ISSN: 2584-0495 Vol. 1, Issue 3, pp.184-194

International Journal of Microsystems and IoT

ISSN: (Online) Journal homepage: https://www.ijmit.org

Detection of Plant Diseases using Convolutional neural network & Augmented Techniques

Ashu Sharma, Shashank Pushkar

Cite as: Ashu Sharma, & Shashank Pushkar. (2023). Detection of Plant Diseases using Convolutional neural network & Augmented Techniques. International Journal of Microsystems and IoT, 1(3), 184–194. https://doi.org/10.5281/zenodo.8384892

Published online: 21 August 2023.

DOI: https://doi.org/10.5281/zenodo.8384892

International Journal of Microsystems and IoT
Vol. 1, Issue 3, pp.184-194; DOI: https://doi.org/10.5281/zenodo.8384892

Detection of Plant Diseases using Convolutional neural network & Augmented Techniques

Ashu Sharma, Shashank Pushkar

Birla Institute of Technology, Mesra, Ranchi, India

ABSTRACT

In present scenario plants disease detection is a major challenge. To accomplish it we have taken Brinjal & Tomato leaves for achieving better results of diseases detection from large set of images database. In this research work, we have used Machine Learning including Edge Detection Filters & CNN. For making more accurate analysis of Disease Detection, we have applied augmented techniques to make enriched images database. In addition to this (CNN) for automatic plant disease detection has also applied. In compared to state-of-the- art detection model, computed results have shown significantly improved disease detection analysis. Using CNN based technique along with augmented approach has provided accurate disease detection results.

KEYWORDS

AC; ANN; CNN; ConV; DC; DCT; DL; DNN; EA; ESCA; Keras; LSR; ML; SVM; Tensorflow; WSGI

1. INTRODUCTION

Among different countries in farming outputs, India has position within top five countries. As per 2017-15 data Agriculture employment was more than 50% being one of the reasons in India's GDP contribution near about 18% having first position in the highest net cropped area compared to USA & China. Plant infections have a negative role in the development of plants. Plants are important sources of energy to solve the problem of global warming. Various diseases in plants reduce productivity rate and economic advantages that affect across globe. To overcome this problem earlier details are important in the process of plants' growth. According to research work based on Generative Adversarial Network under Limited Training Set" [I6], In this paper, The generative network with gradient is merged with reduction in randomness to optimize the fore- casting plants disease results and different bugs on datasets. In paper authors have used image processing-based Plant Leaf diseases Identification and Classification [17], authors Identification of Grape disease using Artificial Intelligence approach in the paper used to focus on different diseases related to leaves that damage the tissues of plant branches as well included segmentation techniques. In this paper deep learning methods for classification of plant diseases [18] for analyzing the detection measures, for it used the Plant Village open dataset for analyzing plant diseases & complexity factors in terms of primitive & non-primitive features variations obtained in plants tissues.

2. GENERAL CAUSES FOR PLANTS DISEASES

To save plants against specific diseases, to meet the food production rate as required. As in follow up research work different diseases caused due to random climate changes and various infected agents are combined affect and increase in loss of food approximately 45% across globe. (Oerke 2006). There are various germs that produce problem in growth of Crop as a result loss proportion increase a lot, it makes loss in quality and quantity of crops globally; it brings lot of disease cure measures to protect crops loss.

(Zadoks 2015). Due to improper caring of plants that reduces food production in big scale it creates huge decline in different sectors of society at various levels.

For this reason, prior knowledge related to plants & crops including caring techniques are very important.

2.1 Detection Vs Diagnosis

Detection is the way to find something which is hidden not available directly; In Plants few diseases difficult to identify using basic visuals on affected segments of plants.

Diagnosis is the process through practical analysis on different patterns which are present on plant leaves or on other organs that may not be part of healthy plants or crops to recognize types of disease and cure mechanisms to protect from suffered issues.

2.2 Purpose behind Plants diseases Detection

Due to lack of prior knowledge among farmers to protect plants and crops from diseases, need to get better understanding of plants' health and about their diseases. To help users in detection and prevention of plant diseases with the use of using meta-heuristics and machine learning approach made things in right direction to generate proper and robust solution in diagnosing and curing plants and crops diseases.

2.3 Stakeholders

Farmers, Argo Industry Business Owners, Research Scholars, Scientists.

2.4 Traditional Approaches for Plants diseases Detection

Using Visual plant disease estimation used approach was Computer Vision in past inspire of lack of lab infrastructure and expertise, disease detection and classification accuracy increased up to 30%. In use of microscopic evaluation of morphology features to identify pathogens, it is used to determine structure of diseases by historical data. It works for detecting complete affected disease area. In Molecular diagnostic techniques Plant Pathology covers at the molecular level, including electron microscopy, tissue culturing in disease detection. Another technique is Serological assays for plant viruses proved very valuable detection tools for the plant viruses in past few years. Microbiological diagnostic techniques used to utilize pathogen infections in plant tissues.

2.5 Overview of Visual Estimation Approach

Through visuals methods recognizing affected diseased plants for examples fungus, germs based different characteristics of plants and crops. This approach is good for finding basic details of disease and if it gets on prior basis will protect plants from any random nature which causes problem in growth of plants. While following different visual techniques based on trained and tested methodologies worked well on this visual based disease detection. (Bock et al. 2010; Nutter 2010).

3. PROBLEM STATEMENT

The rational of the research is to study pathogens and its functions that are responsible for plants diseases, assess infection in plants using Image Processing and different AI techniques. In it focus is also on development of Desktop and Mobile Application for its easy use with maximum accuracy of the model to detect plants diseases and to prevent infection spread further.

4. LITERATURE SURVEY

During Literature Survey of proposed work in following paper, research based on picture element-based classification [1] deployed picture element-based recognition of disease affected for detecting unhealthy regions in leaf images is presented. Support Vector Machine has used to perform recognition of specific disease and reduces misclassification error.

According to published paper on Detection and Classification of Plant Leaf [2] tried to use disease identification using pattern retrieval approach used input as 2D image for further processing. Training & Testing used on specific diseases which were based on colour and shape. Segmented Image technique used K Means approach also applied in this research work.

In another work in published paper", An image classification approach [3] tried to be applied various steps as First, perform segmentation to segment input 2d image, next step, A supervised learning technique is applied to recognize the class of each picture element. And in the last step, reduced misclassification error for given input from previous step [15].

In other work in following paper Advance Computing Enrichment Evaluation of Cotton Leaf Disease [4] applied for Boundary based approach used Edges while used RGB colour model to recognize disease affected regions in which, the received images are used first. Then R, G, B colour based 2D Image is carried out to generate diseased region. In other paper through Image Processing [5] evaluated and designed model for autonomous disease detection and classification of Image Processing. In next paper authors worked on plant Leaf Disease Detection Using pattern retrieval [6] tried to implement non primitive texture-based approach of disease detection. conversion of different colour models under processing of Image, different colour models are used as R G B & HSV. [7] implementation of disease detection techniques to protect healthy plants from any side effect. Dedicated methods required for fulfilling expected accuracy to make plants healthy and free from any disease. Pattern Retrieval based Image classification was good to achieve desired results. In next paper deployed approach was Automated Colour Prediction of Paddy Crop Leaf using Image Processing [8] they have resolved few challenges as Acquisition for accurate image was major task as input image as plant leaves. Fetching images with complete information used memory size approx. 10-15 MB was key. High Resolution Camera was used to accomplish this task. Rest issues as intensity distribution illuminated image factors had also been resolved. To adjust threshold limit and other LCC related measurements worked well in this context. In next paperwork was based on Plant Leaf Disease Detection Using Image Processing Techniques [9] Challenging tasks were upfront as maintain Crop productivity and working towards more robust solution to detect and cure crops, researchers have gone through various techniques in this work, visual analysis is not enough to detect different patterns of diseases in crops for Agriculture people.

In next paper leaf classification disease for applied ANN Method [10] explained mechanism based on perceived by Human Eye to make decisions while applying methodologies on plant diseases. Through it disease identified but more modifications were needed to be apply to make solution more accurate. In next paper using pattern retrieval [11] as per this input image data set taken by different devices as camera process towards remote system through intelligent system and generated computer knowledge for identification of plant diseases derived as while finding affected segments of leaves have become easy to find.

In next survey paper based on Detection and Classification of Cotton Leaf Diseases [12] As per identification of cotton different diseased pattern using pattern retrieval techniques and involving AI approach solution have been tried to generate. In this work any noise error related to image reduction of such image-based noises were detected using threshold, edge, region, and cluster-based segmentation techniques. For such disease identification technique different colour models have worked in good way and improved detection of plants diseases in the improved manner. To get optimize solution towards Plants disease detection Threshold, edge, region, and cluster-based segmentation techniques are useful for such disease identification. These techniques work well on different color models have Worked in good way and improved detection of plants diseases in the specified manner.

In above figure1 Input data from Kaggle, three convolution layers used as pre-trained with sparse encoder for unsupervised learning, three convolution layers used for matching filters that are derived directly from the data.
CNN is useful in Visual based approach that are good at primary level for detection of diseases. During training phase of CNN, it works well on input weights and bias values performed well enough for disease detection. CNN works well from primary level of image processing to advanced implementation. It processes data with accuracy and at time learning also makes it possible to deal with local and global features both. It provides robust solution which is good for plants and crops disease related detection to handle with optimality and in the robust manner.
[13] In this research, it was based on computing model to work on real time and plants/crops health analysis can be done through networks. Based on IOT approach using networking connections, various sensors & through communicating net- work devices, worked well for information transfer in correct and accurate manner. It has several advantages anyone can work remotely to analyse their crops and plants. It has worked significantly to fulfil desire outcomes, used of IOT brought lot of improvement in disease prediction.

Usage of Deep learning (DL) was involved as a major contributor in this Research Work became an important machine learning approach that has been widely used for plants disease identification and curing mechanism. Inclusion of Deep Neural Network made this approach more intelligent to find disease patterns for micro analysis. As customization of hidden layers introduced into Neural Network for better learning and results accuracy [22].

[14] In this research work it has included various processes to modify the dataset. Dataset was consisted of Brinjal Images, inclusion of Deep Neural Network, at last, a huge high-quality training dataset of size more than 39,000 generated from approx. 350 sample images taken from the real field and more than 1350 high-resolution images used for plant disease prediction.

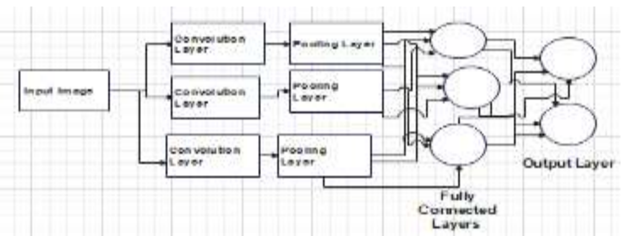

Fig.1 Prototype Workflow

Fig.2 Basic ML Steps

Figure 2 flow chart represents basic Machine Learning Steps as collection of Data Images used Kaggle dataset to enrich number of Data Images applied various augmented techniques also. In follow up step applied Data Preparation and Cleaning included filters to remove aliasing and noise effects. After that build Data Training including processing to generate classification results. These steps worked well to deploy prototype functionalities of Plant Diseases Detection [19].

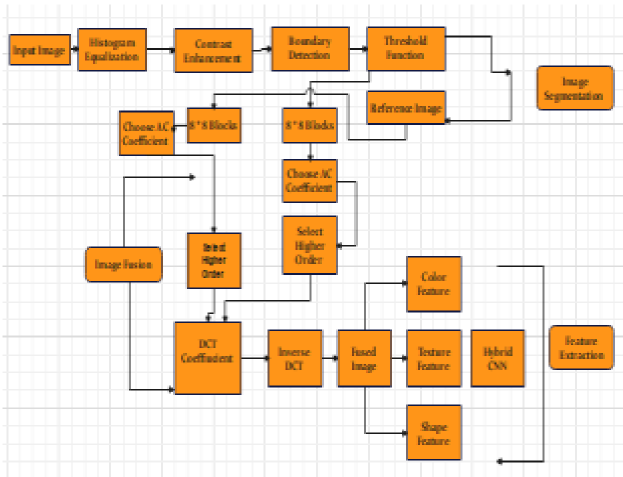

Fig. 3 Flowchart for Disease Detection Process

In above figure 3 major steps have included as Image Segmentation, Image Fusion and Feature Extraction. Image segmentation is a method of dividing image into small sub-groups on basis of Edge Based, Threshold Based, Region Based Segmentation [20]. These techniques are useful to identify important features. Deep neural network is also important for Image Segmented related task [23].

Usage of Region based segmentation technique is useful in different patterns detection in object. As process of a full image to retrieve required information from those further operations has performed by detector on bounding system derived by segmentation technique. It overall enhances the performance for processing an image with improved accuracy and decrease in inferences in histogram. As in different phases of Histogram Equalization, Contrast Enhancement, Boundary Decision and Threshold Function. Next step is Image Fusion, defined as collecting all the information from various images and their inclusion into lesser number of images. In its AC and DC coefficients in follow up with Inverse DCT has used. further merging of two or more images into one composite image integrates the information from separate images. As a result, is an input image that contains more information as compared to other individual images. After Image Fusion, Feature Extraction Phase has been included based on Colour, Texture and Shape Features. In follow-up Hybrid CNN to maintain record of global features into 1D data format and uses CNN towards local characteristics by 2D Dataset [21].

5. TOOLS & FRAMEWORKS

In research work, used dataset Kaggle for faster GPU processing Google Collab being used along with Jupiter note- book of Anaconda Distribution. Along with this used library as TensorFlow it has used for Deep Learning models, Kera's acted as an interface for the TensorFlow other libraries as NumPy as array processing package for scientific computing, Matplotlib is useful library for creating interactive visualizations included static and animated results, Pandas is used for data manipulation & analysis. Google Drive is used for data repository.

6. METHODOLOGY

The main purpose is to detect the diseased part of the plant leaves using Machine Learning, Evolutionary algorithm including convolutional neural networks that are implemented to classify the diseased part in optimum manner with minimum cost [24]. As per workflow, I have first collected Data from Kaggle, to make large dataset, I have also applied various augmented techniques then pre-process that data after performing cleaning operation through pre trained model, tried to build own model for validation purpose of outcomes. In follow-up performed training also and tried to classify as per number of classes. In second way used ML Steps to work on collected data, trained, and then processed it for generating results for detection of plants diseases. In another technique used three steps as Image Segmentation, Image Fusion including AC & DC coefficients and at the end Image Extraction to accurately detect diseased portion of Plant Leaves. In process of implementing these above-mentioned techniques, I have used two Plants for that as Brinjal and Tomato Leaves.

7. CNN Model Architecture and Configuration

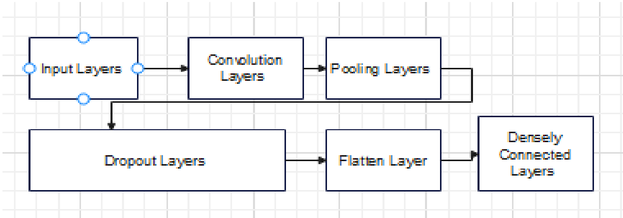

Fig.4 Model Architecture and Configuration

In figure 4 model Architecture and Configuration given.

7.1 Model Layers Input Layer:

Input shape: (IMG_WIDTH, IMG_HEIGHT, and IMG_DEPTH). This is where your input image data is fed into the network. IMG_WIDTH, IMG_HEIGHT, and IMG_DEPTH represent the width, height, and depth (number of channels, e.g., 3 for RGB images) of your input images.

7.2 Convolutional Layers (Conv2D):

Layer 1: 32 filters with a 3x3 kernel and ReLU activation function.

Layer 2: 64 filters with a 3x3 kernel and ReLU activation function.

Layer 3: 128 filters with a 3x3 kernel and ReLU activation function.

Layer 4: 256 filters with a 3x3 kernel and ReLU activation function.

These layers learn spatial hierarchies of features in the input data through convolution operations.

7.3 Pooling Layers (MaxPooling2D):

After each Convolutional layer, a max-pooling layer is applied. Each pooling layer reduces the spatial dimensions of the feature maps and helps in preserving important features while reducing computation. Pooling size: 2x2.

7.4 Dropout Layers:

After each pooling layer, a dropout layer with a dropout rate of 0.25 is applied. This helps in preventing over fitting by randomly setting a fraction of the input units to 0 at each update during training.

7.5 Flatten Layer:

After the last dropout layer, a flatten layer is used to convert the 2D feature maps into a 1D vector, which can be fed into a densely connected neural network.

7.6 Densely Connected Layers (Dense):

Layer 1: 512 units with ReLU activation.

Layer 2: The output layer with NUM_CLASSES units and a SoftMax activation function, which is typically used for multi-class classification tasks.

8. INPUT AND HIDDEN LAYER CONFIGURATION WITH POOLING AND PADDING

Table. 1 mentioned Input and hidden layers With Pooling and Padding and Table. 2 shows Input and hidden layers With Pooling and Padding.

Table. 1 Input and hidden layers With Pooling and Padding

Layer Type	**Filters/Kernels**	**Activation**	**PoolingPadding**
Input Layer	None	None	None None
Convolutional Layer	32 filters, 3x3 kernel	ReLU	2x2Max Valid
Convolutional Layer	64 filters, 3x3 kernel	ReLU	2x2Max Valid
Convolutional Layer	128 filters, 3x3 kernel	ReLU	2x2 Max Valid
Convolutional Layer	256 filters, 3x3 kernel	ReLU	2x2 Max Valid
Flatten Layer	None	None	None None

Table. 2 Input and hidden layers With Pooling and Padding

Layer Type	**Filters/Kernels**	**Activation**	**PoolingPadding**
Dense Layer 1	512 units	ReLU	None None
Dense Layer 2 (Output)	NUM_CLASSES units	Softmax	None None

9. DATA SET NATURE AND CNN UTILIZATION

9.1 Dataset Nature

Image Dimensions: The images in the dataset have a size of 224x224 pixels, and each image has 3 colour channels (RGB).

9.1.1 Number of Classes: There are 3 classes in the dataset, which means it's a multi-class classification problem where the goal is to classify each image into one of the 3 categories.

9.1.2 Data Augmentation: As mentioned earlier, during training, data augmentation techniques are applied to artificially increase the diversity of the training dataset. This includes random rotations, shifts, zooms, and flips.

9.2 CNN Utilization

9.2.1 Input Layer (Input Shape: 224x224x3): The input layer accepts images of size 224x224 pixels with 3 colour channels (RGB).

9.2.2 Convolutional Layers:

Layer 1 (32 filters, 3x3 kernel, and ReLU activation): This layer learns 32 different feature maps from the input images. Each feature map represents different low-level features such as edges, textures, or simple patterns. The ReLU activation function introduces non-linearity, allowing the model to learn complex relationships within these features.

Layer 2 (64 filters, 3x3 kernel, and ReLU activation): Building on the previous layer, this layer extracts more complex and higher-level features from the feature maps produced by the first Convolutional layer.

Layer 3 (128 filters, 3x3 kernel, and ReLU activation): This layer continues to capture even more abstract and complex features from the previous layer's outputs.

Layer 4 (256 filters, 3x3 kernel, ReLU activation): The deepest Convolutional layer in the network, it aims to learn the most complex and high-level representations of features within the images.

9.2.3 Max-Pooling Layers:

After each Convolutional layer, max pooling is applied (2x2 pooling size). Max-pooling reduces the spatial dimensions of the feature maps by half, which helps in reducing computational complexity and focuses on the most important features.

9.2.4 Dropout Layers:

After each max-pooling layer, dropout is applied with a rate of 0.25. Dropout helps in preventing over fitting by randomly deactivating a portion of the neurons during training.

9.2.5 Flatten Layer:

After the last dropout layer, the feature maps are flattened into a 1D vector, preparing them for the fully connected layers.

9.2.6 Densely Connected Layers:

Layer 1 (512 units, ReLU activation):

This densely connected layer processes the flattened features and learns complex patterns and relationships within the features.

Layer 2 (Output Layer) (3 units, SoftMax activation):

The final layer produces the output probabilities for each of the 3 classes using the SoftMax activation function. It assigns a probability to each class, and the class with the highest probability is the predicted class for the input image.

9.3 Training Parameters

9.3.1 Batch Size: During training, the model processes 8 images at a time. This is known as batch processing and helps in efficient training using available computational resources.

9.3.2 Epochs: The model undergoes 10 epochs, meaning it goes through the entire training dataset 10 times to learn and refine its parameters.

10. MATHEMATICAL MODEL OF DATA SETS WITH AUGMENTATION

10.1 Train_datagen (Training Data Generator)

Image Data Generator is a Keras utility that generates batches of augmented images during the training process. It helps in artificially increasing the diversity of your training dataset, which can improve the model's generalization.

Rescale=1./255: This scales the pixel values of your images to be in the range [0, 1]. It's a common practice to normalize the input data.

rotation range=20: This parameter allows random rotation of the input images by a maximum of 20 degrees. This helps the model become more robust to variations in object orientation.

width_shift_range=0.1 and height_shift_range=0.1: These parameters enable horizontal and vertical shifting of the images by a maximum of 10% of the total width or height. This simulates variations in the position of objects within the images.
zoom_range=0.2: This parameter allows zooming in or out on the images by a maximum of 20%. It introduces scale variations in the training data.

horizontal_flip=True and vertical_flip=True: These parameters enable horizontal and vertical flipping of images. It increases the diversity of the dataset by creating mirror images.

10.2 Test_datagen (Testing Data Generator)

This generator is used for preprocessing the test data but does not perform data augmentation. It only applies rescaling (rescale=1./255), which ensures that the pixel values of test images are also in the [0, 1] range and are consistent with the pre-processed training data.

11. RESULTS DISCUSSION

11.1 Augmented Techniques Result

To increase Image Datasets applied augmented techniques on 20 images wrt i/p 2 images as a sample. To accomplish it also used Keras & Tensor flow. One of the most important techniques to increase data set for proper data analysis to detect diseases as Augmented approach which is the technique of changing the size of dataset used for training a model as per user requirements. For better predictions, the deep learning models often require a lot of input training data. It was required to augment existing data to implement better outcomes. In scaling or resizing techniques, the image dimension has changed to the input defined size e.g., the width of the image can be doubled. In cropping also, a segment of the image is selected the cropped image is returned. In flipping, the image is flipped horizontally or vertically. In padding, the image (figure 5) is padded with a given value for every sides. The image rotation is performed randomly. The affine transformation also retains different points, straight lines, and planes. It can be used for different 2D type transformations.

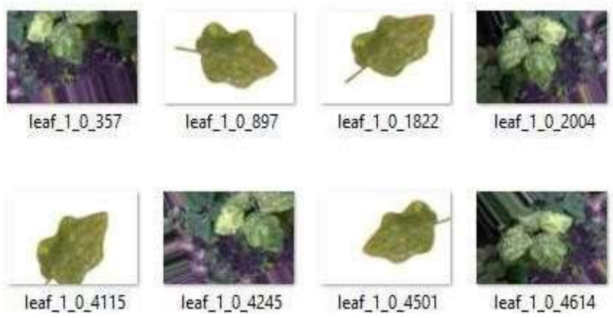

Fig.5 Sample Datasets of Brinjal Leaves Images

These are important techniques to increase data set for proper data analysis to detect diseases as Augmented approach which is the technique of changing the size of dataset used for training a model as per user requirements. For better predictions, the deep learning models often require a lot of input training data. It was required to augment existing data to implement better outcomes. In scaling or resizing techniques, the image dimension has changed to theinputdefined sizee.g., thewidth of the image can be doubled. In cropping also, a segment of theimage is selected the cropped image is returned. In flipping, the image is flipped horizontally or vertically. In padding, the image is padded with a given value for every side. The image rotation is performed randomly. The affine transformation also retains different points, straight lines, and planes. It can be used for different 2D type transformations.

11.2 RGB to HSV Conversion & Edge Detection Operators Result

I have performed various operators as Prewitt, Sobel, Laplacian, on an input augmented image and for detection of the object by color converted it to HSV color space. There are different Edge Detection Operators used as Sobel Edge, Prewitt, Laplacian. Sobel Operator characteristics are based on convolving the image with separable, small and integer valued filter. filter horizontally & vertically. The Sobel Edge has the advantage of providing edge response and reduces noise concurrently.

Prewitt operator has very much common characteristics-tics as the Sobel operator and is used for detecting edges on real and imaginary axis, this operator has not tendency to affect the picture elements that are near to the center of the mask.

Laplacian is dissimilar by nature from theprocedures, Unlike Sobel and Prewitt's edge detectors, it uses only one.

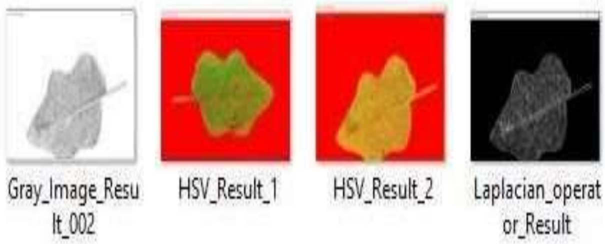

Fig.6 Result set 1 Brinjal Leaves Images

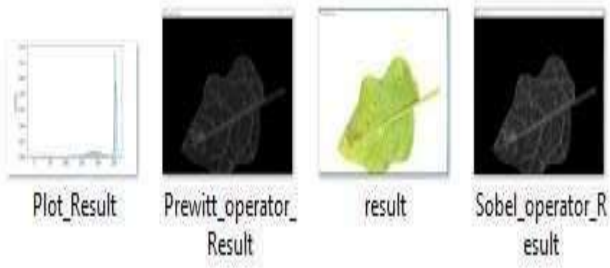

Fig.7 Result set 2 Brinjal Leaves Images

Kernel, the image generally derived has become Gaussian smoothed before applying this filter. Convolve Gaussian mask with the Laplacian mask & implement to the image in single pass. Figure 6 and Figure 7 shows result set 1 and result set 2 brinjal leaves images respectively.

11.3 Classification Result

To process classification following steps performed:
1. Conversion of Augmented Disease Images from one color model to another color model. 2. Implementation of Thresh oldling based Segmentation. 3. Implementation to find number of Augmented Images classes on basis of distribution of number of images. Output: Found 159 images belonging to 2 classes.

11.4 Model Training Phase Result

In this step tried to train the build model and computed accuracy results of model to predict classification and accuracy of model working. Here Ihave included one.

11.5 Tomato Plant Disease Detection Results

In this step tried to use Tomato Plant disease detection and make one User Interface that can run on Server IP.
Figure 8 shows Image Dataset Classification. Figure 9 shows
CNN Compilation Process, Figure 10 shows User Interface for Healthy & Fresh Tomato Leave, Figure 11 shows User Interface for Diseased Tomato Leaf.

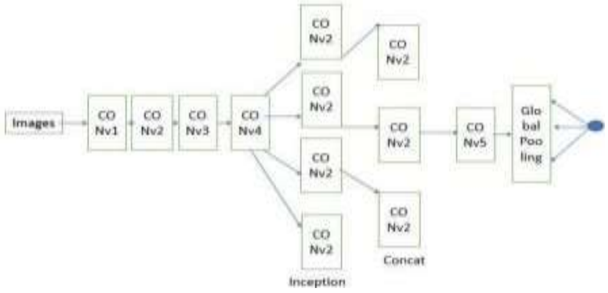

Fig.8 Image Dataset Classification

12. TRAINING PHASE

As in above given figures, included model Training & few raw results of Plant leaves disease detections. In training set target size is (128,128), batch size is 6, class mode 'categorical'; for valid set target size is (128,128), batch size is 3, class mode is 'categorical' have been used. In addition, step Per Epoch is 20 & epochs are 50 have been also applied. In next Image raw result values have been generated. Few resulting values are as follows:

Input posted: Tomato Bacterialspot.JPG Predicting class Time Quantum:

1/1- 0s

74ms/step

Result

Set1:

Raw result Values 1 = [[9.9999887e-01 6.7583996e-01 4.2829311e-01 8.7967366e-01 3.1313947e-01 9.4945007e-046.7335713e-01 4.3218904e-03 3.1741764e-02 8.2293677e-01]]

Result Set2:

Raw result Values2 = [[9.9994868e-01 1.0165111e-01 9.1468817e-01 5.1956487e-01 7.0995659e-01 5.6110375e-049.6219891e-01 1.7377082e-02 1.3240597e-01 3.1335396e-01]]

Fig. 9 CNN Compilation Process

User Interface At following Server with IP & Port Number 127.0.0.1:5002.0-127 belongs to first Octet is

Class A. In this research work, deployment of result has shown in form of web app interface also included front end technologies to show resulting user interface. Werkzeug consider as a WSGI toolkit, which follows requests and responses of objects included various functions. It is useful to create a web framework. The Flask framework applies it, major used packages in this work are Tensor Flow, Flask, Matplotlib, Conv2D, Maxpooling, Sequential, Flatten, Dense included Keras which is a Software Library that provides a Python interface for ANN. It acts as Interface for the Tensor Flow Library.

Fig. 10 User Interface for Healthy & Fresh Tomato Leave

In above given figure 10 a fresh and healthy Tomato Leaf with its resulting characteristics as leaf has not affected by any disease has shown. Diseased Tomato Plant Leaf including Precaution Strategies for below given figure 11 as detailed view of disease type & caring strategies to protect plant leaves from disease Bacterial Spot infections.

Fig. 11 User Interface for Diseased Tomato Leaf

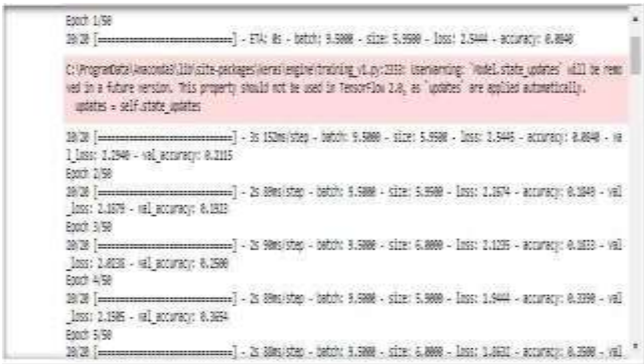

Fig. 12 Model Training

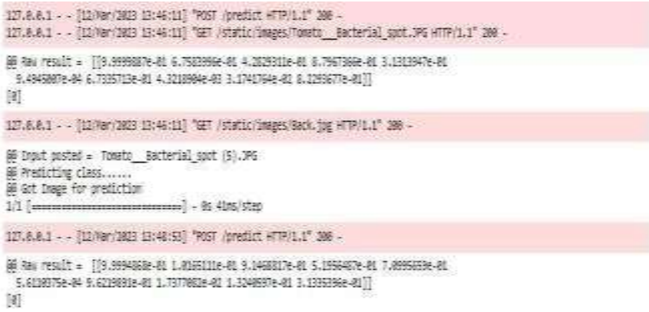

Fig. 13 Raw result for Tomato Bacterial spot

Figure 12 and 13 shows Model Training and Raw result for Tomato Bacterial spot respectively.

13. MODIFIED ARCHITECTURE WITH VGG16

13.1 VGG16 (Visual Geometry Group 16)

It is a strong architecture and can achieve good accuracy on various image classification tasks. It often reaches accuracies in the range of 80% to 90% or even higher on common datasets like ImageNet.

VGG16 is a Convolutional neural network architecture characterized by its deep stack of Convolutional layers. It consists of 16 weight layers, including 13 Convolutional layers and 3 fully connected layers.

13.2 Architecture

13.2.1 Convolutional Layers: VGG16 uses small 3x3 Convolutional kernels with a small stride and same padding, which allows it to capture fine details in images. It relies on stacking these layers to progressively learn complex features from low-level edges to high-level object parts.

13.2.2 Pooling Layers: After some Convolutional layers, VGG16 employs max-pooling layers to reduce spatial dimensions, which helps in reducing computational complexity and focusing on essential features.

13.2.3 Fully Connected Layers: The final part of VGG16 consists of fully connected layers that perform classification. These layers use features learned from earlier layers to make predictions about the input image's class.

13.2.4 Strengths: VGG16 is known for its simplicity and uniform architecture, making it easy to understand and implement. It performs well on a wide range of image classification tasks but may have many parameters.

13.3 Updated Results and discussion

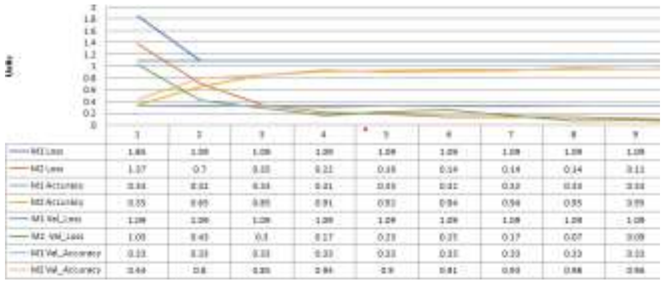

Fig. 14 Loss and Accuracy Comparison Graph

After inclusion of VGG16 as per given architecture improved accuracy from 33 % to 97 %, here are following factors improved accuracy (validated) as follows:

13.3.1 Depth and Capacity: VGG16's architecture is deep, consisting of 16 weight layers. This depth allows it to capture intricate patterns and features in the dataset, leading to improved accuracy.

13.3.2 Small Convolutional Kernels: VGG16 uses small 3x3 Convolutional kernels, which are effective in capturing fine details and intricate structures within images.

13.3.3 Pooling and Stride: Proper use of max-pooling layers and a small stride helps reduce spatial dimensions, making the network more robust to variations in object position and scale.

13.3.4 Regularization: Techniques like dropout are employed in the architecture, preventing over fitting and enhancing generalization.

13.3.5 Transfer Learning: Pre-training on large datasets like ImageNet provides a strong initialization, enabling the model to learn high-level features effectively.

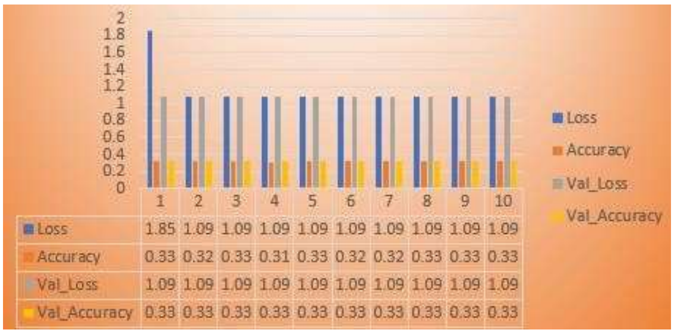

Fig. 15 Model Performance without VGG16

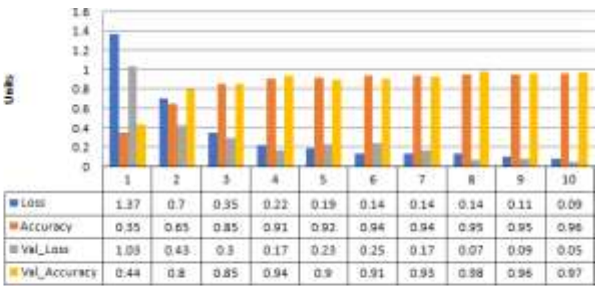

Fig. 16 Model Performance with VGG16

VGG16's success in achieving a 97% accuracy can be attributed to its depth, use of small Convolutional kernels, pooling layers, regularization, and transfer learning, which collectively allow it to learn complex representations and perform well on image classification tasks. Figure 14, 15, 16 shows Loss and Accuracy Comparison Graph, Model Performance without VGG16, Model Performance with VGG16 respectively.

14. CONCLUSION

After deployment of above methodologies using tools & Frameworks able to perform following tasks: Downloaded Im- age Dataset form Kaggle Repository applied Augmentation Techniques to increase dataset for more accurate results used CNN and augmentation techniques for disease detection. Also apply different Edge Detection Operators on different Colour Models for disease detection purpose, further apply Machine Learning including CNN and image segmentation and image fusion techniques for disease detection. Next build model on basis of training for classification purpose & User Interface to run it on Server. Finally debug and run the process using above methodologies for disease detection & classification results with accuracy. For further improvement with VGG16, the deep stack of Convolutional layers progressively abstracted image features, enabled the model to learn hierarchical representations, which are essential for achieving high accuracy. Training VGG16 on a sufficiently large and diverse dataset helped it generalize well to different types of images, boosting its accuracy.

15. SCOPE OF WORK

Scope of work in future context is to include more Plants except Brinjal & Tomato. For accuracy of Disease detection Model: spatial features must be included for images of leaves. Development of Desktop & Mobile application for usage purpose. Cloud Computing deployment is also required for live access of bulk dataset from anywhere at any time for stakeholders.

ACKNOWLEDGMENT

I would like to express my gratitude to my mentor for his advice and encouragement. He is an inspirational, supportive, and positive thinker. I am thankful for his valuable guidance during the planning and development of this research work.

REFERENCES

1. S. Madhgoria, M. Schikora, and W. Koch (2009), Pixel-Based Classification Method for Detecting Unhealthy Regions in Leaf Images, Proc. International Conference on Image Processing.

2. H. A. Bashish, M. Braik, and S. Bani-Ahmad (2010), A Framework for Detection and Classification of Plant Leaf and Stem Diseases, IEEE Transactions on Image Processing, 3456-3466 (Vol. 15).

3. M. Schikora, B. Neupane, S. Madhogaria, and W. Koch (2012), An image classification approach to analyze the suppression of plant immunity by the human pathogen Salmonella Typhimurium, BMC Bioinformatics, 146, (Vol. 13).

4. P. Revathi and M. Hemalatha (2012), Advanced Computing Enrichment Evaluation of Cotton Leaf Spot Disease Detection Using Image Edge Detection, Proc. International Conference on Computer Communication and Network Technology. https://ieeexplore.ieee.org/document/6395903/

5. P. S. Landge, S. A. Patil, and D. S. Khot (2013), Automatic Detection and Classification of Plant Disease through Image Processing, International Journal of Advanced Research in Computer Science and Software Engineering, 137-140, (Vol. 3).

6. M. Young (2013), The Technical Writer's Handbook, University Science.

7. S. D. Khirade and A. B. Patil (2015), Plant Disease Detection Using Image Processing, Proc. International Conference on Computing Communication Control and Automation. https:// doi.org/10/1109/ICCUBEA.2015.153
8. A. Singh and M. L. Singh (2015), Automated Color Prediction of Paddy Crop Leaf using Image Processing, Proc. International Conference on Technological Innovations in ICT for Agriculture and Rural Development. http://dx.doi.org/10.1109/TIAR.2015.7358526

9. M. Mathi, K. Aruli, S. M. Nizar, and A. S. Selvaraj (2015), A Survey on Plant Leaf Disease Detection Using Image Processing Techniques, International Research Journal of Engineering and Technology, 1206-1210, (Vol. 2), Corpus ID: 212579171

10. M. R. Manasi, R. Weginwar, N. Joshi, and A. B. Ingole (2015), Detection and Classification of Leaf Disease using Artificial Neural Network, International Journal of Technical Research and Applications, 1-3, (Vol.3), Corpus ID: 245128832

11. Y. Sanjana, A. Sivasamy, and S. Jayanth (2015), Plant Disease Detection Using Image Processing Techniques, International Journal of Innovative Research in Science, Engineering and Technology, 4716-4722, (Vol. 4).

12. B. S. Prajapati, V. K. Dabhi, H. B. Prajapati (2016), A Survey on Detection and Classification of Cotton Leaf Diseases, Proc. International Conference on Electrical, Electronics, and Optimization Techniques. https://ieeexplore.ieee.org/document/7755143
13. L. Gupta, K. Intwala, K. Khetwani, T. Hanamshet, and R. Somkuwar (2017), Smart Irrigation System and Plant Disease Detection, International Research Journal of Engineering and Technology, 2235-2240, (Vol.4).

14. A. Darwish, A. E. Hassanien, and S. Das (2019), A Survey of Swarm and Evolutionary Computing Approaches for Deep Learning, Springer Nature. https://doi.org/10.1007/s10462-019-09719-2

15. K. R. Hariharan, M. Sundar Anand, P. Mathikshara, A. Johnson, and M. R. Menaka (2020), Attention Embedded Residual CNN for Disease Detection in Tomato Leaves, Elsevier Applied Soft Computing, 86,105933. https://doi.org/10.1016/j.asoc.2019.105933

16. B. Gorad and S. Kotrappa (2020), Novel Dataset Generation for Indian Brinjal Plant Using Image Data Augmentation, Proc. International Conference on Frontiers in Engineering, Science, and Technology. https://iopscience.iop.org/article/10.1088/1757-899X/1065/1/012041

17. Luning.Bi and Guiping Hu (2020), Improving Image-based Plant Disease Classification with Generative Adversarial Network under Limited Training Set, Original Research, https://doi.org/10.3389/fpls.2020.583438

18. K.Gayatri,R.D.Kanti, V. Ch. Sekhar Rao Rayavarapu, B. Sridhar and V. R. G. Bobbili (2021), Image Processing and Pattern Recognition based Plant Leaf Diseases Identification and Classification, Journal of Physics: Conference Series, 1804012160. http:// dx.doi.org/10.1088/1742-6596/1804/1/012160

19. W. Albattah, M. Nawaz, A. Javed and M. Masood (2021), A Novel Deep Learning Method for Detection and Classification of Plant Diseases, Journal of Physics: ConferenceSeries, 1804012160.http:// dx.doi.org/10.1007/s40747-021-00536-1

20. P. S. Kanda, K. Xia, A. Kyslytysna, and E. O. Owoola (2022), Tomato Leaf Disease Recognition on Leaf Images Based on Fine-Tuned Residual Neural Networks, Plants, 2935 (Vol.11). https://doi.org/10.3390/plants11212935

21. S. Abisha, A. M. Mutawa, M. Murugappan, and S. Krishnan (2023), Brinjal Leaf Diseases Detection

based on Discrete Shearlet Transform and Deep Convolutional Neural Network, PLOS ONE, (Vol. 18).e0284021. https://doi.org/10.1371/journal.pone.0284021

22. A. Pandey, D. Prasad, K. Reddy, K. Venkatesh, A. Chand, V. Nath, (2023), Face Detection Using Convolutional Neural Network. In: Nath, V., Mandal, J.K. (eds) Microelectronics, Communication Systems, Machine Learning and Internet of Things. Lecture Notes in Electrical Engineering, (Vol 887). Springer, Singapore. https://doi.org/10.1007/978-981-19-1906-0_55
23. K. K. Sharma, J.B. Sharma, V. Nath, (2023), Automatic Brain Tumor Detection and Segmentation from MRI Using Fractional Sobel Filter and SOM Neural Network. In: Nath, V., Mandal, J.K. (eds) Microelectronics, Communication Systems, Machine Learning and Internet of Things. Lecture Notes in Electrical Engineering, (Vol 887). Springer, Singapore. https://doi.org/10.1007/978-981-19-1906-0_39
24. M. Bhardwaj, Aradhana, A. Kumar, P. Kumar, V. Nath (2021), Digital Implementation of Sigmoid Function in Artificial Neural Network Using VHDL. In: Nath V., Mandal J. (eds) Nanoelectronics, Circuits and Communication Systems. Lecture Notes in Electrical Engineering, (Vol 692). Springer, Singapore. https://doi.org/10.1007/978-981-15-7486-3_6

Ashu Sharma received his BTech degree in Computer Science and Engineering from Hindustan College of Science and Technology Mathura Uttar Pradesh, India in 2008 and MTech degree in Computer Science from Birla Institute of Technology Extension Centre Noida of (BIT Mesra), India in 2014. He is currently pursuing PhD at the Department of Computer Science and Engineering, Birla Institute of Technology, Mesra, India. His areas of interest are Image Processing, Artificial Intelligence, Machine Learning, Deep Learning, Java Programming, Database, and Software Testing.

Corresponding Author E-mail: ashu.jaipur.1985@gmail.com

Shashank Pushkar received his MCA degree from Assam Engineering College, Guwahati, and MBA degree from SMU Sikkim. He received his Ph.D. degree from National Institute of Technology, Jamshedpur. His areas of interest are Machine Learning, Optimization Heuristics, Data Mining and Computational Intelligence, Soft Computing Remote Sensing Data Analytics Software Engineering and Software Development.

E-mail: shashank@bitmesra.ac.in