Detection and Classification of Leaf Diseases in Tomato Plants and Recommendations for Controlling the Spread

Sarakesh R.[1], Jenila Livingston L. M.[1][a], Rajkumar S.[1][b] and Agnel Livingston L. G. X.[2][c]
[1]School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, TN, India
[2]St. Xavier's Catholic College of Engineering, Nagercoil, TN, India

Keywords: Leaf Disease Detection, Deep Learning Techniques, Disease Classification, Convolutional Neural Network, Ensemble Model, Agricultural Practices, Crop Yield.

Abstract: Leaf disease in tomatoes is the most important factor influencing crop output quantity and quality, hence proper diagnosis and classification are essential. Different diseases affect tomato production. This study focuses on employing deep learning techniques for the disease's detection in tomato plants. Despite tomatoes being a versatile ingredient highly sought after year-round, the significant annual loss in crop yields due to diseases poses a substantial challenge in cultivation. The objective is to create a system capable of precisely identifying various tomato diseases by analysing images. The dataset utilized in this study encompasses different types and stages of tomato diseases, including Bacterial Spot, Early or Late Bright, Leaf Mold, Spider Mites, Target Spot, Septoria Leaf Spot, Mosaic and Yellow Leaf Curl Viruses. Upon disease identification, the study presents information on methods to control its spread. The models employed in this study include Convolutional Neural Network (CNN), DenseNet169, and an ensemble model combining pre-trained CNN and DenseNet169. The classification results of the study demonstrated an accuracy of 95% for the ensemble model, surpassing the accuracy of individual models. This success in recognizing diseases in tomato plants holds promise for enhancing agricultural practices.

SCIENCE AND TECHNOLOGY PUBLICATIONS

1 INTRODUCTION

The intersection of deep learning and agriculture has ushered in a transformative era for the detection and identification of plant diseases. As the global population burgeons, the imperative to ensure food security has never been more critical, underscoring the need to safeguard the health of crops, the bedrock of our sustenance. Deep learning, a sophisticated facet of artificial intelligence, emerges as a formidable solution, employing intricate neural networks to scrutinize images and plant data with unprecedented precision. This technological innovation facilitates the early identification of diseases, empowering farmers to implement timely preventive measures and thwart potential threats.

This project is aimed to harness the capabilities of deep learning algorithms, offering a nuanced and accurate discernment of specific plant issues. By mitigating the need for extensive chemical interventions, the initiative embraces sustainable farming practices, fostering an agricultural landscape that is both productive and environmentally responsible. This pioneering approach not only augments agricultural efficiency but also champions sustainability, contributing to a robust and resilient global food supply chain. The profound implications of these advancements resonate far beyond fields and farms, promising a future where crops are shielded, and environmental impacts are curtailed, heralding a new chapter in agriculture that harmonizes productivity with ecological well-being.

2 RELATED WORK

Muammer Türkoğlu et. al. (2022) developed a CNN Ensemble to detect plant diseases and pests. The

[a] https://orcid.org/0000-0002-6333-5751
[b] https://orcid.org/0000-0001-5860-7161
[c] https://orcid.org/0000-0002-2222-1643

R., S., L. M., J. L., S., R. and L. G. X., A. L.
Detection and Classification of Leaf Diseases in Tomato Plants and Recommendations for Controlling the Spread.
DOI: 10.5220/0012884700004519
Paper published under CC license (CC BY-NC-ND 4.0)
In *Proceedings of the 1st International Conference on Emerging Innovations for Sustainable Agriculture (ICEISA 2024)*, pages 151-160
ISBN: 978-989-758-714-6
Proceedings Copyright © 2025 by SCITEPRESS – Science and Technology Publications, Lda.

findings for deep feature extraction outperformed the traditional classifiers. The work based on deep feature extraction and classification with fine-tuned CNN including fc6 layer of the AlexNet, Loss3 layer for GoogleNet and fc1000 layer for ResNet50, ResNet101 and DenseNet20. The majority voting ensemble model attained the highest level of accuracy (97.56%), next to the early fusion ensemble model (96.83%).

Norhalina Senan et. al., (2020) proposed a model that can reliably recognize the affected and healthy paddy leaves, which is useful in automated paddy categorization applications. The findings show that the proposed CNN model outperformed (83% accuracy) traditional classification techniques in paddy leaf disease detection and classification.

Yong et. al. (2020) created the Inception-ResNet-v2 model for early identification of pests. Experiments demonstrated the recognition accuracy of 86.1% and the results reveal that this hybrid network model has a greater recognition accuracy than the classic model and may be used to successfully detect and classify the plant diseases and insect pests.

Morteza Khanramaki et. al. (2021) developed an ensemble technique for identifying citrus pests that outperformed competing methods. Data augmentation increases the quantity of pictures in the dataset, which enhances classifier generalizability. For the experimental analysis, a 10-fold cross validation was performed to determine accuracy, and it obtained 99.04%.

Lucas et al. (2021) implemented an integrated CNN architecture that combines instance segmentation with a Mask R-CNN and semantic segmentation with UNet and PSPNet to detect diseases and pests in coffee leaves. The MIoU for the UNet and PSPNet networks was 94.25% and 93.54%, respectively. The two networks produced very similar results, with the UNet slightly outperforming the PSPNet. However, PSPNet can be selected since its lesion marker extends somewhat beyond its edge, which can assist in lesion categorization, as the intersection of the lesion and the healthy portion of the leaf is not always immediately identifiable.

Several studies used neural networks to identify and classify diseases. Earlier research employs shape, color, and texture feature extraction approaches, as well as typical machine learning classifiers. In more recent investigations, CNN-based models have shown significant success in the automated detection of plant diseases and pests in leaves (Lu 2017, Liu 2017, Wallelign 2018, Picon 2019, Zhang 2019, Rahman 2020, Wang 2020).

3 PROPOSED WORK

The existing manual methods for predicting disease in plants are often crucial, labour-intensive, time-consuming, lack of accuracy, not scale effectively to meet the demands of large-scale agriculture. So the objective of this research is to overcome these challenges by utilizing the potential benefits of deep learning techniques. The primary objective is to design and develop an advanced deep learning-based system capable of automatically detecting and identifying plant diseases from images of plants.

This system will utilize CNNs and other deep learning architectures to analyze visual data, providing farmers with rapid, precise, and scalable solutions for monitoring crop health. Ultimately, this research will contribute to reducing yield losses, promoting sustainable agriculture, and enhancing global food production.

3.1 Research Challenges

Finding effective data augmentation and preprocessing strategies to enhance image quality, remove noise, and improve model robustness. Designing models and algorithms that can scale to handle large volumes of agricultural images efficiently for timely detection of plant diseases.

Deep learning models often struggle with generalizing their knowledge to new and unseen conditions. For plant disease and pest detection, models need to perform well across different seasons, regions, and plant species. Achieving this level of generalization while maintaining high accuracy is a significant research challenge.

Enhance the interpretability of deep learning models for plant disease detection. Understanding how models make decisions is crucial for gaining trust in their recommendations, especially in agricultural decision-making.

3.2 Scope of the Project

In this research, we develop a comprehensive system for the detection and identification of diseases in plants through deep learning techniques. The scope encompasses the collection of diverse plant data, the implementation of advanced models, emphasizing scalability and ethical considerations. It also involves exploring novel methods.

3.3 Data Preprocessing

Rescale: Rescaling involves adjusting the pixel values of the images to fit within a specific range, typically [0, 1] to ensure that all pixel values are proportionate to each other. This process ensures uniformity in pixel values across different images and helps in standardizing the data for better processing by deep learning models.

Shear Range: Shearing is a geometric transformation that distorts an image by shifting one part of it in a fixed direction. In this context, a shear range of 0.2 means that the image can be distorted by shifting parts of it by a maximum of 20% in a specific direction. Shearing is useful for introducing variations in images, which can be beneficial for tasks such as data augmentation in image classification.

Brightness Range: Adjusting the brightness of images randomly within a specified range is a technique used to augment image data. Randomly adjusting brightness helps in making the model more robust and versatile to variations in diverse lighting conditions during inference.

4 METHODOLOGY

Initially, preprocessing and augmentation were performed, and the dataset was divided into training, testing, and validation subsets. The evaluation metrics for each model were gathered through training, testing, and validation methods, allowing for a full assessment of model performance. In this study, CNN, Densenet169, and Ensemble techniques were used to detect diseases in plants (Figure 1).

4.1 Convolutional Neural Network

The adeptness of CNNs in capturing intricate spatial relationships within images makes them indispensable tools for tasks requiring nuanced visual understanding. CNNs unparalleled effectiveness is underscored by their remarkable performance in a spectrum of computer vision tasks, ranging from accurate image classification to precise object detection and nuanced image segmentation. Through the seamless integration of convolutional, pooling, and fully connected layers, CNNs stand as a cornerstone in the realm of deep learning, offering robust solutions to complex challenges in image analysis and interpretation.

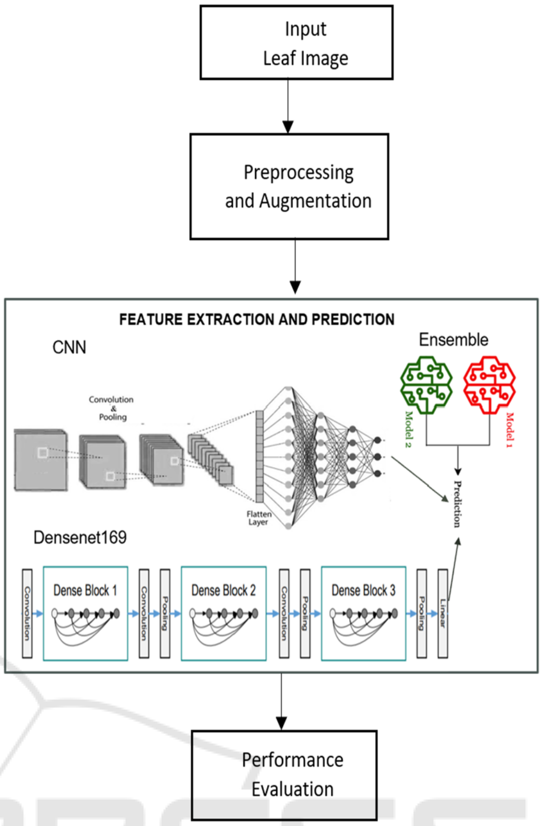


Figure 1: Proposed System Architecture

CNNs play a vital role in leveraging deep learning for disease detection in plants. The use of CNNs in this context involves the analysis of images of plants to identify signs and symptoms of diseases. Filters (also called kernels) are small learnable matrices that slide over the input data to perform element-wise multiplications, producing feature maps. The architecture diagram of CNN model is given in Figure 2.

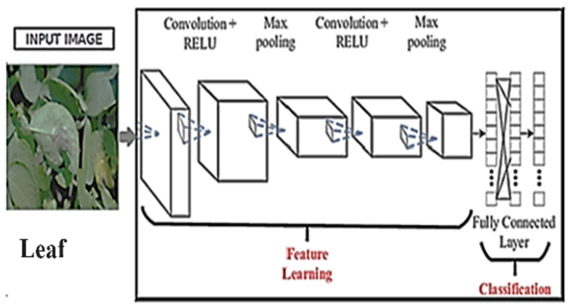


Figure 2: Architecture of CNN.

Pooling Layers: Pooling layers are used to minimize the spatial dimensions of the input volume, with typical pooling procedures including max pooling and average pooling.

Convolutional layers apply convolution operations to input data using learnable filters or kernels. These filters slide over the input, capturing local patterns and produces feature maps that represent the presence of specific features or patterns in the input data. in capturing spatial hierarchies and are crucial for tasks like image recognition, where local patterns are essential.

Flattening: Before feeding the output of convolutional and pooling layers into fully connected layers, the data is usually flattened into a vector.

Fully Connected Layers: After several convolutional and pooling layers (Figure 3), CNNs often include one or more fully connected layers for making predictions based on the learned features. Dense layers are fully connected layers, perform weighted sum operations, applying activation functions to produce non-linear mappings between inputs and outputs.

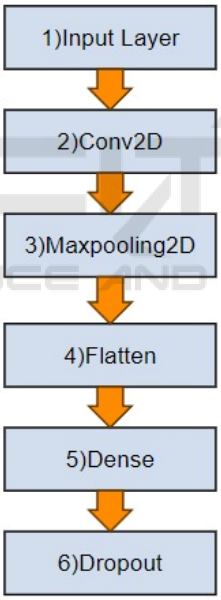


Figure 3: Layers of CNN.

Dropout: Dropout is a regularization technique commonly used in CNNs to prevent overfitting. It randomly drops a certain percentage of neurons during training to promote more robust learning.

Batch normalization is another regularization technique that normalizes the inputs of a layer, helping to stabilize and accelerate the training process.

Loss Function: The choice of a loss function depends on the specific task; for classification tasks, cross entropy loss is used

4.2 DenseNet169 Model

Densenet169 is a variation of DenseNet with 169 layers intended to create a deeper network than the original DenseNet. DenseNet169, being a deeper model within the DenseNet family, may be suitable for tasks that require capturing more intricate patterns in the data.

4.2.1 Key Features and Characteristics of DenseNet169

Dense Blocks: The network is organized into dense blocks, each containing multiple densely connected layers. Within each dense block, the output of each layer is concatenated with the feature maps of all previous layers, facilitating feature reuse and compact model representation. Dense Connectivity for Feature Extraction: Leverage the dense connectivity within DenseNet169 for effective feature extraction. The dense blocks allow for the reuse of features from previous layers, enabling the model to capture hierarchical representations of disease-related patterns in plants.

Bottleneck Layers: DenseNet169 includes bottleneck layers within dense blocks to reduce computational complexity. These bottleneck layers consist of a 1x1 convolution layer followed by a 3x3 convolution layer, which helps in efficient feature extraction. These layers help reduce the number of parameters and computational load while maintaining the capacity to capture complex patterns.

Freezing Base Layers: Freezing base layers, uses a loop to set all layers in the pre-trained DenseNet169 base model to non-trainable. This prevents these layers from being updated during the subsequent training process, ensuring that the previously learned features remain fixed and only the additional layers on top are fine-tuned for the specific task. This helps retain valuable pre-trained knowledge while adapting the model to a new classification objective.

Transition Layers: Transition layers are inserted between dense blocks to control the spatial dimensions of feature maps and manage the spatial resolution, contributing to the overall efficiency of the model. These transition layers typically consist of a 1x1 convolution layer followed by average pooling.

Global Average Pooling: DenseNet169, like other DenseNet models, employs global average pooling at the end of the network instead of traditional fully connected layers. This contributes to a fixed-size feature vector for each image, which is then used for disease classification.

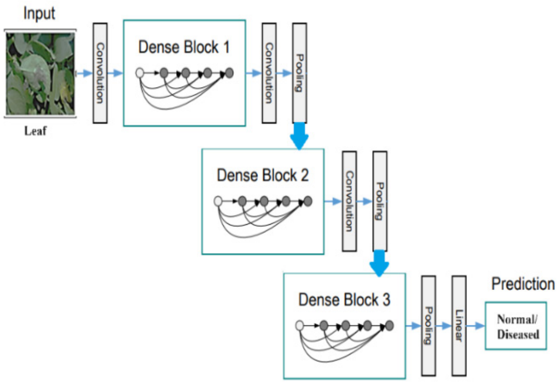


Figure 4: Architecture of Densenet169 model.

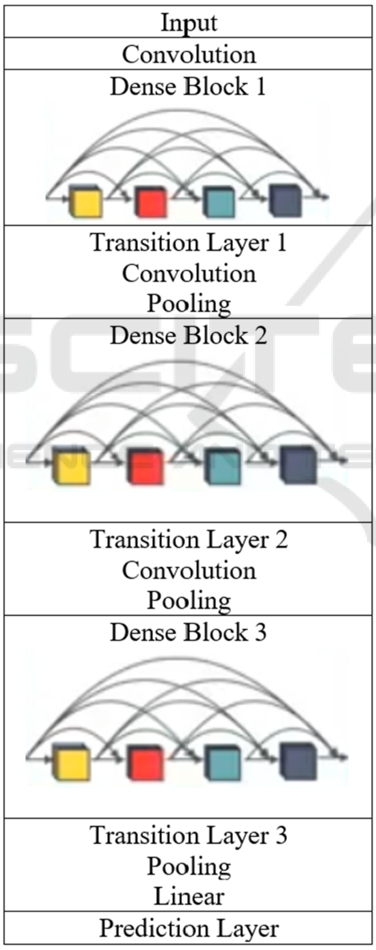


Figure 5: layers of Densenet169

Parameter Efficiency: DenseNet169 is designed to be parameter-efficient. The dense connectivity structure allows the model to achieve competitive performance with fewer parameters compared to traditional architectures.

Transfer Learning: Pre-trained versions of DenseNet169 on large datasets, such as ImageNet, are available, making it suitable for transfer learning on tasks with limited labeled data. It allows the model to leverage knowledge gained from a broader set of visual features before fine-tuning on the specific plant disease dataset.

DenseNet169 is a specific variant (Figure 4), known for its increased depth with 169 layers. It incorporates several key layers, similar to other DenseNet models. These layers collectively contribute to the unique architecture of DenseNet169. The dense connectivity, bottleneck layers, and transition layers (Figure 5) are key features that allow DenseNet to capture complex patterns in the data effectively.

4.3 Ensemble Model of CNN and DenseNet169

Ensemble models in deep learning involve combining predictions from multiple individual models to improve overall performance and generalization. Ensemble models are particularly useful when dealing with diverse data, reducing overfitting, and improving model robustness. They are often employed in situations where individual models may have different strengths and weaknesses. While creating ensemble models requires more computational resources, they can yield better generalization performance compared to individual models. The key features of ensemble model are as follows:

Bagging (Bootstrap Aggregating): Train multiple deep learning models on different subsets. Bootstrap aggregating the predictions through voting or averaging.

Boosting: In deep learning, boosting can be applied by training shallow networks sequentially, where each subsequent network focuses on the misclassifications of the previous ones. Examples include AdaBoost and Gradient Boosting.

Stacking: In deep learning, the base models can be different architectures or variations of the same architecture. The meta-model, often a simpler model, learns to weight or combine the outputs of the base models for predictions.

Weighted Average Ensembles: Assign different weights to the predictions of each model based on their individual performance. Weights can be determined through cross-validation or performance on a validation set. The final prediction is a weighted sum of the individual predictions.

Model Distillation: The student model learns not only from the ground truth labels but also from the soft labels (probabilities) provided by the teacher model. Train a larger, more complex model (teacher model) and then use its predictions to train a smaller model (student model).

4.3.1 Workings of Ensemble Model of CNN and DenseNet169

Creating an ensemble model with a combination of CNN architecture (e.g., ResNet, VGG, Inception) and DenseNet169 (Figure 6) for plant disease detection involves leveraging the strengths of each architecture to enhance overall performance of image classification.

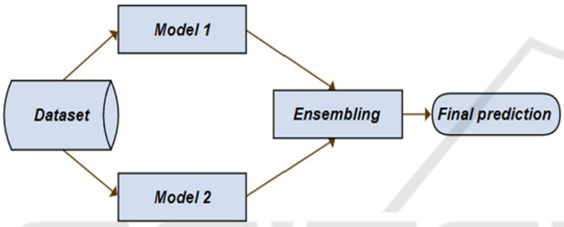


Figure 6; Architecture diagram of Ensemble model

Individual Model Training: Train the CNN and DenseNet169 models independently on the training dataset. Use transfer learning by initializing the models with pre-trained weights on large-scale datasets like ImageNet.

Model Outputs: For each input image, both the CNN and DenseNet169 models produce predictions, indicating the likelihood of the presence of a disease. If it's a binary classification task, the models output probabilities or binary predictions.

Ensemble Aggregation: Combine the predictions from both models using an ensemble strategy. Common strategies include:

Voting (Hard or Soft): For binary classification, use a majority vote for hard voting or average the probabilities for soft voting.

Weighted Averaging: Assign different weights to the predictions of each model based on their individual performance. The final prediction is a weighted sum of the individual predictions.

Ensemble Model Output: The final output of the ensemble model is the aggregated prediction from both the CNN and DenseNet169 models

5 RESULTS AND DISCUSSION

Figure 7 shows the model configuration of a CNN for plant disease detection and classification. Figures 8 and 10 show the accuracy and test loss of a CNN model, respectively.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 254, 254, 32)      896

 max_pooling2d (MaxPooling2  (None, 127, 127, 32)      0
 D)

 conv2d_1 (Conv2D)           (None, 125, 125, 64)      18496

 max_pooling2d_1 (MaxPoolin  (None, 62, 62, 64)        0
 g2D)

 conv2d_2 (Conv2D)           (None, 60, 60, 64)        36928

 max_pooling2d_2 (MaxPoolin  (None, 30, 30, 64)        0
 g2D)

 conv2d_3 (Conv2D)           (None, 28, 28, 64)        36928

 max_pooling2d_3 (MaxPoolin  (None, 14, 14, 64)        0
 g2D)

 flatten (Flatten)           (None, 12544)             0

 dense (Dense)               (None, 512)               6423040

 dropout (Dropout)           (None, 512)               0

 dense_1 (Dense)             (None, 128)               65664

 dense_2 (Dense)             (None, 10)                1290

=================================================================
Total params: 6583242 (25.11 MB)
Trainable params: 6583242 (25.11 MB)
Non-trainable params: 0 (0.00 Byte)
```

Figure 7: Model layout of CNN model

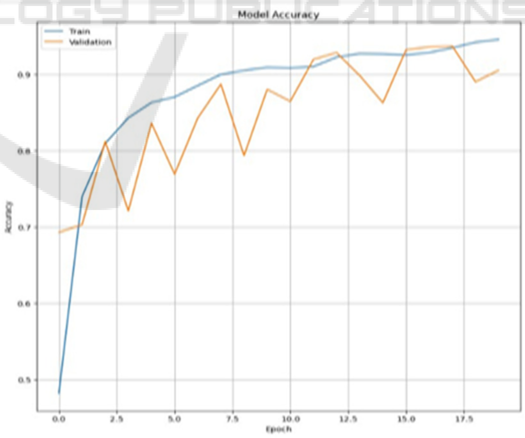
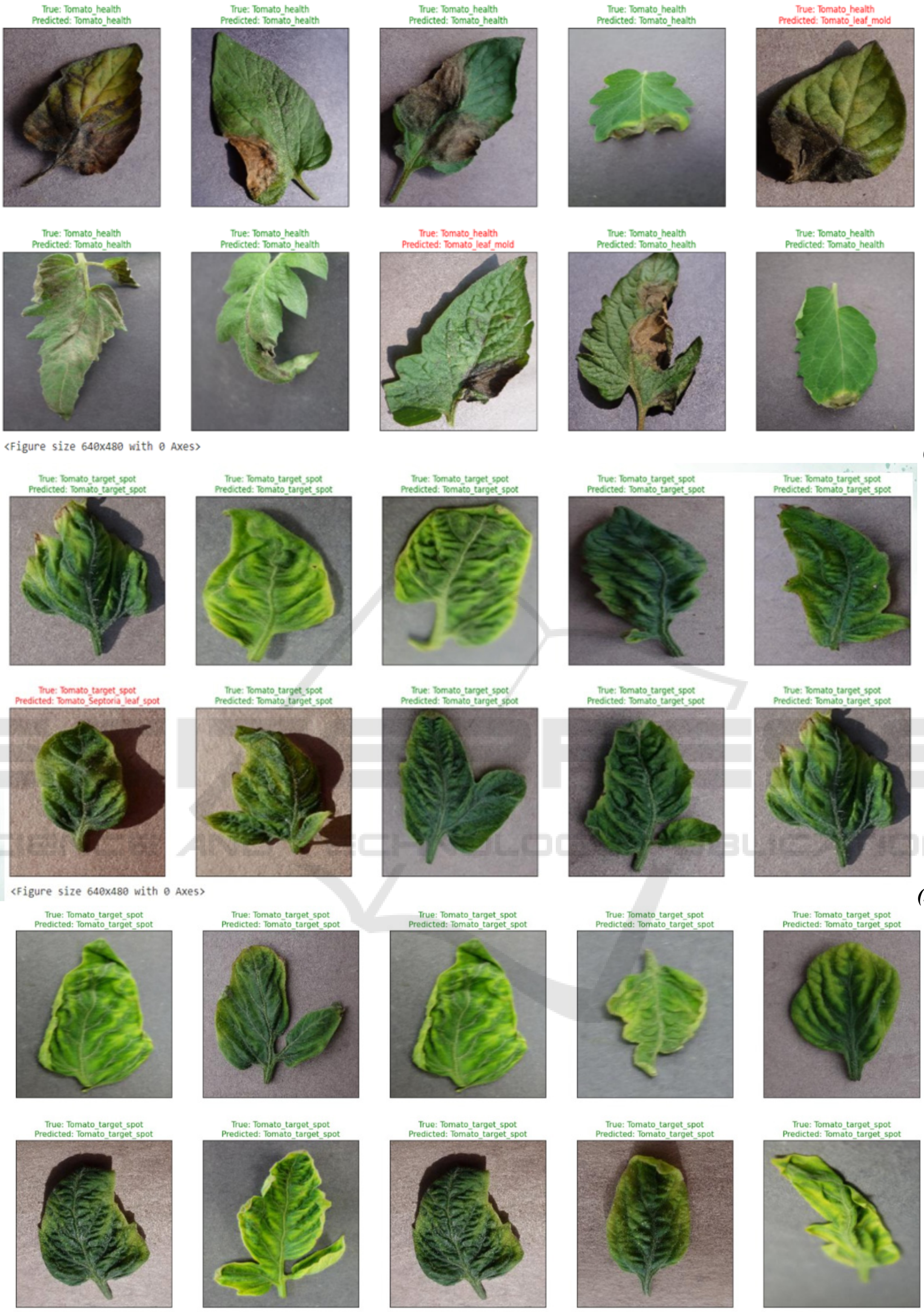


Figure 8: CNN model Accuracy

The following images (Figure 9 a,b,c) show the prediction of tomato leaf diseases with true and predicted labels using CNN, Densenet169, and Ensemble models.



(c)

Figure 9: Prediction of tomato leaf disease with true labels and predicted labels. a) Prediction using CNN b) Prediction using Densenet169 c) Ensemble model

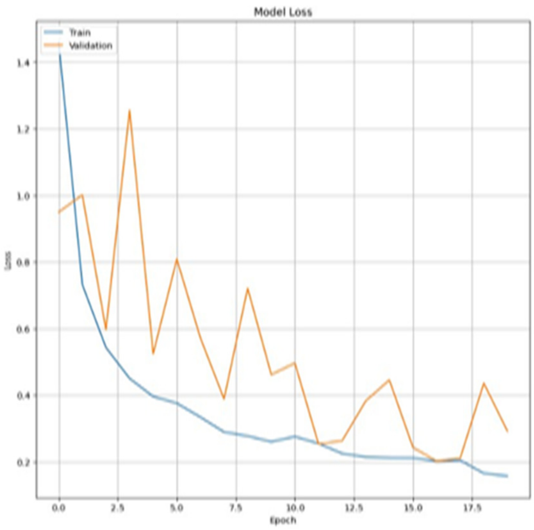
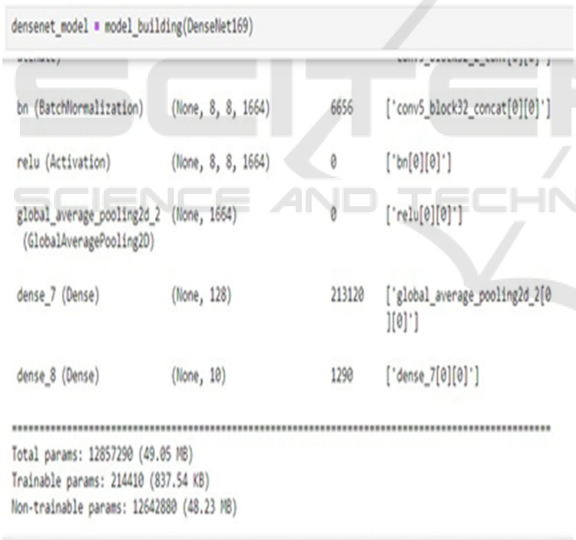


Figure 10: CNN model test loss

The below image (Figure 11) depicts the architecture of the Densenet169 model, the total params has increased compared to the previous model.



```
densenet_model = model_building(DenseNet169)

bn (BatchNormalization)      (None, 8, 8, 1664)   6656     ['conv5_block32_concat[0][0]']
relu (Activation)            (None, 8, 8, 1664)   0        ['bn[0][0]']
global_average_pooling2d_2   (None, 1664)         0        ['relu[0][0]']
 (GlobalAveragePooling2D)
dense_7 (Dense)              (None, 128)          213120   ['global_average_pooling2d_2[0
                                                           ][0]']
dense_8 (Dense)              (None, 10)           1290     ['dense_7[0][0]']

==================================================================================
Total params: 12857290 (49.05 MB)
Trainable params: 214410 (837.54 KB)
Non-trainable params: 12642880 (48.23 MB)
```

Figure 11: Model layout of Densenet169 model

A confusion matrix is a table that defines a classification model's output based on test data with known true values. Figures 12, 13 and 14 shows the confusion matrix of CNN, Densenet169, and ensemble model respectively.

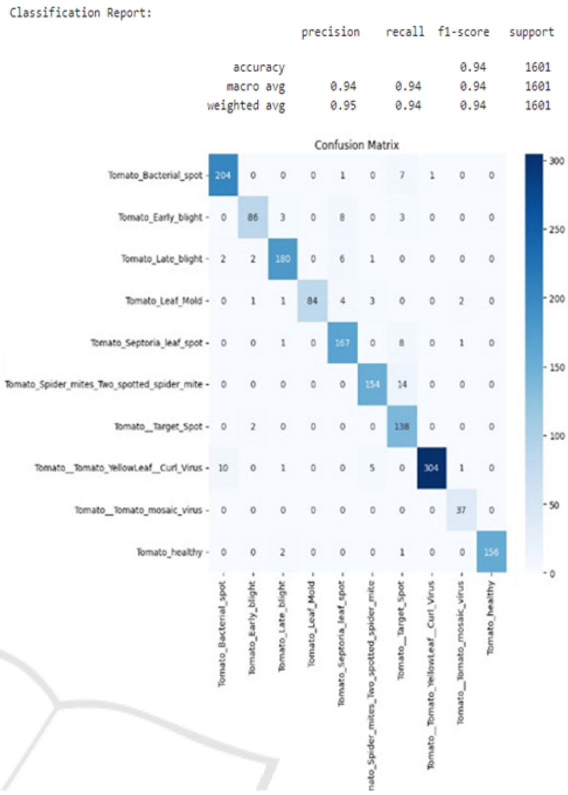


Figure 12: Confusion matrix of CNN model

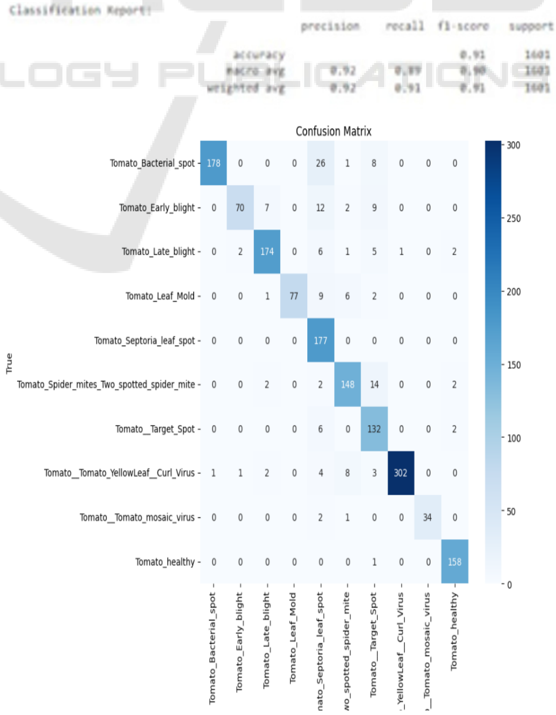


Figure 13: Confusion matrix of Densenet169 model

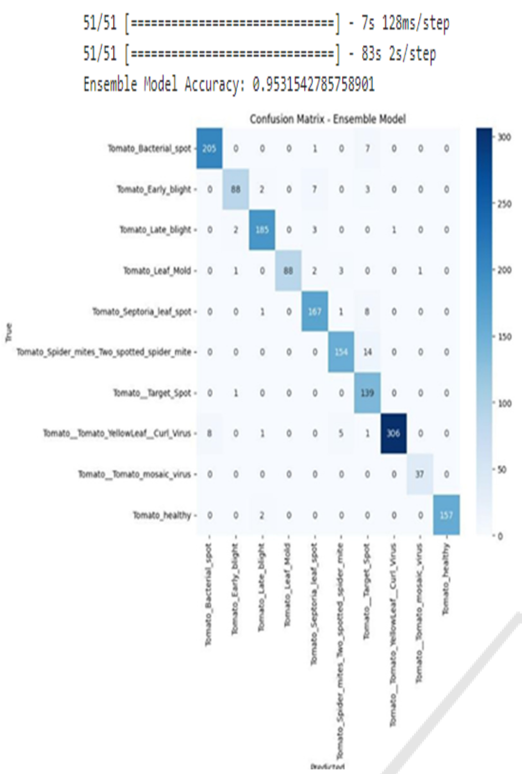


Figure 14: Confusion matrix of ensemble model

Table 1 represents the accuracy and test loss of the models. Among the three models, the ensemble model demonstrated superior accuracy in identifying plant diseases, achieving an impressive 95.3% accuracy rate. This demonstrates the usefulness of combining the strengths of CNN and DenseNet169 in a collaborative framework to improve performance.

Table 1: Accuracy and Test loss of models

Models	Accuracy	Test loss
CNN	94%	17.2%
Densenet169	91%	28.1%
Ensemble	95.3%	9.1%

6 CONCLUSION AND RECOMMENDATION

This study focused on the crucial task of detection and identification of disease in tomato plant, addressing a pressing need in agriculture for early and accurate diagnosis. Through the utilization of advanced technology such as deep learning, the project successfully developed a robust system capable of recognizing and categorizing plant diseases efficiently. The implemented solution demonstrated its effectiveness in automating the detection process, allowing for timely interventions to prevent the spread of diseases and mitigate potential crop losses. By leveraging the power of deep learning, the project not only enhanced the speed and accuracy of disease identification but also provided a scalable and adaptable framework that can be extended to various crops and regions. Furthermore, the project contributes to sustainable agriculture practices by promoting precision farming, reducing the reliance on misuse of chemical treatments, and ultimately fostering a more resilient and productive food supply chain.

This study focused solely on investigating few diseases affecting only one crop species, excluding others such as brinjal, ladies finger, chili, and their respective diseases. Hence, the next phase involves acquiring additional images of crop species and diseases for research purposes. Despite achieving commendable recognition accuracy, the models warrant further exploration and optimization. Simultaneously, there's a need to develop a network model capable of classifying crop images with greater precision.

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

REFERENCES

Turkoglu, M., Yanikoğlu, B. and Hanbay, D. Plant Disease Net: Convolutional Neural Network Ensemble for Plant Disease and Pest Detection. *SIVIP* 16, 301–309 (2022). Https://Doi.Org/10.1007/S11760-021-01909-2

Norhalina Senan , Muhammad Aamir , Rosziati Ibrahim , N. S. A. M Taujuddin, and W.H.N Wan Muda, An Efficient Convolutional Neural Network for Paddy Leaf Disease and Pest Classification, *International Journal of Advanced Computer Science and Applications*(IJACSA), Volume 11 Issue 7, (2020).

Yong Ai, C. Sun, J. Tie and X. Cai, "Research on Recognition Model of Crop Diseases and Insect Pests Based on Deep Learning in Harsh Environments," In *IEEE Access*, Vol. 8, Pp. 171686-171693, (2020), Doi: 10.1109/Access.2020.3025325.

Morteza Khanramaki, Ezzatollah Askari Asli-Ardeh and Ehsan Kozegar , Citrus Pests Classification Using an Ensemble of Deep Learning Models , *Computers and Electronics in Agriculture*, Volume 186, (2021), Https://Doi.Org/10.1016/J.Compag.2021.106192

Lucas M. Tassis, João E. Tozzi De Souza , and Renato A. Krohling, Deep Learning Approach Combining Instance and Semantic Segmentation to Identify Diseases and Pests of Coffee Leaves from In-Field Images, *Computers and Electronics in Agriculture*, Volume 186, (2021), Https://Doi.Org/10.1016/J.Compag.2021.106191

Lu, Y., Yi, S., Zeng, N., Liu, Y., and Zhang, Y.: Identification Of Rice Diseases Using Deep Convolutional Neural Networks. *Neurocomputing* 267, 378–384 (2017)

Liu, B., Zhang, Y., He, D., and Li, Y.: Identification of Apple Leaf Diseases Based on Deep Convolutional Neural Networks. *Symmetry*, 10(1), Art. 11 (2017).

Wallelign, S., Polceanu, M., and Buche, C.: Soybean Plant Disease Identification Using Convolutional Neural Network. In K. Brawner and V. Rus (Eds.), *Proceedings Of 31st International Flairs Conference*, Pp. 146–151 (2018).

Picon, A., Alvarez-Gila, A., Seitz, M., Ortiz-Barredo, A., Echazarra, J., and Johannes, A.: Deep Convolutional Neural Networks For Mobile Capture Device-Based Crop Disease Classification In The Wild. *Comput. Electron. Agric*. 161, 280–290 (2019)

Zhang, S., Huang, Z., and Zhang, C.: Three-Channel Convolutional Neural Networks for Vegetable Leaf Disease Recognition. *Cogn. Syst*. Res. 53, 31–41 (2019)

Rahman, C.R., Arko, P.S., Ali, M.E., Khan, M.A.I., Apon, S.H., Nowrin, F., and Wasif, A.: Identification and Recognition of Rice Diseases and Pests Using Convolutional Neural Networks. *Biosys. Eng*. 194, 112–120 (2020).

Wang, L., Sun, J., Wu, X., Shen, J., Lu, B., and Tan, W.: Identification of Crop Diseases Using Improved Convolutional Neural Networks. *Iet Comput. Vision* 14(7), 538–545 (2020)

SCITEPRESS
SCIENCE AND TECHNOLOGY PUBLICATIONS