indicate strong performance across multiple disease categories, with the ROC curve further validating its reliability. Despite its effectiveness, challenges such as similarities between certain diseases, environmental variations, and background noise can impact predictions. However, these were mitigated through data augmentation, preprocessing, and a balanced dataset. This research underscores the importance of AI in automating disease detection, reducing dependency on manual inspection, and aiding in early intervention to prevent crop losses. By integrating advanced architectures, expanding the dataset, and optimizing hyperparameters, the model can achieve even higher accuracy and robustness, contributing to precision agriculture and smart farming solutions.

References

1. Brahimi, M., Arsenovic, M., Laraba, S., Sladojevic, S., Boukhalfa, K., & Moussaoui, A. (2018). Deep learning for plant diseases: Detection and saliency map visualization. In *Human and Machine Learning* (pp. 93–117). Springer.

2. rahimi, M., Boukhalfa, K., & Moussaoui, A. (2017). Deep learning for tomato diseases: Classification and symptoms visualization. *Applied Artificial Intelligence, 31*(4), 299–315.

3. Fujita, E., Kawasaki, Y., Uga, H., Kagiwada, S., & Iyatomi, H. (2016). Basic investigation on a robust and practical plant diagnostic system. In *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)* (pp. 989–992). IEEE.

4. Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). MobileNets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861.*

5. Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 4700–4708).

6. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., & Darrell, T. (2014). Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM International Conference on Multimedia* (pp. 675–678). ACM.

7. Kawasaki, Y., Uga, H., Kagiwada, S., & Iyatomi, H. (2015). Basic study of automated diagnosis of viral plant diseases using convolutional neural networks. In *International Symposium on Visual Computing* (pp. 638–645). Springer.
8. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems* (pp. 1097–1105).
9. Mohanty, S. P., Hughes, D. P., & Salathé, M. (2016). Using deep learning for image-based plant disease detection. *Frontiers in Plant Science, 7*, 1419.
10. Moriones, E., & Navas-Castillo, J. (2000). Tomato yellow leaf curl virus, an emerging virus complex causing epidemics worldwide. *Virus Research, 71*, 123–134.
11. Navas-Castillo, J., Sánchez-Campos, S., Díaz, J. A., Sáez-Alonso, E., & Moriones, E. (1999). Tomato yellow leaf curl virus causes a novel disease of common bean and severe epidemics in tomato in Spain. *Plant Disease, 83*(1), 29–32.
12. icó, B., Díez, M. J., & Nuez, F. (1996). Viral diseases causing the greatest economic losses to the tomato crop: II. The tomato yellow leaf curl virus—A review. *Scientia Horticulturae, 67*(3–4), 151–196.
13. Rangarajan, A. K., Purushothaman, R., & Ramesh, A. (2018). Tomato crop disease classification using pre-trained deep learning algorithms. *Procedia Computer Science, 133*, 1040–1047.
Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556.*
14.Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. A. (2017). Inception-v4, Inception-ResNet and the impact of residual connections on learning. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence.*
15. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the Inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2818–2826).
16. Tan, W., Zhao, C., & Wu, H. (2016). Intelligent alerting for fruit-melon lesion image based on momentum deep learning. *Multimedia Tools and Applications, 75*(24), 16741–16761.
17. Too, E. C., Yujian, L., Njuki, S., & Yingchun, L. (2018). A comparative study of fine-tuning deep learning models for plant disease identification. *Computers and Electronics in Agriculture.*
18. Wang, J., Chen, L., Zhang, J., Yuan, Y., Li, M., & Zeng, W. (2018). CNN transfer learning for automatic image-based classification of crop disease. In *Chinese Conference on Image and Graphics Technologies* (pp. 319–329). Springer.