# Precision Tomato Disease Detection with a Hybrid Deep Learning Model Based on VGG-16 and ResNet-50

Sakshi Bajirao Patil, Bhagyashala Jadhawar, Ashwini Mandale
*Ashokrao Mane Group of Institutions Vathar*

**Abstract: One of the most significant crops in the world, tomato plants are widely cultivated in different regions of India. Tomato crops, however, are extremely vulnerable to a number of illnesses that can result in a significant loss of output. In this work, we suggest a deep ensemble learning method for tomato plant disease detection that makes use of VGG-16 and ResNet-50. In order to automatically extract and learn discriminative features from tomato leaf images and categorize them into healthy or diseased groups, the suggested method makes use of the advantages of both CNN architectures. We used a publicly accessible dataset of tomato leaf photos with five distinct disease classifications to assess the efficacy of the method. To increase its size and diversity, the dataset underwent pre-processing and augmentation. The ensemble model underwent testing and training on dataset to achieving an overall accuracy of 95.78% on the test set. Furthermore, we compared the performance of the proposed deep ensemble model with other state-of-the-art classification techniques, demonstrating that the ensemble model outperformed them. This deep ensemble-based tomato plant disease detection system can serve as a valuable tool for farmers and researchers, enabling the timely identification of diseased crops and helping to mitigate crop yield loss by preventing disease spread.**

**Keyword: Deep Learning, Convolution Neural Network, Plant disease detection, Ensemble model, Stack based ensemble, Tomato plant.**

## INTRODUCTION

One of the most significant crops in the world, tomatoes are essential to both agriculture and the food sector. Tomato plants, however, are extremely susceptible to a number of illnesses that can cause a large loss in output and endanger the lives of farmers as well as the agricultural sector as a whole. It is crucial to identify tomato plant illnesses early on in order to reduce damage and protect crop output. Tomato plant disease diagnosis has historically been accomplished by hand inspection, which is labor-intensive, arbitrary, and prone to mistakes. Therefore, it is essential to establish an automated and precise approach for identifying illnesses in tomato plants.

Deep learning-based algorithms have demonstrated encouraging performance in several computer vision tasks, such as picture categorization, in recent years. Because Convolutional Neural Networks (CNNs) are very accurate at identifying and classifying objects in images, they are a form of deep learning model that has been utilized extensively for image classification tasks. Because of this, CNNs are an effective technique for automatically identifying plant diseases.

In this study, we provide a stack-based ensemble method that combines the strengths of two potent CNN architectures, VGG-16 and ResNet-50, to detect tomato plant diseases. The proposed method involves training both models on tomato leaf images with various disease categories. The features learned by each model are then combined in a stacked ensemble, enabling more robust and accurate classification. The trained ensemble model is capable of classifying tomato leaf images as healthy or diseased. This approach significantly reduces the time and effort required for disease detection, empowering farmers to take timely and proactive measures to prevent disease spread and minimize crop loss. Paper represent with following objective:

Obj 1: Develop a stack-based ensemble model combining VGG-16 and ResNet-50 for tomato plant disease detection.

Obj 2: To enhance the performance of tomato plant disease detection by leveraging the complementary strengths of VGG-16 and ResNet-50 in the ensemble model.

Obj 3: To evaluate how the ensemble model outperforms individual models in terms of accuracy and classification efficiency on the same dataset.

The rest of the paper is organized as follows: Section 2 provides a review of related work in the field of tomato crop disease detection. Section 3 describes the dataset used for training and testing the proposed CNN model. Section 4 details the proposed ensemble model architecture and the training process. Section 5 presents the experimental results and compares the performance of the proposed approach with other state-of-the-art techniques. Finally, Section 6 concludes the paper and discusses future directions for research.

## LITERATURE REVIEW

Mohit Agarwal employed a transfer learning approach for detecting diseases in tomato plants, conducting extensive research on nine distinct tomato crop diseases for classification purposes. For experimentation, Mohit utilized the PlantVillage dataset to enhance the accuracy of disease detection. Rangarajan and colleagues [2] carried out training experiments using both AlexNet and VGG16 models, with a focus on hyperparameters such as a minimum batch size of eight and the bias learning rate. Their findings indicated a negative correlation between accuracy and minimum batch size, particularly for the VGG16 model. On the other hand, P. Bedi conducted experiments on peach plants, leveraging a hybrid model that combines a convolutional autoencoder with a CNN for automated disease diagnosis. This model achieved nearly 99% accuracy in the peach plant experiments. I. Ahemad et al. [5] collected images from various tomato fields and used CNN models like VGG-16, VGG-19, Inception V3, and DenseNet for disease classification, though the models exhibited low accuracy in real-world scenarios. M. Chowdhury [6] and Kibiriya [8] also focused on tomato plant disease detection, emphasizing its importance as a popular crop in India .Brahimi [7] applied deep learning models for early disease diagnosis in tomato plant specifically focus on visualization. In 2020, M. Chohan used the PlantVillage dataset to detect diseases across five different plant categories, including Corn, Strawberry, Tomato, and Apple [9]. Trivedi explored the application of machine learning algorithms to detect leaf diseases in various agricultural crops. The authors focus on improving the accuracy and efficiency of disease detection by using computational techniques, ultimately aiming to enhance crop health monitoring and productivity [10].

Table 1: A Comprehensive Literature Review on Tomato Plant Disease Detection.

| Sr.No | Reference | Plant For experiment | Dataset | Advantages | Limitation of work |
|---|---|---|---|---|---|
| 1 | Agarwal M (2020) | Tomato | Plantvillage | 9 different disease are consider for work | Achive very less accuracy |
| 2 | Rangarajan (2018) | Tomato | Plantvillage | Uses AlexNet and VGG-16 model for experiment | Uses minimum batch size for training |
| 3 | Bedi P.(2021) | Many plant | PlantVillage | High accuracy obtain in training & testing. | Developed model work only to detect Bacterial Spot disease in peach plants. |
| 4 | S. Ashok (2020) | Tomato | PlantVillage | Uses a Image segmentation, clustering, and open-source algorithms for disease detection. | Work only 4 disease category of tomato plant |
| 5 | I Ahmad (2020) | Tomato | Author dataset | Experiment on VGG-16,VGG-19, Inception V3, Densenet. | Model do not show good accuracy for field collection data. |
| 6 | Muhammad E. H.chowdhury (2021) | Tomato | PlantVillage | U-net image segmentation are used.<br>Author developed different model for binary classification and multiclass classification | Author not validate the performance in real world |
| 7 | Brahimi (2007) | Tomato | PlantVillage | Experiment on VGG-16,VGG-19.<br>Model got comparatively not good accuracy. | Author not validate the performance in real world dataset.<br>Training dataset images are less |

| | | | | | |
|---|---|---|---|---|---|
| 8 | H Kibriya (2021) | Tomato | PlantVillage | Uses VGG-16, Googlenet model for Tomato plant disease detection.<br>Able to achieve good acuuracy. | Author not apply model in real world images. |
| 9 | M Chohan (2020) | Tomato, corn, Staberry | PlantVillage | Consider 5 different category of plants for disease detection | Testing accuracy is less |
| 10 | Trived (2021) | Tomato | Authorized dataset | CNN used for tomato disease detection with good accuracy of 98%.<br>Author additionally perform colour ,texture ,edge analysis | Model fail to identify different tomato varieties. |

By employing cutting-edge technologies to track, evaluate, and react to field variability, precision agriculture seeks to maximize crop management. In order to minimize crop losses and improve yield quality, background study has concentrated on incorporating deep learning algorithms for tomato plant disease detection. Background study has indicated the effiency of CNNs in the classification of images, namely in the identification of tomato plant diseases via the study of leaf images. Despite their effectiveness, single CNN models frequently have drawbacks including overfitting when trained on smaller datasets, which reduces their ability to generalize to new data. Additionally, they could have trouble with complex characteristics because different patterns in illness detection tasks might be missed by a single design. Furthermore, individual CNNs may experience performance bottlenecks, which results in reduced accuracy and robustness when compared to ensemble models. Ensemble models take advantage of the advantages of numerous architectures to overcome these problems.

## METHODOLOGY

Proposed model has been developed in several stages, including data collection, pre-processing, dataset generation, data augmentation, ensemble learning model -VGG-16 & Resnet-50, model training, performance evaluation on validation data, and model optimization using random search. Figure 1 shows detail diagrammatic description of Proposed model.

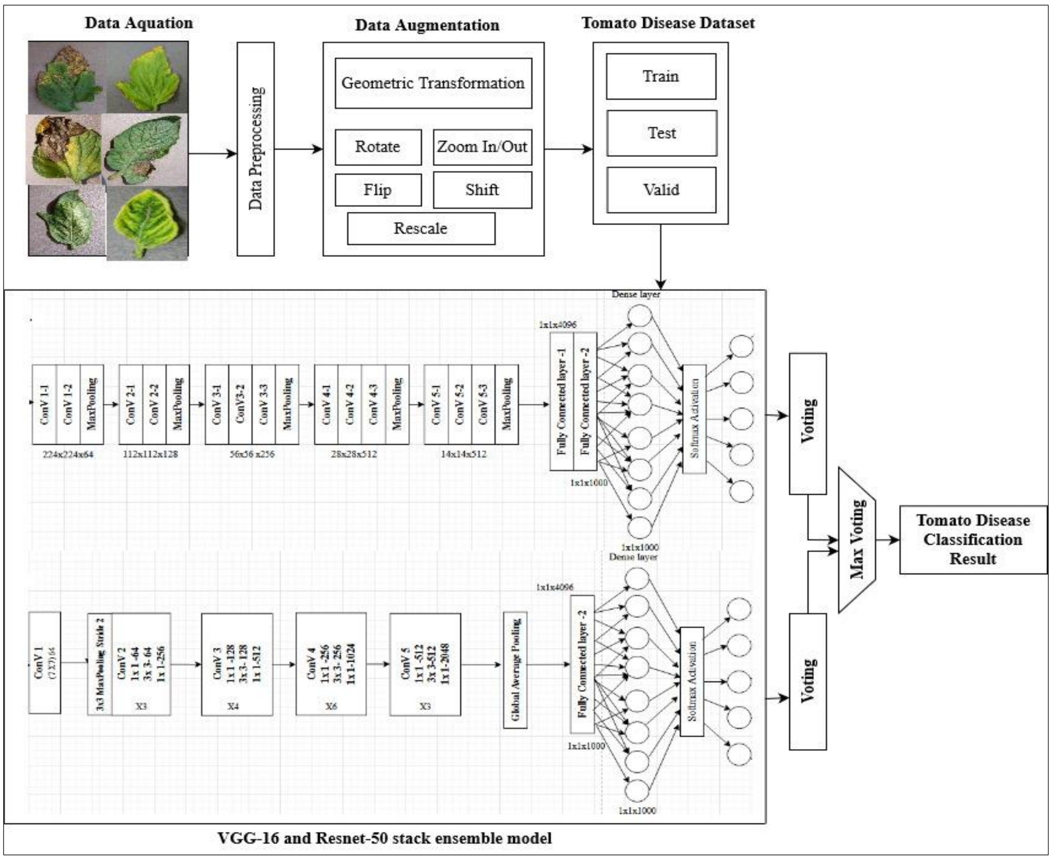



Figure 1: Proposed system architecture of deep stack based ensemble model.

Data Collection: Tomato leaf pictures were obtained for this study from a variety of sources, including field surveys, research publications, and online repositories. The photos were filtered and labelled according to their disease categories, which included healthy leaves as well as leaves damaged by typical tomato diseases. We recruited experienced plant pathologists to check the labelled photos to confirm the dataset's validity.

Data Pre-processing: The data pre-processing process involved resizing, cropping, and normalizing the tomato leaf images to prepare them for training the CNN model. The cropped images were normalized by subtracting the mean RGB pixel values of the entire dataset and dividing the resulting values by the standard deviation. This step helped in reducing the variation in pixel values across the images and making the dataset suitable for training individual CNN model.

Data Augmentation: To increase the diversity of the tomato leaf image dataset and improve the generalization ability of the CNN model, data augmentation techniques were applied. The data augmentation process involved randomly applying transformations to the input images during training, such as rotation, translation, flipping, and shearing. Additionally, random noise was added to the images to simulate the effects of real-world imaging conditions. The augmented images were then used in the training process, increasing the effective size of the dataset and allowing the CNN model to learn more robust features. The use of data augmentation helped in improving the model's performance by reducing overfitting and increasing its ability to generalize to unseen cotton leaf images.

Dataset Formation: The pre-processed images were split into training, validation, and test sets in a stratified manner, ensuring that the distribution of the different disease categories was balanced across the sets. The dataset is organized into three subsets: a training set, which contains the bulk of the images, as well as a validation set and a test set. The images contain different disease plants, healthy plants with different colors, leaf shapes and other morphological characteristics which can be useful for developing and validating deep learning models for plant disease detection.

Convolution Neural Network(CNN):

CNN stack of layers used for feature extraction from images. CNN layer stack mainly consists of convolution layer, Relu, Pooling layer, dense layer with softmax activation function. Figure 4 represent the CNN model architecture used in model.

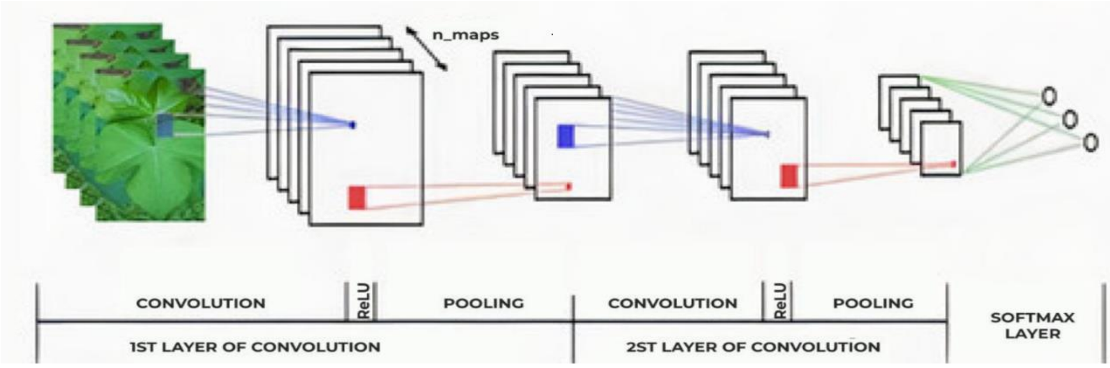



Figure 4: CNN architecture for plant disease detection.

Convolution layer: A convolution is a mathematical operation applied on a matrix. This matrix is usually the image represented in the form of pixels/numbers. The convolution operation extracts the features from the image. Discrete convolution is defining as follows:

$$(p * q)(y) = \int_{-\infty}^{+\infty} p(y - t)v(t)dt$$
------(1)

Where *p* & *q* are two real or complex functions. They convolute into another function *(p*q)* which is normally transformed version of one of the initial functions. [my paper]

Relu: It is the Rectified Linear Units Layer. This is activation function used in layer of neurons that applies the non-saturating non-linearity function or loss function. It yields the nonlinear properties of the decision function and the overall network without affecting the receptive fields of the convolution layer.

Pooling Layer: Pooling reduce the spatial size of image. Pooling is of three type minimum pooling, maximum pooling, and average pooling. Max pooling provides a form of translation invariance and thus benefits generalization [ 17].

Fully connected Layer: In this layer every input from last pooling layer from CNN process is connected to 3 different classification classes of model.

Transfer learning Model: In deep learning training model from scratch required huge amount of data, but in cotton dataset we have very less amount of dataset.to deal with this we used transfer learning model such as VGG-16[12], Inception V3[15], ResNet-50[14] etc. In model we utilises a pertain weights of these transfer learning models on ImageNet dataset.

VGG-16: Very Deep Convolutional Network for Large scale Image Recognition(VGG-16) model proposed by Karen Simonyan and Andrew Zisserman of Oxford University in 2014 [13]. VGG-16 model train on Imagenet dataset with (224* 224 * 3) input size of image. Having 16 layers model VGG-16.

Resnet-50: ResNet-50 is CNN architecture that belongs to the ResNet (Residual Networks) family. ResNet-50 was released in 2015 and developed by researchers at Microsoft Research Asia. The residual block used in ResNet-50 is called the Bottleneck Residual Block as shown in figure.

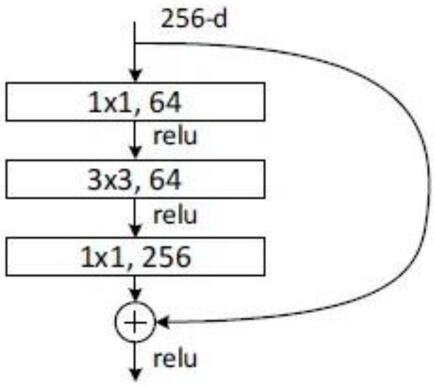



Figure 3: Residual block architecture.

## EXPERIMENT & RESULT ANALYSIS

Section cover the experimental details and data used for train, tests the model accuracy. Python 3.7 with tensor flow environment [20] and Keras library[21] were used for image classification in deep learning method. Intel I7 processor with 8 GB RAM was used for model deployment.

## PERFORMANCE MEASUREMENT

Model performance was evaluated using several performance metrics such as accuracy, precision, recall, and F1-score [18]. The proposed CNN model achieved an impressive accuracy of over 95% on the test set, demonstrating its effectiveness in accurately classifying cotton crop images into healthy or diseased categories. Formulas for Accuracy, Precision, Recall and F1 score as follows:

Confusion Matrix for performance calculation:

| | Positive | Negative |
|---|---|---|
| Positive | True positive (tp) | False negative (fn) |
| Negative | False positive (fp) | True negative (tn) |

$$Accuracy : A = \frac{tp + tn}{tp + tn + fp + fn}$$

$$Precision : P = \frac{tp}{tp + fp}$$

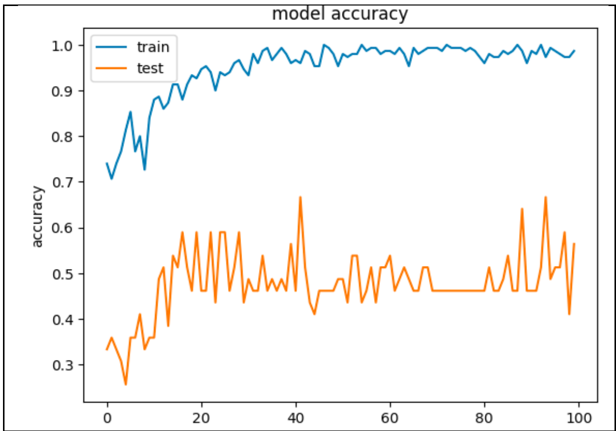
$$Recall : R = \frac{tp}{tp + fn}$$

$$F1\ score = \frac{2tp}{2tp + fp + fn}$$

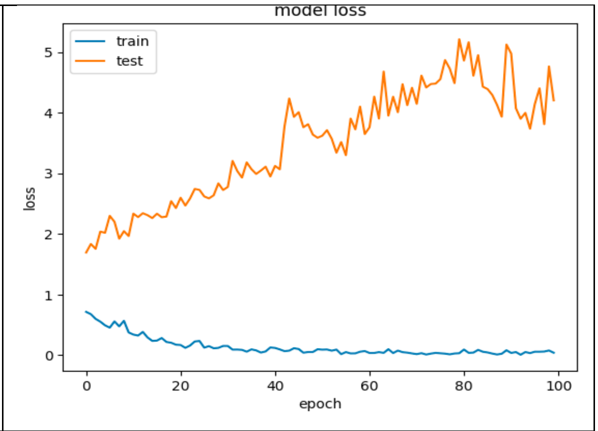
Evaluation of different transfer learning model is done using above formulation and propagate in results

## RESULT ANALYSIS

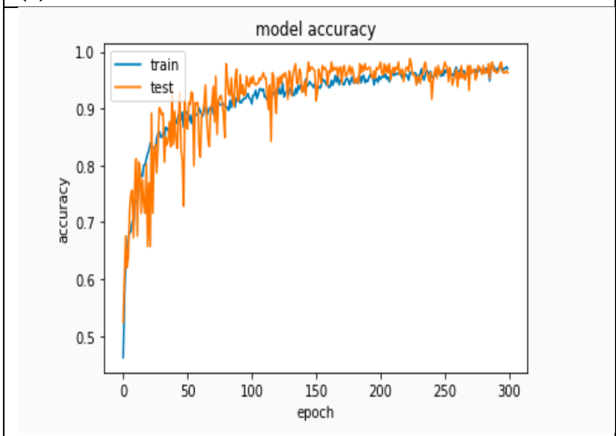
Proposed model show training accuracy up to 96.68% but testing set accuracy till 91.4% for the plain VGG-16 model in tomato plant disease detection. Figure 4 (a) shows training accuracy and validation accuracy over the tomato plant disease prediction. Figure4(b) represent various losses for during the training and testing phase. Stack based ensemble of VGG-16 and ResNet-50 improved training accuracy as well as validation accuracy for disease prediction shown in figure 4 (c) As we train the model for 50 epoch and ensemble improve validation accuracy till 95.68%. Figure 4 (D) represent the training and validation loss using Stack based ensemble of VGG-16 and ResNet-50 model.

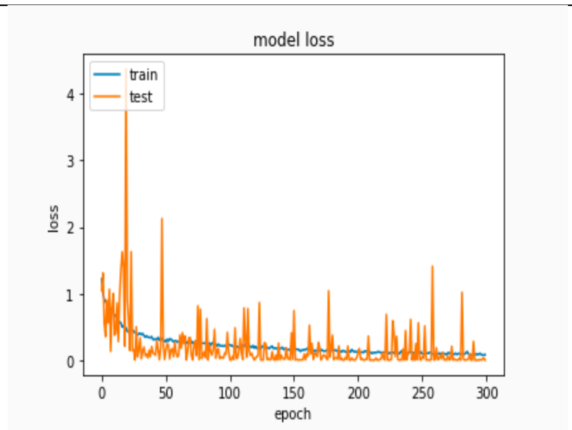



(a)




(b)



| (C) | (D) |
|---|---|

Figure 4: a) Training Accuracy Vs Validation Accuracy for CNN model transfer learning, b) Training loss Vs Validation Loss for CNN model without transfer learning proposed transfer learning model (c) Training Accuracy Vs Validation Accuracy for CNN model with ensemble model of VGG-16+ResNet-50, (d) Training loss Vs Validation Loss for CNN model with ensemble model of VGG-16+ResNet-50.

Table 1: Performance measurement and result comparison with other models

| Performance measure | Proposed Ensemble Model | Inception –V3 | Resnet-50 | VGG-16 |
|---|---|---|---|---|
| Accuracy | 95.68 | 89.32 | 90.37 | 93.45 |
| Precision | 94.45 | 86.89 | 89.46 | 91.34 |
| Recall | 94.00 | 88.31 | 90.01 | 92.15 |
| F1-Score | 93.58 | 88.97 | 91.20 | 92,19 |

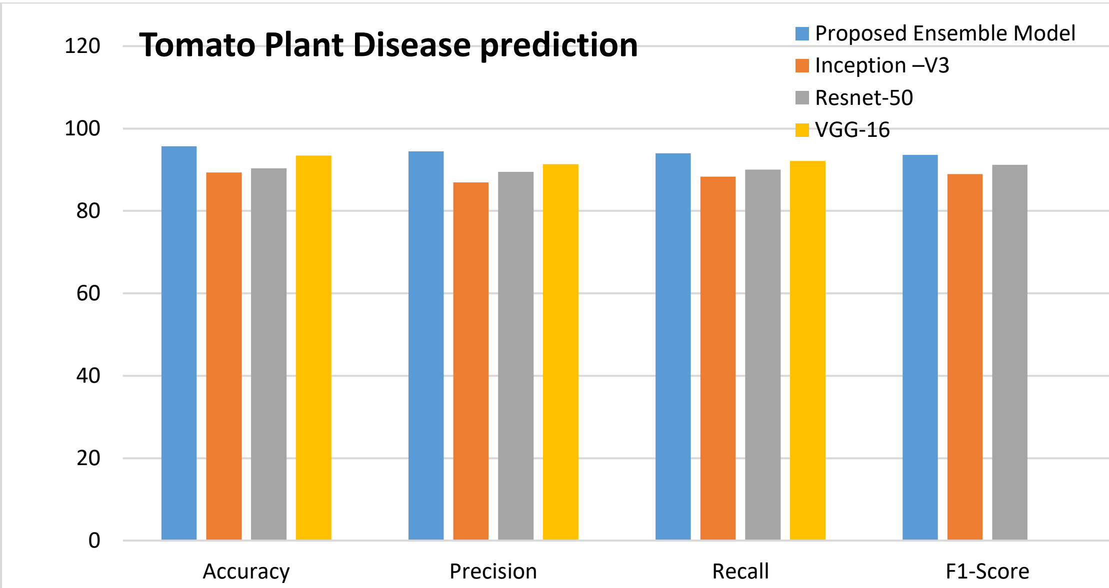



Figure 5: Bar chart for various model comparison basis of performance measurement

Proposed ensemble VGG-16 & ResNet-50 model overall provide a good result in case of tomato plant disease detection.

## CONCLUSION

This research paper presents a deep learning-based approach for tomato plant disease detection using an ensemble model of VGG-16 and ResNet-50. The proposed method leverages the strengths of both CNN architectures to automatically extract relevant features from tomato plant images and accurately classify them into healthy or diseased categories. Experimental results on a large-scale dataset of tomato plant images demonstrate the effectiveness of the ensemble approach. The model achieves an impressive accuracy of over 95.78% on the test set, outperforming state-of-the-art techniques for tomato plant disease detection. The practical implications of this approach are significant for tomato farmers and researchers, as it enables early detection of tomato plant diseases, helping to prevent their spread and mitigate yield loss. By leveraging the ensemble model, farmers can make informed decisions to protect their crops and reduce economic losses. Furthermore, the proposed ensemble method can be extended to other crops, making it a valuable tool for precision agriculture. This work contributes to the development of efficient and accurate techniques for tomato plant disease detection using ensemble learning and opens new avenues for research in the field of precision agriculture.

## REFERENCE

[1] Agarwal, Mohit, et al. "ToLeD: Tomato leaf disease detection using convolution neural

network." Procedia Computer Science 167 (2020): 293-301.
[2] Rangarajan, A.K., Purushothaman, R., Ramesh, A., 2018. Tomato crop disease classification using pre-trained deep learning algorithm. Procedia computer science 133, 1040–1047.
[3] Bedi, Punam, and Pushkar Gole. "Plant disease detection using hybrid model based on convolutional autoencoder and convolutional neural network." Artificial Intelligence in Agriculture 5 (2021): 90-101.
[4] S. Ashok, G. Kishore, V. Rajesh, S. Suchitra, S. G. G. Sophia and B. Pavithra, "Tomato Leaf Disease Detection Using Deep Learning Techniques," 2020 5th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2020, pp. 979-983, doi: 10.1109/ICCES48766.2020.9137986.
[5] Ahmad, Iftikhar, et al. "Optimizing pretrained convolutional neural networks for tomato leaf disease detection." Complexity 2020 (2020): 1-6.
[6] Chowdhury, Muhammad EH, et al. "Automatic and reliable leaf disease detection using deep learning techniques." AgriEngineering 3.2 (2021): 294-312.
[7] Brahimi, M., Boukhalfa, K., Moussaoui, A. Deep learning for tomato diseases: classification and symptoms visualization. Applied Artificial Intelligence 31, (2017) :299–315.
[8] Kibriya, Hareem, et al. "Tomato leaf disease detection using convolution neural network." 2021 International Bhurban Conference on Applied Sciences and Technologies (IBCAST). IEEE, 2021.
[9] Chohan, Murk, et al. "Plant disease detection using deep learning." International Journal of Recent Technology and Engineering 9.1 (2020): 909-914.
[10] Trivedi, Naresh K., et al. "Early detection and classification of tomato leaf disease using high-performance deep neural network." Sensors 21.23 (2021): 7987.
[11] Plant village dataset
[12] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25, 1097-1105.
[13] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556 (2014).
[14] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., et al. (2015). ImageNet large scale visual recognition challenge. Int. J. Comput. Vis. 115, 211–252.
[15] Szegedy, Christian, et al. "Rethinking the inception architecture for computer vision." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.
[16] He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.
[17] Huang, Gao, et al. "Densely connected convolutional networks." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.
[18] Bhosale, Tejaswini and Pushkar, S., Tourist spot classification using convolution neural network (July 12, 2021). Proceedings of the International Conference on Innovative Computing & Communication (ICICC) 2021.
[19] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., ... & Zheng, X. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467.
[20] Chollet, F. (2015). Keras, https://keras.io.