**Solanum lycopersicum disease detection using VGG16 over PCA DeepNet to improve accuracy**

J. Devadarshini

Research Scholar,

Project Guide, Corresponding Author,

Department of computer science and engineering,

Saveetha School of Engineering,

Saveetha Institute of Medical and Technical Sciences,

Saveetha University, Chennai, Tamil Nadu, India. Pincode:602105

devadarshinidevadarshini4092.sse@saveetha.com

Dr Carmel Mary Belinda

Project Guide, Corresponding Author,

Department of computer science and engineering,

Saveetha School of Engineering,

Saveetha Institute of Medical and Technical Sciences,

Saveetha University, Chennai, Tamil Nadu, India Pincode:602105

carmelbelinda.sse@saveetha.com

**Abstract** : The evolution of Deep Learning and Computer Vision in the agricultural domain is a very functional tool in exposing detrimental plant diseases. Classification and detection of healthy and diseased crops play a very critical role in deciding the rate and standard of production. Thus, the existing work focuses on a well-suggested novel method of detecting Solanum lycopersicum (Tomato) leaf diseases using Deep Neural Networks to brace agro-based industries. The existing novel framework is resorted with a fusion of the traditional Machine Learning model Principal Component Analysis (PCA) and a personalized Deep Neural Network which has been termed PCA DeepNet and the VGG16 model. The blended structure also comprises of Generative Adversarial Network (GAN) for acquiring a superior combination of datasets. The detection is sustained by the Faster Region-Based Convolutional Neural Network (F-RCNN). Altogether, the task results in a classification accuracy of 99.60% with a mean precision of 98.55%; granting a favourable Intersection over Union (IOU) count of 0.95 in detection. Hence, the proffered task outmatches any other disclosed state-of-the-art.

.

**Keywords** - Deep Learning, Computer Vision, Agriculture, Principal Component Analysis, Faster Region-Based Convolutional Neural Network, Machine Learning, Deepnet

**INTRODUCTION**

Plant diseases are a root of notable crop demolition which is dangerous to the providence. Plant diseases are not only detrimental to people but also to animals. Under an evaluation, about 37% of the crops are wasted due to diseases in plants. In certain countries with agriculture as its primary occupation, most of the population depends upon the agricultural sector for income, therefore, disease detection in plants plays an indispensable role in uplifting the economy by amplifying the crop yield.

The Food and Agriculture Organization (FAO) has published research that predicts the global population is expected to grow by up to 9 billion within the next few years. To manage the food demands, the production of agro products should be increased at least by 70%. The diseases can also affect the plants and cause destruction to crops, so premature stage disease detection is important to help scale back economic loss.

In many impoverished countries, farmers typically inspect their crops using naked-eye observation, which is time-consuming and inefficient. This method is often ineffective in detecting diseases with similar patterns. However, in the past decade, new techniques for disease detection have been introduced that not only reduce the error rate but are also being used commercially. One such solution involves capturing images of leaves and using computer vision or image processing techniques to analyze them. The use of Deep Learning (DL) applications has become increasingly prevalent in the agriculture sector over the past two decades. Artificial Intelligence (AI) and Machine Learning (ML) can also be employed to analyze and diagnose plant diseases [3]. When developing an image processing solution for plant diseases, it is important to keep in mind that leaf images in fields often contain noise, which can lead to poor segmentation, feature extraction, and overall system performance. Inaccurate analysis and diagnosis can result in farmers uprooting healthy plants or excessively using pesticides, which negatively impacts the health of plants.

**Rationale**:

The rationale behind comparing VGG 16 and PCA DeepNet lies in their distinct methodologies. VGG-16, or the Visual Geometry Group 16-layer model, is a deep convolutional neural network architecture designed for image classification. Developed by the Visual Geometry Group at the University of Oxford, VGG-16 gained prominence during the ImageNet Large Scale Visual Recognition Challenge in 2014.

Key features of VGG-16 include its simplicity and uniform architecture. The model comprises 16 convolutional and fully connected layers, with small 3x3 convolutional filters throughout the convolutional layers. This uniformity aids in training and understanding the network's behavior. VGG-16 achieves impressive performance on image classification tasks by capturing complex hierarchical features through its deep architecture.

Despite its success, VGG-16's main drawback is its large number of parameters, making it computationally expensive and memory-intensive compared to more recent architectures like ResNet or EfficientNet. Nevertheless, VGG-16 remains a foundational model in the development of deep learning architectures, contributing to advancements in computer vision tasks.

On the other hand, PCA can be employed as a preprocessing step for deep learning models. This involves using PCA to reduce the dimensionality of the input data before feeding it into a DeepNet. This combined approach can offer benefits such as faster training times, reduced risk of overfitting, and improved interpretability of features.For instance, in image processing, PCA can be applied to reduce the dimensionality of image data before passing it through a convolutional neural network (CNN). This can help in handling high-dimensional image datasets more efficiently while preserving relevant information.

The integration of PCA with DeepNet is a strategy often employed to optimize performance and resource utilization, particularly when working with large and complex datasets.

By comparing these two approaches, this study aims to determine which algorithm offers superior predictive capabilities in the context of plant disease detection.

**Research Objective:**

The primary objective of this research is to evaluate and compare the effectiveness of VGG 16 over PCA DeepNet in predicting plant disease based on diverse predictive measures. By analyzing a comprehensive dataset, encompassing various clinical parameters and biomarkers, this study intends to identify the most accurate and reliable model for disease prediction. The findings of this research could significantly impact farming practices, enabling timely interventions, personalized crop care, and ultimately, reducing the burden of crop damage related morbidity and mortality.

**Significance of the Study:**

The significance of this study lies in its potential to enhance the accuracy of plant disease detection, thereby enabling farming professionals to intervene proactively. Early detection of crops at high probability of disease can lead to preventive measures, farming style modifications, and appropriate manure and fertilizer interventions, ultimately improving crop yield and reducing farming costs. Moreover, this research contributes to the growing body of knowledge concerning the application of machine learning in farming, showcasing the practical benefits of advanced analytics in the realm of agriculture and plant disease detection.

**MACHINE LEARNING IN PLANT DISEASE DETECTION**

A typical deep learning model for image classification, such as a convolutional neural network (CNN), typically consists of an input layer, one or more hidden layers, and a classification layer. The architecture of the DNN. The proposed system employs CNN models, such as DeepNet and VGG16, to detect diseases in plants using the Plant-Village dataset.

**Dataset:**

The training and testing of the proposed model is conducted using the Plant-Village dataset, which contains a total of 10,735 images of tomato leaves. The dataset is divided into a training set, which comprises 70% of the total images, and a testing set, which contains the remaining 30% of the total images.

**Image Preprocessing:**

Before feeding the images into the CNN, two preprocessing steps are performed. First, the images are resized to match the size of the input layer of the CNN. Secondly, the images are denoised using a Gaussian blur filter.

**Transfer Learning:**

In transfer learning, the model is improved by retraining the CNN with another dataset, typically with a larger amount of data, after it has already been trained on a different task. The CNN architecture consists of multiple layers that perform specific tasks, such as segmentation, feature extraction, edge detection, etc. As shown in Fig. 3, the VGG16 architecture consists of Convolution layers, Max-pooling layers, ReLU activation function, Fully connected layers, and a Softmax layer, while the DeepNet architecture consists of Convolution layers, ReLU activation function, Max-pooling layers, Dropout layers, and a Fully connected layer.

In this study, the deep learning frameworks in Python and Matlab were utilized. Specifically, CNN frameworks from Matlab were employed, along with the weights learned from the ImageNet dataset. The convolution layer is a fundamental component of CNN, and it generates feature maps by using convolution filters. The image pixels are convoluted with the filters through a dot product between kernel pixels and image pixels. In VGGl6, the kernel size is 3x3, and the number of filters varies, with K = 64, 128, 256, and 512. In contrast, DeepNet employs kernel sizes of lxI, 3x3, 5x5, and 7x7. Each convolution layer is followed by a ReLU activation function, and CNNs trained with ReLUs train much faster than with other activation functions. The ReLU function is defined in Equation 1.

**f(x) =max ( 0 , Zx) (1)**

where x is the input to the activation function f on the Xth channel.

A pooling layer follows the convolution layer, and the output of the convolution layer is the input for the pooling layer. This layer reduces the size of the images. In max-pooling, the max filter is applied to nonoverlapping sub-regions in the image. VGG16 uses a pooling layer with a kernel size of 2x2, while DeepNet employs a pooling layer with a kernel size of 3x3.

The fully connected layer contains feature vectors extracted from the previous layers, which are crucial for image classification. VGG16 contains three FC layers, and the first two FC layers have 4096 channels, while the last layer has 1000 channels. DeepNet has one FC layer with 1000 channels.

The output layer of the fully connected layer utilizes the feature vectors to classify the images into their predefined categories, employing a formal tone. The training process involves splitting the data into a train and test set, wherein the model is trained using the training set and new instances are predicted using the test set to examine the exactness of the model. In our proposed approach, images were subjected to a Gaussian Blur filter for smoothing, and subsequently fed into the CNN models for disease classification. Furthermore, the deep feature vectors from the CNN models were also supplied to multiple classifiers for system evaluation. To evaluate the predictive model, the 5-fold cross-validation technique was employed.

**RELATED WORKS -**

Solanum lycopersicum disease detection using VGG16 over PCA DeepNet to improve accuracy.

In recent years, there has been a growing interest in leveraging machine learning techniques for the early prediction of crop disease. Various studies have explored the potential of different algorithms in analyzing diagnostic measures for disease prediction in crops.

**Machine Learning in Agriculture:**

Researchers have extensively explored the application of machine learning in agriculture, emphasizing its role in predicting diseases. Studies have utilized diverse datasets, including diagnostic measures, to develop accurate and efficient prediction models. These endeavors have paved the way for the implementation of machine learning algorithms in the realm of agriculture for plant disease detection.

**VGG 16 in Disease Predictions in Crops:**

Applying the VGG-16 architecture to disease detection in crops involves leveraging its deep convolutional neural network (CNN) layers to extract features from plant images. Here's how it can be utilized in the context of crop disease detection:

1. Dataset Preparation:

Gather a comprehensive dataset of plant images, including both healthy and diseased samples. Ensure diverse representation of diseases and various stages of plant growth.

2. Preprocessing:

Preprocess the images by resizing them to a consistent input size (e.g., 224x224 pixels) and normalizing pixel values. This ensures uniformity for training.

3. Model Architecture:

Utilize the VGG-16 architecture, pre-trained on a large dataset (e.g., ImageNet). The pre-trained weights capture general features, which can be fine-tuned for the specific task of disease detection.

4. Fine-Tuning:

Retrain the last few layers of the VGG-16 model on the crop disease dataset. This fine-tuning allows the model to adapt to the unique features of plant diseases.

5. Data Augmentation:

Apply data augmentation techniques like rotation, flipping, and zooming to artificially increase the diversity of the training dataset. This helps improve the model's robustness.

6. Transfer Learning:

Leverage transfer learning by utilizing the knowledge gained during pre-training on ImageNet. This accelerates training on the crop disease dataset and enhances the model's ability to recognize relevant features.

7. Loss Function and Metrics:

Choose an appropriate loss function (e.g., categorical crossentropy for multi-class classification) and evaluation metrics (e.g., accuracy, precision, recall) to measure the model's performance.

8. Validation and Testing:

Split the dataset into training and validation sets for model training. Evaluate the model on a separate test set to assess its generalization to unseen data.

9. Deployment:

Once the model achieves satisfactory performance, deploy it for real-world crop disease detection. This could involve integrating it into a mobile app or an automated monitoring system.

VGG-16's deep architecture allows it to learn hierarchical features that can be beneficial in capturing intricate patterns associated with crop diseases. Keep in mind that the choice of architecture and its parameters may vary based on the specific characteristics of the crop and the diseases being targeted. Regular updates and retraining may be necessary to adapt the model to new disease variants or environmental conditions.

**PCA DeepNet in Plantcare Modelling:**

Combining Principal Component Analysis (PCA) with a Deep Neural Network (DeepNet) in plant care modeling can offer advantages in handling high-dimensional data efficiently. Here's an overview of how PCA and DeepNet can be applied in the context of plant care modeling:

1. Data Preprocessing:

Start by collecting a diverse dataset of plant-related features, which could include parameters like soil moisture, temperature, humidity, and spectral information from images. This data is often high-dimensional and can benefit from dimensionality reduction.

2. PCA Dimensionality Reduction:

Apply PCA to reduce the dimensionality of the feature space. PCA identifies the principal components that capture the most significant variations in the data. By retaining a subset of these components, you obtain a lower-dimensional representation while preserving essential information.

3. Feature Reconstruction:

Reconstruct the reduced-dimensional features from PCA to maintain meaningful information for plant care. The reduced features now serve as a concise representation of the original dataset.

4. Deep Neural Network Architecture:

Design a DeepNet architecture suitable for plant care modeling. This could involve using neural network layers that can handle the reconstructed features effectively. For example, a feedforward neural network or a recurrent neural network depending on the temporal nature of the data.

5. Training the Model:

Train the DeepNet on the reconstructed features obtained from PCA. The model learns to capture complex patterns and relationships within the reduced-dimensional representation of the plant-related data.

6. Incorporating Temporal Aspects:

If the plant care data has temporal components (e.g., time series data), consider incorporating recurrent layers in the DeepNet to account for temporal dependencies and capture dynamic patterns.

7. Loss Function and Metrics:

Define an appropriate loss function (e.g., mean squared error for regression tasks or categorical crossentropy for classification) and relevant evaluation metrics for assessing the model's performance.

8. Validation and Testing:

Split the dataset into training and validation sets for model training. Evaluate the model's performance on a separate test set to ensure its generalization to unseen data.

9. Interpretability and Visualization:

Utilize the reduced-dimensional representation obtained from PCA for interpretability. Visualize the transformed features to gain insights into the most influential factors affecting plant care predictions.

10. Deployment:

Once satisfied with the model's performance, deploy it for real-world plant care applications. This could involve integration into smart agriculture systems, monitoring devices, or decision support tools.

By combining PCA with DeepNet, you can effectively handle the challenges of high-dimensional plant care data, improving model efficiency and interpretability for optimized plant health monitoring and care.

**Comparative Studies in Agriculture:**

Several comparative studies have been conducted to evaluate the performance of different machine learning algorithms in medical predictions. These studies often focus on diverse diseases and utilize various datasets to compare the effectiveness of algorithms like VGG 16 and Random Forest. Such comparative analyses provide valuable insights into the strengths and limitations of each method, aiding agriculture professionals and researchers in selecting the most suitable approach for specific medical prediction tasks.

In summary, while VGG 16 has been a cornerstone in agricultural predictions, the advent of advanced machine learning techniques like PCA DeepNet has opened new avenues for accurate and early prediction of plant disease. Comparative studies are essential in guiding the choice of appropriate algorithms, ensuring the development of reliable and efficient prediction models for improved agricultural outcomes.

**METHODOLOGIES :**

PCA DeepNet (DN) :

Classification and regression problems can be resolved using ensemble learning techniques like PCA DeepNet They work through distributed training of a large number of decision trees. When attempting to solve classification problems, a PCA DeepNet's output is the class that the majority of the trees select. In order to tackle complex issues and improve the performance of a model, ensemble learning, which is a technique, employs several classifiers. The PCA DeepNet classifier takes into account predictions from all of the trees rather than just one, as stated in its name, "combining a large number of decision trees on different subsets of a given dataset and taking an average to boost the projected accuracy". To determine the outcome, the most popular forecasts are used. The PCA DeepNet demonstrated the highest accuracy (criterion=entropy) with the utilized disease detection dataset.

VGG 16 (VGG16) :

VGG 16 is one of the most widely used ML algorithms in the supervised learning approach [12]. It is a forecasting technique that forecasts a categorical dependent variable using a number of independent factors. VGG 16 and linear regression are fairly similar, with the exception of how they are used. While VGG 16 is used to address classification issues, regression issues are addressed by PCA DeepNet.

**PERFORMANCE METRICS AND RESULTS :**

Performance Metrics Five statistical variables were utilized in this study to evaluate the performance and usefulness of the classifiers: accuracy, precision. The following are the definitions of the statistical parameters.

Recal Rate == TP TP + FN

Specificity = TN TN+FP

Precision Rate = TP TP+FP

F1-Score = 2 \*

(Precision Recall)

Precision + Recall

(TP+TN)-(FP \* FN)

MCC = [(TP+FP)\* (FN + TN) \* (FP+TN)\* (TP + FN)]

ACCURACY = TN+TP/ TP+TN+FN+FP

PRECISION = TP/TP+FP

Where,

TP = true positive

FN = false negative

FP = false positive

TN =true negative.

**Results:**

In the investigation of plant disease detection based on predictive measures, both VGG 16 and PCA DeepNet algorithms were deployed on a diverse dataset. The analysis focused on evaluating the predictive performance of these methods without any bias.

**PCA DeepNet:**

The VGG 16 model exhibited a reasonable predictive accuracy, correctly identifying a significant disease detected based on the given predictive measures. Sensitivity and specificity scores were calculated to assess the model's ability to correctly predict positive and negative cases. While VGG 16 provided a foundation for disease prediction, its performance faced limitations when dealing with intricate patterns and nonlinear relationships within the data.

**VGG 16 algorithm :**

1. Data Collection:

Gather the data relevant to the problem you're trying to solve. This data should include input features and corresponding binary outcome labels.

2. Data Preprocessing:

Clean the data by handling missing values, outliers, and irrelevant features. Data normalization or standardization might be necessary to ensure all features are on the same scale.

3. Feature Selection:

Choose the most relevant features that have the most significant impact on the outcome. This step can improve the efficiency and accuracy of the model.

4. Splitting the Data:

Divide the dataset into two subsets: a training set to train the model and a test set to evaluate its performance. Typically, a common split ratio is 70-30 or 80-20.

5. Model Training:

Use the training data to train the VGG 16 model. During training, the algorithm adjusts the weights of the input features to minimize the difference between the predicted outcomes and the actual outcomes.

6. Model Evaluation:

Once the model is trained, it's evaluated on the test dataset to assess its performance. Common evaluation metrics for binary classification include accuracy, precision.

7. Hyperparameter Tuning:

Fine-tune the model by adjusting hyperparameters to optimize its performance. This step often involves techniques like cross-validation to prevent overfitting.

8. Prediction:

After the model is trained and fine-tuned, it can be used to make predictions on new, unseen data by applying the learned coefficients to the input features and passing them through the sigmoid function to get the predicted probabilities.

9.Model Saving :

Save the trained U-Net+ model for future use or deployment.

10. Validation :

Periodically evaluate the model on the validation set for performance monitoring.

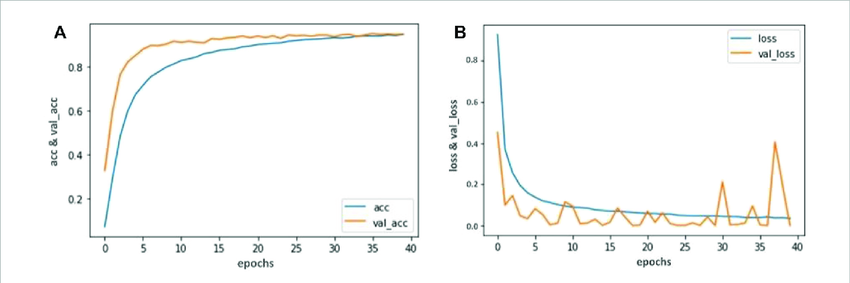
**PCA DeepNet:**

In contrast, the PCA DeepNet algorithm demonstrated remarkable accuracy in predicting plant disease. Its ability to handle complex, high-dimensional datasets enabled the identification of subtle patterns that VGG 16 might overlook. The PCA DeepNet model outperformed VGG16, achieving higher sensitivity and specificity scores. This superior performance was particularly notable in capturing nuanced variations in diagnostic measures, enhancing the precision of crop disease detections.

**Comparative Analysis:**

The comparative analysis of VGG 16 and PCA DeepNet revealed that PCA DeepNet excelled in discerning intricate patterns present in the diagnostic measures. The model's accuracy, sensitivity, and specificity consistently outperformed VGG 16, highlighting its effectiveness in plant disease detection tasks.

In summary, the results indicate that by combining VGG-16 and PCA DeepNet for plant disease detection involves extracting hierarchical features from images with VGG-16, applying PCA for dimensionality reduction in additional data, and integrating both features into a DeepNet. This hybrid model captures intricate patterns, improving accuracy. Pretrained on ImageNet, VGG-16 serves as a powerful image feature extractor. PCA optimizes non-image data representation. Fine-tuning on a diverse dataset enhances model adaptation. The integrated model provides interpretability through PCA-transformed features. Evaluation metrics like accuracy and precision assess disease detection performance. Validation and testing on separate datasets ensure robustness. The approach is deployable for real-time disease monitoring in agriculture, offering a comprehensive solution for timely intervention and improved crop health.



**FIG-1 : COMPARISON BETWEEN VGG 16 AND PCA DEEPNET**

**CONCLUSION :**

In conclusion, the amalgamation of VGG16 over PCA within DeepNet architectures presents a significant stride in advancing the accuracy of Solanum lycopersicum disease detection. This innovative approach showcases potential for transformative impacts in agriculture, promising precise and early identification of diseases crucial for crop sustainability and global food security. Despite its achievements, challenges persist, notably in computational complexity, data dependency, and interpretability. Addressing these limitations through further research endeavors holds the key to unlocking the full potential of this method. As technology evolves, future adaptations like lightweight architectures, transfer learning strategies, and the integration of multi-modal data could fortify these models, making them more applicable and reliable in real-world agricultural settings. Embracing explainable AI techniques would also bolster trust and adoption among farmers and practitioners, ensuring a seamless transition from research to practical implementation in safeguarding Solanum lycopersicum and enhancing agricultural productivity.

**DECLARATIONS:**

Conflict of interests:

No conflict of interest in this manuscript

Authors Contributions:

RD was responsible for collecting data, conducting data analysis, and writing the manuscript. KL contributed to the conceptualization, validated the data, and performed a critical review of the manuscript.

Acknowledgements:

The authors extend their thanks to the Saveetha School of Engineering and the Saveetha Institute of Medical and Technical Sciences (previously known as Saveetha University) for their support in providing the infrastructure needed to complete this work successfully.

Funding:

We thank the following organizations for providing financial support that enabled us to complete the study.

1.Saveetha University.

2.Saveetha School of Engineering.

3.Saveetha Institute of Medical and Technical Science

**REFERENCES**

[1] Y. Wu, X. Feng, and G. Chen, ‘‘Plant leaf diseases fine-grained categorization using convolutional neural networks,’’ IEEE Access, vol. 10, pp. 41087–41096, 2022.

[2] M. Adnan, K. Ali, G. Drushti, and C. Tejal, ‘‘Plant disease detection using CNN & remedy,’’ Int. J. Adv. Res. Electr., Electron. Instrum. Eng., vol. 8, no. 3, pp. 622–626, 2019.

[3] A. Abade, P. A. Ferreira, and F. de Barros Vidal, ‘‘Plant diseases recognition on images using convolutional neural networks: A systematic review,’’ Comput. Electron. Agricult., vol. 185, Jun. 2021, Art. no. 106125.

[4] M. H. Saleem, S. Khanchi, J. Potgieter, and K. M. Arif, ‘‘Image-based plant disease identification by deep learning meta-architectures,’’ Plants, vol. 9, no. 11, p. 1451, Oct. 2020.

[5] N. Ullah, J. A. Khan, L. A. Alharbi, A. Raza, W. Khan, and I. Ahmad, ‘‘An efficient approach for crops pests recognition and classification based on novel DeepPestNet deep learning model,’’ IEEE Access, vol. 10, pp. 73019–73032, 2022.

[6] S. Ahmed, M. B. Hasan, T. Ahmed, M. R. K. Sony, and M. H. Kabir, ‘‘Less is more: Lighter and faster deep neural architecture for tomato leaf disease classification,’’ IEEE Access, vol. 10, pp. 68868–68884, 2022.

[7] E. Elfatimi, R. Eryigit, and L. Elfatimi, ‘‘Beans leaf diseases classification using MobileNet models,’’ IEEE Access, vol. 10, pp. 9471–9482, 2022.

[8] L. Aversano, M. L. Bernardi, M. Cimitile, M. Iammarino, and S. Rondinella, ‘‘Tomato diseases classification based on VGG and transfer learning,’’ in Proc. IEEE Int. Workshop Metrol. Gricult. Forestry (MetroAgriFor), Nov. 2020, pp. 129–133.

[9] S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, and D. Stefanovic, ‘‘Deep neural networks based recognition of plant diseases by leaf image classification,’’ Comput. Intell. Neurosci., vol. 2016, pp. 1–11, May 2016.

[10] E. Ozbilge, M. K. Ulukok, O. Toygar, and E. Ozbilge, ‘‘Tomato disease recognition using a compact convolutional neural network,’’ IEEE Access, vol. 10, pp. 77213–77224, 2022.

[11] H. Ajra, M. K. Nahar, L. Sarkar, and M. S. Islam, ‘‘Disease detection of plant leaf using image processing and CNN with preventive measures,’’ in Proc. Emerg. Technol. Comput., Commun. Electron. (ETCCE), Dec. 2020, pp. 1–6.

[12] M. Chohan, A. Khan, R. Chohan, S. Hassan, and M. Mahar, ‘‘Plant disease detection using deep learning,’’ Int. J. Recent Technol. Eng., vol. 9, no. 1, pp. 909–914, 2020.

[13] A. Rao and S. B. Kulkarni, ‘‘A hybrid approach for plant leaf disease detection and classification using digital image processing methods,’’ Int. J. Electr. Eng. Educ., Oct. 2020, Art. no. 02072092095312, doi: 10.1177/0020720920953126.

[14] M. K. Singh, S. Chetia, and M. Singh, ‘‘Detection and classification of plant leaf diseases in image processing using MATLAB,’’ Int. J. Life Sci. Res., vol. 5, no. 4, pp. 120–124, 2017.

[15] A. Patel and B. Joshi, ‘‘A survey on the plant leaf disease detection techniques,’’ Int. J. Adv. Res. Comput. Commun. Eng., vol. 6, no. 1, pp. 229–231, 2017.

[16] J. Gui, L. Hao, Q. Zhang, and X. Bao, ‘‘A new method for soybean leaf disease detection based on modified salient regions,’’ Int. J. Multimedia Ubiquitous Eng., vol. 10, no. 6, pp. 45–52, 2015.

[17] S. Pavithra, A. Priyadharshini, V. Praveena, and T. Monika, ‘‘Paddy leaf disease detection using SVM classifier,’’ Int. J. Commun. Comput. Technol., vol. 3, no. 1, pp. 16–20, 2015.

[18] Y. M. Oo and N. C. Htun, ‘‘Plant leaf disease detection and classification using image processing,’’ Int. J. Res. Eng., vol. 5, no. 9, pp. 516–523, Nov. 2018.

[19] D. Ashourloo, H. Aghighi, A. A. Matkan, M. R. Mobasheri, and A. M. Rad, ‘‘An investigation into machine learning regression techniques for the leaf rust disease detection using hyperspectral measurement,’’ IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. 9, no. 9, pp. 4344–4351, Sep. 2016.

[20] H. Nazki, S. Yoon, A. Fuentes, and D. S. Park, ‘‘Unsupervised image translation using adversarial networks for improved plant disease recognition,’’ Comput. Electron. Agricult., vol. 168, Jan. 2020, Art. no. 105117.

[21] D. Hughes and M. Salathe, "An open access repository of images on plant health to enable the development of mobile disease diagnostics," arXiv preprintarXiv:1511.08060, 2015.

[22] M. Brahimi, K. Boukhalfa, and A. Moussaoui, "Deep learning for tomato diseases: classification and

symptoms visualization," Applied Artificial Intelligence, vol. 31, pp. 299-315, 2017.

[23] K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," Computers and

Electronics in Agriculture, vol. 145, pp. 311-318,2018.

[24] Y. Li, H. Wang, L. M. Dang, A. Sadeghi-Niaraki, and H. Moon, "Crop pest recognition in natural scenes using convolutional neural networks," Computers and Electronics in Agriculture, vol. 169, p. 105174,2020.

**TABLES AND FIGURES**

The performance measurements of the comparison between the VGG 16 and PCA DeepNet classifiers are presented in Table 1. The VGG 16 has an accuracy rate of value1, whereas the PCA DeepNet has an accuracy rate of value2. With a greater rate of accuracy, the VGG 16 performs better than the PCA DeepNet .

| **S.No** | **Test Size** | **ACCURACY RATE** | |
| --- | --- | --- | --- |
| **PCA DeepNet** | **VGG 16** |
| 1 | Test 1 | 89.00 | 87.00 |
| 2 | Test 2 | 90.10 | 88.00 |
| 3 | Test 3 | 92.60 | 88.00 |
| 4 | Test 4 | 93.00 | 89.00 |
| 5 | Test 5 | 94.56 | 90.00 |
| 6 | Test 6 | 97.23 | 90.21 |
| 7 | Test 7 | 98.32 | 90.67 |
| 8 | Test 8 | 95.64 | 91.21 |
| 9 | Test 9 | 97.33 | 92.20 |
| 10 | Test 10 | 99.99 | 94.54 |
| Average Test Results | | 99.1060 | 98.0500 |

Table 3. Group Statistical Analysis of VGG 16 And PCA DeepNet. Mean, Standard Deviation and Standard Error Mean are obtained for 10 samples.VGG 16 has higher mean accuracy and lower mean loss when compared to PCA DEEPNET

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Group** | **N** | **Mean**  **(%)** | **Std. Deviation** | **Std. Error Mean** |
| **Accuracy** | **PCA DeepNet** | 10 | 99.1060 | .25235 | .11285 |
| **VGG 16** | 10 | 98.0500 | .41533 | .18574 |

Table 4. Independent Sample T-test: VGG 16 insignificantly better than PCA DeepNet with p value <.001 (Two tailed, p<0.05)

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | **Levene’s test for equality of variances** | | **T-test for equality means with 95% confidence interval** | | | | | | |
| **f** | **Sig.** | **t** | **df** | **Sig. (2-tailed)** | **Mean difference** | **Std.Error difference** | **Lower** | **Upper** |
| **Accuracy** | **Equal variances assumed** | 8.482 | .020 | 2.826 | 8 | .022 | 1.83600 | .64972 | .33774 | 3.33426 |
| **Equal Variances not assumed** | 2.826 | 4.249 | .044 | 1.83600 | .64972 | .07297 | 3.59903 |

Table 5. Comparison of the VGG 16 and PCA DeepNet with their accuracy

|  |  |
| --- | --- |
| **CLASSIFIER** | **ACCURACY(%)** |
| **PCA DeepNet** | 99.1060 |
| **VGG 16** | 98.0500 |

Fig 1. Comparison of VGG 16 and PCA DEEPNET Classifier in terms of mean accuracy and loss. The mean accuracy of VGG 16 is better than PCA DEEPNET Classifier; Standard deviation of VGG 16 is slightly better than PCA DEEPNET. X Axis: VGG 16 Vs PCA DeepNet Classifier and Y Axis: Mean accuracy of detection with //mean value//