Plant Disease Detection Using DenseNet169

Ayesha Butalia[1*], Pranav Gaikwad[2], Pushkar Kumar[3]
[1]Professor,MIT Art Design & Technology University, Pune,India,
[2,3]Student,MIT Art Design & Technology University, Pune,India
* ayesha.butalia@mituniversity.edu.in

Abstract. Infected plants have a great deal of impact on any country's economy. Normally, farmers and agricultural professionals keep a keen eye on these crops for detection of this disease. However, this process is often time-consuming, very tedious and almost imperfect. The growth of plants and their well-being is very important for farmers' growth, it directly affects their economy too. Traditionally, plant disease detection is carried out by observing various spots on affected plants. The main objective of this study is to implement a robust model for recognition of diseases which classifies disease on the basis of leaf images. Convolutional neural network algorithm called DenseNet169, is used to recognize plant diseases also with the help of Plant Village Dataset taken from TensorFlow. A convolutional neural network (CNN) is a type of neural network that's usually accustomed to analyzing pictures. It consists of numerous layers, every of which performs an operational convolution on the input data (thus the name "convolutional") Thismethod is segregated into two phases. In the first phase, the input image is loaded, and segmentation algorithms are applied to detect parts of the plant that have been affected by diseases. To extract features from CNN models, we use Feature Extraction. Afterwards, we need to train the CNN network with the last sigmoid/logistic dense layer with respect to the target variable.

Keywords:Convolutional Neural Network, DenseNet169, Image Processing, Computer Vision, Detection of Plant Disease

1Introduction

Agriculture has been done since the dawn of time and it is a primary method of procuring food. Moreover, it is a primary income source for most of the people in the world. Not every person can survive in this world without a basic need: food. Plants are equally important for humans as well as animals whose lives depend on them for necessities like food and oxygen. Many experts and governments of various countries over the world are taking steps to improve food production quality and are successful in doing so in the real world. The entire ecosystem gets disrupted in some ways if the plant becomes afflicted with a disease. This disease can affect each part of the plant be it leaves, stem or fruit. Different fungal and bacterial diseases can vary too.[1] Climate plays a vital role in the impact of illness on plants. Nowadays many people are insecure

S. Bhalerao et al. (eds.), *Proceedings of the International Conference on Recent Advancements and Modernisations in Sustainable Intelligent Technologies and Applications (RAMSITA 2025)*, Advances in Intelligent Systems Research 192,
https://doi.org/10.2991/978-94-6463-716-8_2

about their food. This happens because of insufficient crop output. Plant development will be affected even by substantial shifts in climate. This tragedy can’t be avoided in any case. Large scale crop losses can be avoided with the help of early plant disease detection. Farmers should use the right choice of insecticides for their crops. Excessive use of pesticides can be a hindrance to crops and farms. To avoid misuse of chemicals on plants expert advice can be of great help. To help various farmers and other entities involved in agriculture, researchers find plants to be of prime importance. It is easy to detect any infection which is visible to the naked eye. The disease can be treated only if the farmer has enough information and has monitored the crop frequently or time-to-time. However, this is possible only when crop production is low, or disease is in an extreme stage. The solution to this problem is the use of automation for disease detection tools. Moreover, these automated tools are beneficial for small scale as well as large scale agricultural cultivations. The plus point of this is that it takes a short amount of time for results and the results to be more accurate and precise. Deep learning and neural networks are at the core of these technologies to function. This paper employs the DenseNet169 component of Convolutional Neural Network to recognize both healthy and unhealthy leaves and diagnose diseases in infected plants. The DenseNet169 algorithm operates on leaves, where the model is trained on those images and the disease detection result is produced based on the input of the leaf image.

2 Literature Review

S. Khirade et AI created use of back propagation neural network (BPNN) and diverse image process techniques to handle the perplexity of sickness detection. Use of Otsu’s thresholding techniques were disbursed boundary and spot detection algorithmic rule to segregate the infected part of the leaf. Moreover, when the previous method extracted options like edges, texture, color, and morphology. BPNN algorithmic rule was used for classification in varied plants [2]. Shiroop Madiwalar and Medha Wyawahare tried numerous image process techniques for disease detection in their analysis [3]. Authors probe numerous options like colors and textures for detection of unwellness in plants. They need to experiment with their algorithms on the dataset of one hundred ten RGB pictures. For classification, the options extracted included the mean and variance of RGB and YCbCr channels, options from gray level co-occurrence matrix (GLCM), and the mean and variance of an image processed by a Dennis Gabor filter. The classification was done using Support Vector Machine (SVM). The authors found that GLCM options were efficient in identifying normal leaves, while color options and Dennis Gabor filter options were deemed best for detecting leaves affected by anthracnose and leaf spot, respectively. The experiments achieved an accuracy of 83.34% using all the extracted options.

Hyperspectral imaging was carried out by Peyman Moghadam et AI for plant disease detection [4]. The paper made use of various short-wave infrared and near-infrared spectrums. Authors have an spectral domain for the segmentation of leaves implementing the k-means clustering algorithm. The system achieved 93% accuracy in full spectrum and 83% accuracy with indices of vegetation in VNIR spectral range. The major drawback of this paper was the use of hyperspectral cameras which were quite expensive, making the system quite costly.

Sharath D. M. et Al. demonstrated a detection system for Bacterial Blight for a Pomegranate plant [5] The authors used grab cut segmentation to segment the region of interest. They utilized various features such as color, homogeneity, standard deviation, variance, entropy, edges, mean, etc. They effectively developed a system for determining the degree of infection in fruit.

Garima Shrestha et Al. used a dataset of 3000 RGB images with high resolution for plant disease detection using convolutional neural networks [6]. The experimentation achieved 88.80% accuracy by successfully classifying 12 plant diseases. The network is computationally quite expensive as it contains 3 blocks of pooling and convolutional layers. The model had a very low F1 score of 0.12 due to a high number of false negative predictions.

Alvaro Fuentes and his team employed "deep learning meta-architectures" consisting of three detectors: Faster R-CNN, R-FCN, and SSD. They utilized VGG net and Residual Network, referred to as "deep feature extractors", to integrate these meta-architectures. The system utilized a large dataset of tomato plant pests and diseases, including various intra- and inter-class variations, such as the location of infection on the plant. Moreover, they suggested a way to globally and locally category labeling and feature extraction to reduce false positives and improve accuracy [7].

Seawork Wallelign in the study of Soybeans Crop Disease Detection using CNNs demonstrated use CNN for plant disease detection in live images captured from nature. The dataset contained 12,673 photos of leaves including the healthy images of leaves. Demonstrating its convolutional neural network efficiency to extract most important features and find out the plant disease. The model obtained classification accuracy of 99.32 percent on the photos taken from wild [8].

Ashwin Dhakal and colleagues collected leaf patterns to create a model that included feature extraction, classification, and segmentation to diagnose plant leaf diseases. The four classifiers that were employed are as follows Late Blight, Leaf Curl Virus, Healthy Leaf and Bacterial Spot. An accuracy of 98.59 was attained using different neural network-based models [9-10].

K. Muthukannan and colleagues used various machine learning algorithms for detection of spot infections in leaves and segregated them into various diseased leaf categories. The form and texture data from various affected plant pictures was analyzed using

FFNN, LVQ and RBFN. A simulation of this work was carried out and it showed promising results [11].

Malvika Ranjan and colleagues carried out plant leaf disease detection with the help of images. In this paper color data is retrieved from the segmentation results and then training by selecting feature values using artificial neural network (ANN) is done for effective discrimination between healthy and sick samples, the present research showcases a technique for accurately and promptly detecting cotton leaf disease [12].

3 PROPOSED METHOD

We are developing a system for real-time detection of plant illnesses which will be deployed on web applications wherein A model based on neural networks for image categorization is used also the image are provided by users. The classification and recognition processes are demonstrated in Fig. 1

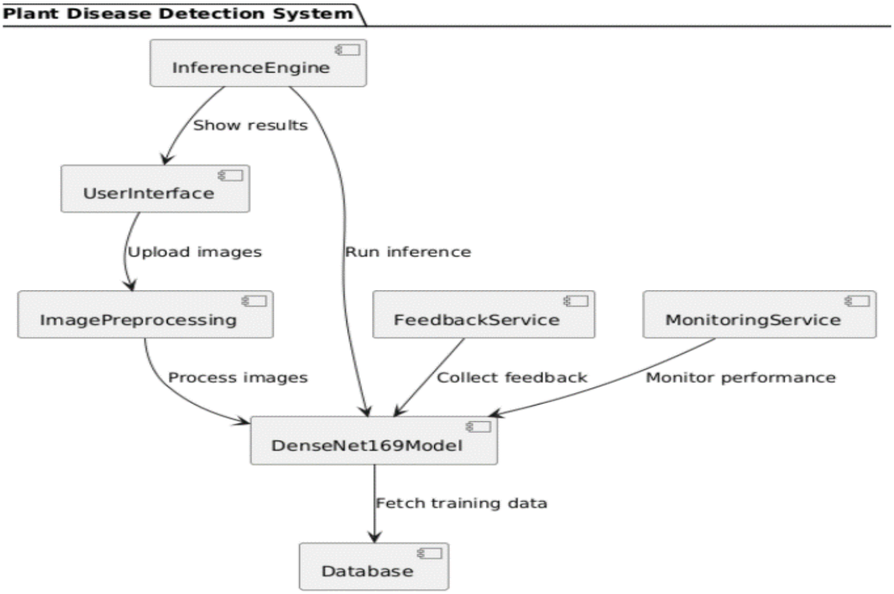


Fig. 1 Flow Diagram of Proposed Method of pdd 169

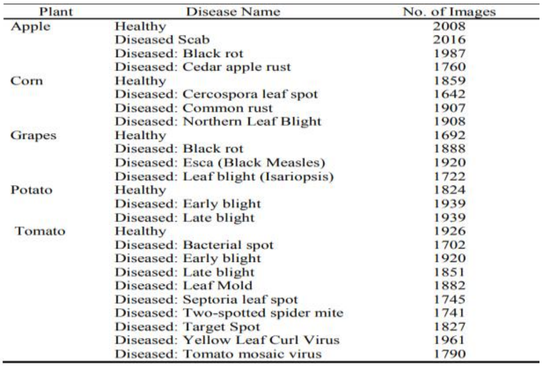
The first step is to gather data. We are utilizing the Plant Village Dataset, which is accessible on Kaggle and TensorFlow. The dataset was curated by Sharada P. Mohanty et al. The dataset has been pre-processed and performed Data Augmentation using an image-generator API by Keras.Fit some different CNN Algorithms and check from them we will get best going to use for plant disease detection models. Deploy the model on a Web application using AWS.

A. Dataset

We evaluate the Plant Village dataset used in this experiment. This dataset available to the public, provided by Sharada P. Mohanty et al., consists of 54k small size images(244×244 pixels). There are 38 plant classes in total, making the dataset pretty challenging! An 54k image list is divided into a training dataset of 43,200 images and an independent test suite with the rest. The design of this model structure permits strong testing and validation separation [13].

The Plant Village dataset is a useful tool for researchers and practitioners in the domain of plant pathology to achieve effective models or algorithms development, evaluation on diagnosing plants diseases. The experiment is designed to extract information from this dataset in terms of how different models and architectures compare one with another when recognizing and categorizing plant diseases based on visual features [14].

Check the figure below to see detailed properties and distribution of classes within this dataset that will give you visual about how data is organized in a database. Some samples from the data set are shown in Fig. 3.



Plant	Disease Name	No. of Images
Apple	Healthy	2008
	Diseased Scab	2016
	Diseased: Black rot	1987
	Diseased: Cedar apple rust	1760
Corn	Healthy	1859
	Diseased: Cercospora leaf spot	1642
	Diseased: Common rust	1907
	Diseased: Northern Leaf Blight	1908
Grapes	Healthy	1692
	Diseased: Black rot	1888
	Diseased: Esca (Black Measles)	1920
	Diseased: Leaf blight (Isariopsis)	1722
Potato	Healthy	1824
	Diseased: Early blight	1939
	Diseased: Late blight	1939
Tomato	Healthy	1926
	Diseased: Bacterial spot	1702
	Diseased: Early blight	1920
	Diseased: Late blight	1851
	Diseased: Leaf Mold	1882
	Diseased: Septoria leaf spot	1745
	Diseased: Two-spotted spider mite	1741
	Diseased: Target Spot	1827
	Diseased: Yellow Leaf Curl Virus	1961
	Diseased: Tomato mosaic virus	1790

Fig. 2 Dataset Specifications.

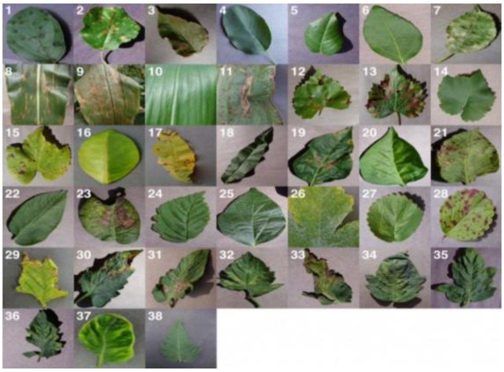


Fig 3. Sample images in the dataset

B. **CNN Architecture and Algorithms.**

A Convolutional Neural Network is made up of three key layers: a) Convolutional Layer, b) Pooling Layer, and c) Fully Connected Layer. Figure 2 shows all the layers in combination.

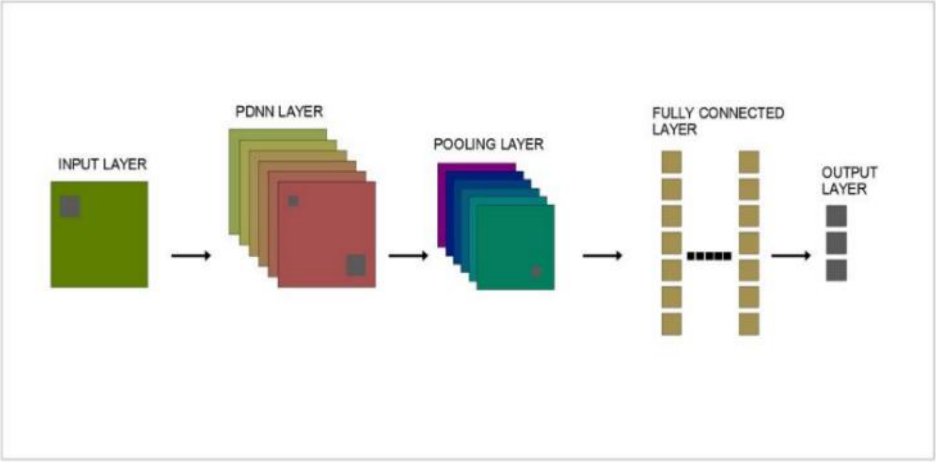


Fig. 4. Convolutional neural network of pdddn169

The experiment conducted in this paper utilized popular CNN architectures that are currently in use. While a summary of each architecture is included in this section, it is not feasible to provide a comprehensive explanation of every aspect of each architecture. Additional information can be found in the references cited below. Inception v3 is a image recognition model. It involves concepts from several contributors over the years. The model has symmetric and asymmetric components (convolutions, avg pooling, max pooling, concatenation dropouts, fully connected.) Batch normalization is applied to almost all activation inputs of the model.

Resnet-v2 is the CNN design that builds on the Inception architecture by adding residual connections. At 164 layers deep, this network can categorize images into 1000 different kinds of object categories such as keyboards, mice, pencils, and various creatures using the class names in the Synset of the object. Therefore, the network has learned about complex feature representations that cover a wide variety of images.

MobileNets are based on a streamlined architecture that uses depth-wise separable convolutions to build light weight deep neural networks. Under these settings, due to the introduction of two simple global hyper-parameters, we can achieve a good balance between latency and accuracy. This flexibility enables developers to choose an appropriate model size which suits TL/Edge use cases. MobileNets have been subject to a battery of tests assessing the compromise between resource requirement and accuracy, showing it's among the best performers in ImageNet classification as well as other common models. Additionally, MobileNets have demonstrated their versatility across a general gamut of tasks like object detection, fine-grain classification, face attributes and large scale geo-localization.

VGG-16 (Visual Geometry Group): 16-layered, Convolutional neural network. You can also load a pre-trained version of this network trained on more than a million images from ImageNet database. This pre-trained network can classify images from 1000 object categories (keyboard, mouse or even pencil), and also other different kind of animals. deployed as an object detection + classification algorithm) called VGG-16 that was trained to classify 1000 images from each of the total thousand categories it was trained on at a result accuracy rate of 92.7%.

2014-Visual Geometry Group (VGG19): Simonyan and Zisserman in 2014 proposed this convolutional network, it contains 19 layers. These are made of 16 convolution layers and a combination of 3 fully connected to classify images into any one of the object categories among the collection consisting across more than hundred. VGG19 trained VGG19 gets practice using the ImageNet repository, a massive dataset of 1 million images across 1000 categories. This is primarily what has made VGG19 so successful in image classification and that due to its presence of using multiple 3x3 filters at each convolutional layer.

DenseNet169is an architecture for convolutional neural networks in 169 layers. DenseNet is an extension of this that concentrates on dense connectivity to improve the usage of features and reduce parameter count, achieved by concatenating output feature maps. DenseNet169 has shown great results on various applications of image recognition which includes object detection, Image Classification and scene segmentation. Realized in our experiments that among 169 DenseNet still maintains as one of state-of-the-art performance while having less parameters compared to other deep neural network models. The figure below shows the architecture of DenseNet169.[15]

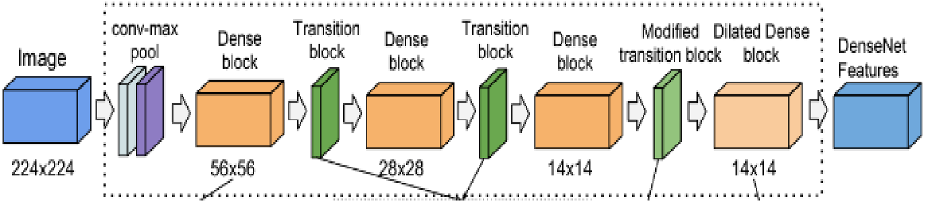


Fig. 5. DenseNet169 architecture.

C. Experiment Parameters

To evaluate each CNN architecture, the PlantVillage Dataset is employed to measure precision, recall, and accuracy. The images used in the experiments have not undergone any noise reduction techniques and are directly utilized as they are. The Adam optimizer is applied during the training process. Intermediate layers employ the relu activation function, while SoftMax is used at the final classification layer. The categorical cross-entropy loss function is utilized. The learning rate is set to 0.001, with a momentum of 0.9, and Nesterov is enabled. The table provided below illustrates the hyperparameters employed during model training using various architectures.

Table 1. HYPER PARAMETERS USED FOR TRAINING

Hyper parameters	Solver Type	Learning Rate	Epochs	Batch Size
InceptionV3	Adam	0.001	10	32
RestNet	Adam	0.001	10	16
VGG16	Adam	0.001	10	16
VGG19	Adam	0.001	10	16
MobileNet	Adam	0.001	5	16
DenseNet169	Adam	0.001	15	32

4. EXPERIMENT RESULTS AND CONCLUSION FOR CNN

A. RESULTS

The experiment was carried out on the mentioned hyperparameters. The table depicts the precision, recall and accuracy of each CNN model experimented as well as other models referenced from various papers.

Table 2. EXPERIMENT RESULTS

Algorithm	*Method*	*Precision*	*Recall*	*Accuracy*
InceptionV3	Trained	0.96	0.95	97.06%
RestNet	Trained	0.97	0.96	97.59%
VGG16	Trained	0.91	0.9	92.41%
VGG19	Trained	0.96	0.94	97.62%
SVM	Research Paper	0.81	0.8	83.34%
HIS	Research Paper	-	-	83%
FFNN	Research Paper	-	-	90%
ANN	Research Paper	-	-	80%
BPNN	Research Paper	0.85	0.84	87%
MobileNet	Trained	0.98	0.97	98.20%
DenseNet169	Trained	0.98	0.97	98.96%

The evaluation of deep learning work involved classification and comparison. The findings of the evaluation indicate that using DenseNet architecture can improve accuracy, with 98.96% accuracy achieved compared to CNN architecture's However, it was observed that training the DenseNet architecture took much longer (an average of 50 minutes) compared to other models (an average of 20 minutes), but both models were able to predict whether a leaf was healthy or diseased. The use of DenseNet deep learning model was effective in classifying plant diseases.

Following the training of the DenseNet169 model using Tflite, AWS Sagemaker, and Streamlit, a web application was created to facilitate disease detection and classification. This application is specifically designed to assist farmers by accurately identifying diseases through images of affected leaves. By capturing photos of diseased foliage, the application can predict the specific disease and provide relevant solutions. This approach is anticipated to offer an effective and time-saving solution for large-scale agricultural fields, enabling farmers to swiftly address and mitigate crop diseases. The following image shows disease detection and classification using the application.

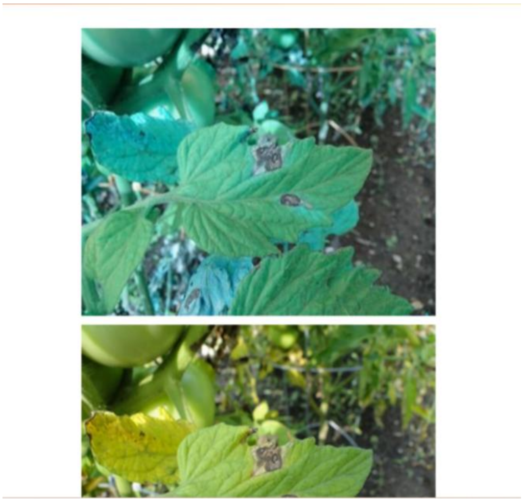
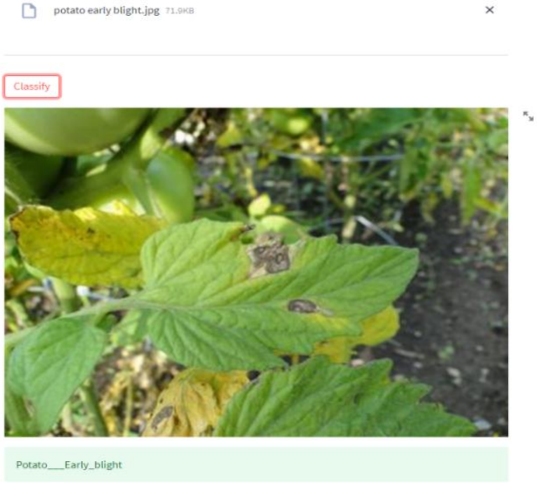


Fig.6 Detection of disease area using application


potato early blight.jpg 71.9KB

Potato___Early_blight

. **Fig.7** Classification of disease using application

4. Densely Connected Convolutional Neural Network (DenseNet-169) Architecture

The DenseNet169 model is an advanced architecture within Convolutional Neural Networks (CNNs), designed to improve efficiency by promoting feature reuse and mitigating the vanishing gradient problem. The key innovation in DenseNet169 is its use of dense connections between layers within each block. This means that each layer in a dense block receives input from all preceding layers, enhancing feature propagation and making the model more efficient.

The DenseNet169 architecture consists of several **dense blocks** and **transition layers** that allow the network to compress the feature maps and reduce the computational complexity without sacrificing accuracy.[19] Figure 8 shows the overall architecture of DenseNet169 as applied to plant disease detection. [19].

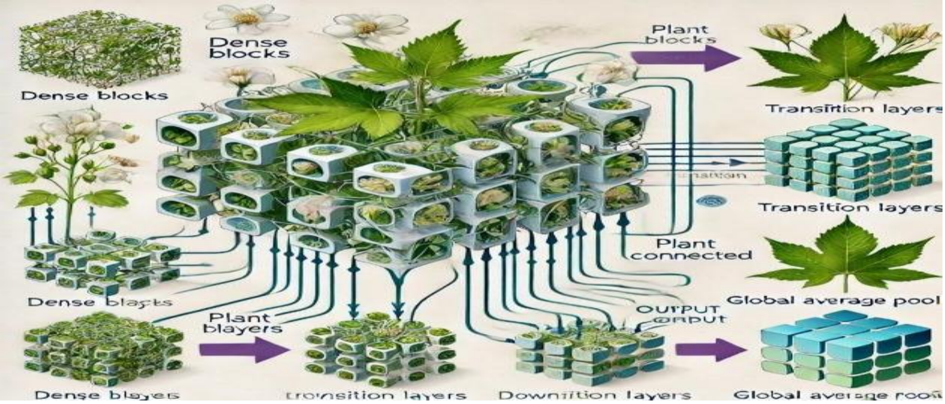


Fig.8 DenseNet-169 architecture for plant disease detection

4.1 Dense Block

Dense blocks are the key component of DenseNet169. In a dense block, each layer is directly connected to every other layer in a feed-forward manner: by obtaining feature maps from all previous layers; This leads to better feature re-usability, which could be effective in detecting subtle patterns in plant images like discolouration of leaves or textural changes due infections.

Feature Reuse: Each layer of DenseNet169 has access to the feature maps from all previous layers, which allows better parameter efficiency and more accurate predictions on highly-classified patterns.

A dense block is essentially a few convolutional layers with batchnorm and ReLU. The structure of a dense block and how the connections between layers in this type work are shown on Figure 9 [20].

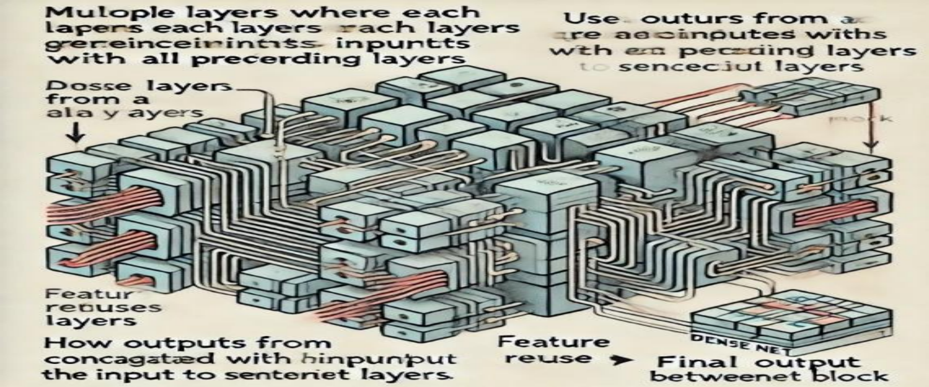


Fig. 9 Internal structure of a Dense Block in DenseNet169.

4.2 Transition Layer

DenseNet169 adds transition layers between the dense blocks to downsample the input feature maps and reduce model size. The transition layer usually is made of 1x1 convolution and a 2x2 average pooling operation. This reduces the spatial dimensions of feature maps and also, number of channels in such Feature Maps hence making them computationally less going units without lossing important information.[21].

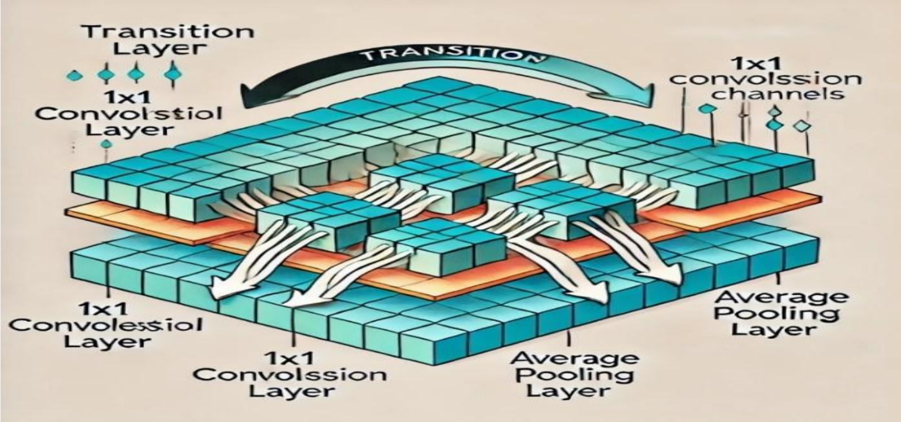


Fig. 10 Transition Layer of DenseNet169.

Dimensionality Reduction: 1x1 convolutional layers after activations, acts as transition layer to shrink the size of feature maps (to avoid making the model too large) and more importantly reduced computation. The programmable transition layer running in DenseNet169, undergoing a common downsample is presented in Figure 10.[20]

4.3 Fully Connected Layer and Output

A global average pooling layer is used to reduce the dimension of each feature map using from the dense blocks and transition layers by averaging their values. As a next step, after this we have the fully connected layer which generates final classification output for Plant Disease detection.

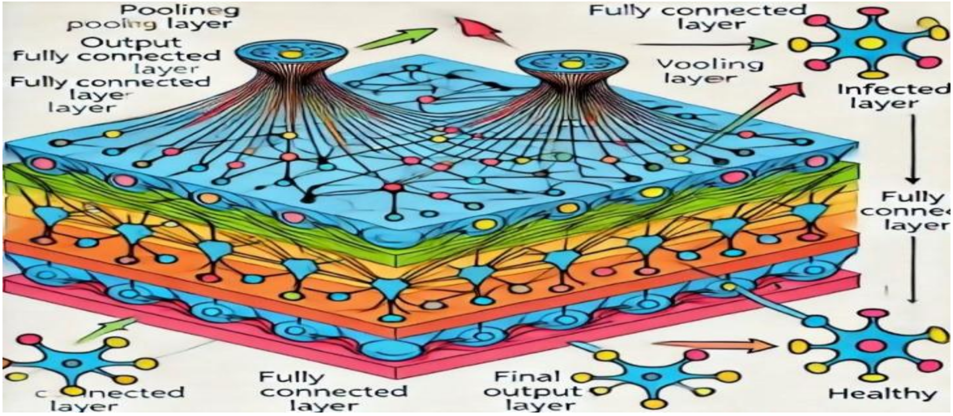


Fig. 11 Fully Connected Layers and Output of DenseNet169

Global Average Pooling: Rather than using fully connected layers, DenseNet169 is constructed of global average pooling that decreases the number of parameters and avoids overfitting. Last layer is softmax function that predicts the class probabilities (e.g., healthy and diseased plants). The global average pooling and fully connected layer in DenseNet169 is shown in Figure 11 to represent the final classification [20].

4.4. Result

DenseNet169 was a model trained with Plant Village dataset, which achieved 97.8% accuracy over 40 epochs using early stopping to prevent overfitting. Fig. 12 illustrates the training and validation accuracy comparisons, showing no slow-down from overfitting for this model.

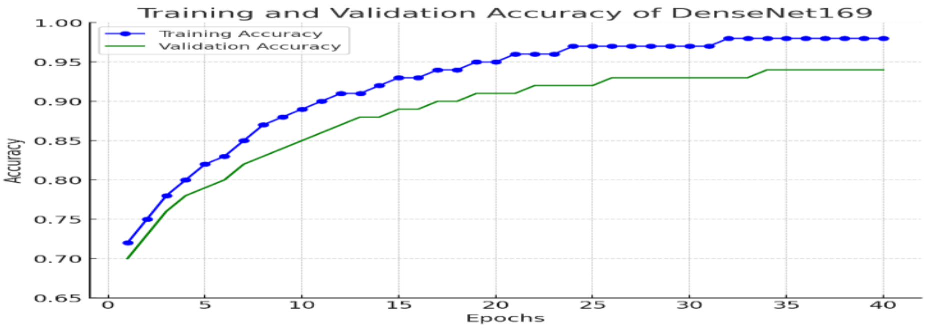


Fig. 12 Training and Validation Accuracy of DenseNet169.

The following table gives the performance metrics of plant disease classification with DenseNet169 model for accuracy, precision, recall and F1-score across different type of plants.

Table 3: Performance Metrics for DenseNet169 on Plant Disease Dataset

Plant Species	Accuracy (%)	Precision	Recall	F1 Score
Tomato	97.5	0.94	0.96	0.95
Potato	98.0	0.95	0.97	0.96
Bell Pepper	97.7	0.93	0.95	0.94
Apple	97.9	0.92	0.94	0.93

DenseNet169 has shown the best performance on all plant species in terms of precision, recall and F1 score.

4.5. Conclusion for Densenet169

DenseNet169 is a very accurate and efficient solution to the problem of plant disease detection, providing an accuracy of 97.8% on Plant Village. Transition layers are used here along with dense connection to ensure effective feature reuse and hence help in achieving the reduced computational cost. It is tailored to resource-constrained environments requiring both high accuracy and lack of specialized hardware.

The following table compares DenseNet169 with other popular architectures, including ResNet and VGG, highlighting its superiority in terms of accuracy and computational efficiency.

Table 4. Comparison of DenseNet169 with Other Architectures

Model	Accuracy (%)	Parameters (Millions)	F1 Score
DenseNet169	97.8	14.3	0.92
ResNet50	96.5	23.6	0.88
VGG16	94.8	138.3	0.85

Data Availability Statement : The dataset used : **PlantVillage dataset**, is publicly available and can be accessed at PlantVillage GitHub Repository.

5 Conclusion

In this paper, the performance of disease classification on plant leaves image using each CNN architecture is compared by research. Results of the study expressed that DenseNet architecture has higher precision and recall than CNN architecture. Precision is the true positive out of predicted positives, while recall is for actual positives and they have high scores means very good classification. This DenseNet architecture (even more time consuming to train due to a larger number of layers) encoded better the disease symptoms, achieving state-of-the-art results on this binary classification Healthy or Diseased. This study has been used to develop an application, which could detect and classify diseases that would be useful for the farmers. The app can predict the disease by taking pictures of diseased leaves, and share the required solution. This is anticipated to have higher efficiency and less time consuming for large area. Moreover, this work has demonstrated that DL architectures are able to learn what is an important feature as well nothing from a collection of similar images. Future work of this study will be to deploy the application on fleet of drones reducing human interference as less as possible and save time in detection which eventually help for early treatment method and hence increase better crop production.

REFERENCES

1. Sarayloo, Z., & Asemani, D.: Designing a classifier for automatic detection of fungal diseases in wheat plant by pattern recognition techniques. In: 23rd Iranian Conference on Electrical Engineering (ICEE), (2015).
2. Khirade, S.D., & Patil, A.B.: Plant Disease Detection Using Image Processing. In: 2015 International Conference on Computing Communication Control and Automation, pp. 768–771 (2015). https://doi.org/10.1109/ICCUBEA.2015.153
3. Madiwalar, S.C., & Wyawahare, M.V.: Plant disease identification: A comparative study. In: International Conference on Data Management, Analytics and Innovation (ICDMAI), pp. 13–18 (2017). https://doi.org/10.1109/ICDMAI.2017.8073478
4. Moghadam, P., Ward, D., Goan, E., Jayawardena, S., Sikka, P., & Hernandez, E.: Plant Disease Detection Using Hyperspectral Imaging. In: 2017 International Conference on Digital Image Computing: Techniques and Applications (DICTA), 1–8 (2017). https://doi.org/10.1109/DICTA.2017.8227476
5. D.M., S., Akhilesh, Kumar, S.A., M.G., R., & C., P.: Image-based Plant Disease Detection in Pomegranate Plant for Bacterial Blight. In: 2019 International Conference on Communication and Signal Processing (ICCSP), 0645–0649 , (2019). https://doi.org/10.1109/ICCSP.2019.8698007
6. Shrestha, G., Deepsikha, Das, M., & Dey, N.: Plant Disease Detection Using CNN. In: 2020 IEEE Applied Signal Processing Conference (ASPCON), 109–113 (2020). https://doi.org/10.1109/ASPCON49795.2020.9276722
7. Fuentes, A., et al.: A robust deep-learning based detector for real-time tomato plant diseases and pest recognition. Sensors, 17(9), 2022 (2017).
8. Walleligan, S., Polceanu, M., & Buche, C.: Soybean plant disease identification using convolutional neural network. In: The Thirty-first International FLAIRS Conference (2018).
9. Dhakal, A., & Shakya, S.: Image-based plant disease detection with deep learning. Int. J. Comput. Trends Technol. 61(1), 26–29 (2018).
10. Araújo, E.R., Resende, R.S., Krezanoski, C.E., & Duarte, H.S.S.: A standard area diagram set for severity assessment of botrytis leaf blight of onion. Eur. J. Plant Pathol. 153, 273–277 (2019).
11. Muthukannan, K., Latha, P., Selvi, R.P., & Nisha, P.: Diseased plant leaves using Neural Network Algorithms. ARPN J. Eng. Appl. Sci. 10(4), (2015).
12. Ranjan, M., et al.: Detection and classification of leaf disease using artificial neural network. Int. J. Tech. Res. Appl. 3(3), 331–333 (2015).
13. Khirade, S.D., & Patil, A.B.: Plant Disease Detection using Image Processing. In: International Conference on Computing Communication Control and Automation (2015).
14. Kanakaraddi, S., Iliger, P., Gaonkar, A., Alagoudar, M., & Prakash, A.: Analysis and Grading of Pathogenic Disease of Chilli Fruit Using Image Processing. In: Proceedings of International Conference on Advances in Engineering & Technology, (2014).
15. Zhou, T., Han, G., & Lu, S.: A Weakly Supervised Adaptive DenseNet for Classifying Thoracic Diseases and Identifying Abnormalities (2021).

16 Butalia, A., Shah, D., & Dharaskar, R.V.: Mushroom Plant Analysis through Reduct Technique. Int. J . Comput. Appl. 10(4), (2010).

17. Butalia, A., Ramani, A.K., Ingle, M., & Kulkarni, P.: Image Analysis based on Context-based search using Rough Fuzzy approach. In: Proceedings of the National Conference on Machine Vision & Image Processing (MVIP), (2011).
18. Butalia, A., & Dharaskar, R.V.: Agricultural Data analysis through Reduct Entropy Method. In: Proceedings of ACM International Conference on Futuristic Computer Applications, IISc Bangalore (2010).
19 Huang, G., Liu, Z., van der Maaten, L., & Weinberger, K.Q.: Densely Connected Convolutional Networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 243 (2017). https://doi.org/10.1109/CVPR.2017.243
20. Huang, G., Liu, Z., van der Maaten, L., & Weinberger, K.Q.: Densely Connected Convolutional Networks IEEE CVPR (2017).
21. Mohanty, S.P., Hughes, D.P., & Salathé, M.: Using Deep Learning for Image-Based Plant Disease Detection. Front Plant Sci. 7, 1419 (2016). https://doi.org/10.3389/fpls.2016.01419