Botanic Image Analyzer For Plant Leaf Identification And Disease Detection

Nalini Jagtap
Dept. of Computer Engineering
Dr Dy Patil Institute of Engineering Management and Research
Pune, India
nalinisjagtap@gmail.com

Saurabh Prakash
Dept. of Computer Engineering
Dr Dy Patil Institute of Engineering Management and Research
Pune, India
saurabhprakash7777@gmail.com

Kshitij Tripathi
Dept. of Computer Engineering
Dr Dy Patil Institute of Engineering Management and Research
Pune, India
kshitij825@gmail.com

Aniket Purushottam Udhane
Dept. of Computer Engineering
Dr Dy Patil Institute of Engineering Management and Research
Pune, India
aniketudhane10@gmail.com

Maheshwari Chittampalli
Dept. of Computer Engineering
Dr Dy Patil Institute of Engineering Management and Research
Pune, India
maheshwari.chittampalli@dypiemr.ac.in

Shivansh Shukla
Dept. of Computer Engineering
Dr Dy Patil Institute of Engineering Management and Research
Pune, India
shuklashivansh3998@gmail.com

Abstract– In a rapidly changing world, the need to efficiently and accurately identify, catalog, and monitor plant species is paramount for ecological preservation, scientific research, and sustainable agriculture. Traditional manual methods of plant identification are time-consuming and often prone to errors, hindering conservation efforts and limiting our understanding of plant life. To bridge this gap and contribute to ecological preservation, our Botanic Image Analyzer project seeks to develop an user friendly application using deep learning to identify plant species and disease if any. In this work, we have presented a convolutional neural network (CNN) for plant leaf disease detection. The CNN model is trained on various crops such as tomato, apple, grape, corn, etc using the Plant Village dataset, which contains images of twelve kinds of leaves with diseases like Scab, Black rot, Cedar rust and gray leaf spot. The extensive experimental results show that the proposed model can effectively recognize plant leaf diseases and achieve 99% classification accuracy. The results also demonstrate that the model is faster and more accurate than several existing deep CNN models.

Keywords- CNN(Convolutional Neural Network) SoftMax, MaxPooling,ReLU(Rectified-Linear Unit)

I. Introduction

Agriculture plays a vital role in any country's development in early phases. Also it is a necessity for a country to grow crops to feed their people. Looking at the current population of the world to have food reserves for any future drastic conditions is a wise action. Like Covid-19 pandemic India had a necessary amount of food reserve which led people in need to have meals to survive. India has an economy of 1.4 billion and to feed such a massive amount of people we have agriculture as the backbone of our economy.

To have a large amount of crop yield, we have to look for different measures to prevent it from diseases. To identify diseases in any crop or plant we don't have any automatic detector, we do this manually by looking into different species and changes in their leaf colors via naked eyes. This process is time as well as money consuming and people from villages still don't have any perfect way to do so. To do this, they have to hire an expert and it will be costly.

Thus, we introduce a Botanic Image Analyzer which detects any disease in a plant via scanning through the plant leaf. It helps predict the disease on the plant if found any.For this we used CNN model which is best suited for image classification. It is a class of deep learning neural networks.This model first processes raw images to gray-scale images which make them easy to study on and predict results.

This model will help farmers and researchers to analyze different crops suffering from diseases and type of diseases a crop is suffering from. Which will help timely cure of such diseases and prevent yield from ruining. It will reduce cost and time taken by conventional methods and revolutionize agriculture.

Disease	Type	Pathogen	Symptoms
Scrab	Fungal	Venturia inaequalis	light green spot,grows like velvety
Cedar rust	Fungal	Diplodia seriata	purple frog-eye like spot on leaves
Black rot	Fungal	G.juniperi-virginianae	pale-yellow spots on leaves

Table.1

II. Literature Review

The paper by G.Saranya, K.Meenakshi and M. Nithya presented a survey on different classification techniques that can be used for plant leaf disease classification. The authors reviewed various classification methods for identifying plant leaf diseases. They found that the k-nearest neighbor algorithm was the easiest and most appropriate for predicting the class of a leaf. However, they also noted that SVM had a limitation when the training data was not linearly separable, as it made it hard to choose the best parameters for the model.[9]

The paper by Emmanual Moupojou,Appolinair Tagne presented a survey to identify disease on raw field images used in conventional neural network , when the training and testing sets are same.The result shows that the existing model are not sufficiently accurate for disease detection and classification of plant leaf image collected directly from the field although the classification task results for field plant are better than those of plantdoc.[8]

Arfat Ahmad Khan proposed a general framework that can go for heavy neural network training to make high disease prediction of apple plants. It applies a series of filters to the image in sliding window fashion to detect more complex features.

Sachin D. Khirade et.al proposed a methodology used for the detection of plant diseases using their leaf images.The authors suggested a way to identify plant diseases by looking at the images,used some methods to get the characteristics of the diseased leaf and to categorize the type of plant disease. They grew crops with the help of image processing and feature extraction to study variation in the leaves. They recommended using artificial neural network methods to classify plant diseases, such as self-organizing maps, backpropagation algorithms, support vector machines, and so on.

Kewen Xia proposed a method in which utilization of cGAN(Conditional Generative Adversarial Network) The author suggested using cGAN to deal with the issue of not having enough or balanced data for training deep learning models. This helped to increase the number of leaf images for training deep neural networks more effectively.The paper was all about how we can tackle the problem of lack of sufficient training data or uneven class balance that could be found within datasets in performing Deep learning tasks.

A. Problem Statement

In a rapidly changing world, the need to efficiently and accurately identify, catalog, and monitor plant species is paramount for ecological preservation, scientific research, and sustainable agriculture. Traditional manual methods of plant identification are time-consuming and often prone to errors, hindering conservation efforts and limiting our understanding of plant life. To bridge this gap and contribute to ecological preservation, a Botanic Image Analyzer project seeks to develop an user friendly application using deep learning to identify plant species and disease if any.

B. Objective

To design and implement a web-based application for botanic image analysis and employ CNN models for accurate leaf image identification for enhanced disease detection capabilities using the Deep- CNN model with accuracy of 99% and above..

III. Material and Methods

This chapter contains full details on the construction of deep convolutional CNN (Conv-2D CNN) to study various crop diseases**.** The process of training and validating deep CNN models is presented with a detailed mathematical description**.**

A. DATASET AND PRE-PROCESSING

The Plant Village project provides a public dataset of plant leaf images. The dataset has 3171 RGB images of leaves that belong to four or five classes. The classes are based on the diseases that affect the leaves, such as black rot, scab, and cedar rust(Table.1). The other class is for healthy leaves that have no disease. The leaf images of size 256 × 256 for each class were taken with a simple background in a lab setting at different stages of plant growth. Fig.1 shows some examples of images from each class. The classes are named after the diseases or the healthy condition of the leaves. The Plant Village dataset is widely used in research. It is important to have diverse images of leaves in the dataset so that the models can learn different features during training. This helps to make the deep CNN model more generalizable. Augmentation is a technique to create artificial variations of the images. This work uses some transformations like shift, shearing, scaling, zoom, and flipping to change the images. These transformations make small changes in the images which help to add diversity to the training set.This helps to prevent overfitting and makes the model more robust and adaptable which helps model to achieve tolerance and generalization.Fig.2 shows how the data augmentation techniques work. Fig. 3 shows all the different species of plant in the dataset.

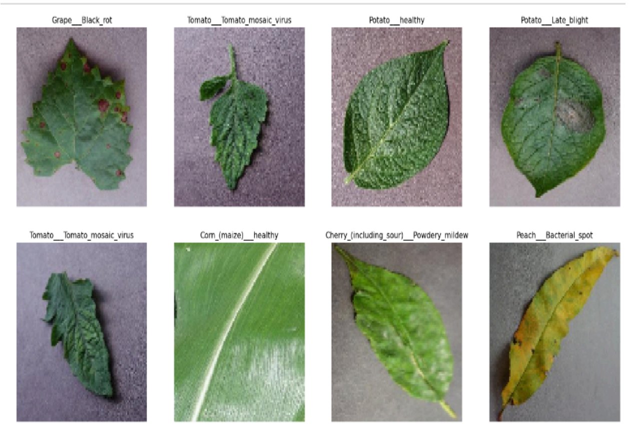

Grape__Black_rot
Tomato__Tomato_mosaic_virus
Potato__healthy
Potato__Late_blight
Tomato__Tomato_mosaic_virus
Corn_(maize)__healthy
Cherry_(including_sour)__Powdery_mildew
Peach__Bacterial_spot

Fig.(1)(Sample leaf image classes)

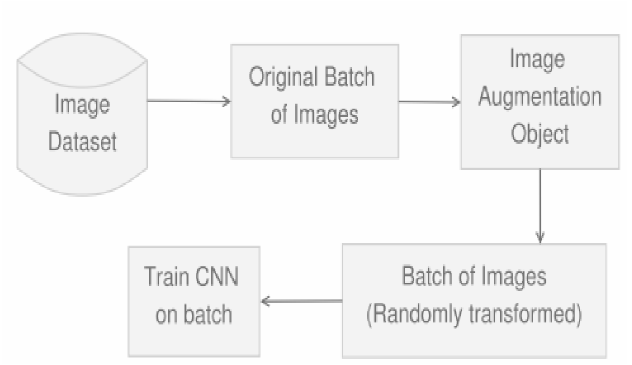


Image Dataset
Original Batch of Images
Image Augmentation Object
Batch of Images (Randomly transformed)
Train CNN on batch

Fig.(2)

B. DEEP CONVOLUTIONAL NEURAL NETWORK

Convolutional Neural Networks (CNNs) are a specific kind of Artificial Neural Network (ANN) that are frequently employed in the field of deep learning. The "depth" of a CNN is characterized by its multiple layers. A standard deep CNN is composed of several layers stacked on top of each other, which include convolutional layers with nonlinear activation functions, pooling layers, and fully connected layers. Deep CNNs have a distinct advantage over traditional machine learning methods in that they eliminate the necessity for manual feature extraction. They have proven to be effective in a variety of areas, including image and text classification, Natural Language Processing (NLP), and precision farming. The initial layers in deep CNNs are tasked with extracting low-level features from the input data.

1. CONVOLUTIONAL LAYER

The convolutional layer applies a filter (kernel) to an input image to detect features. The convolutional layer creates feature maps that show where and how much a feature is present in the previous layers by finding the local conjunctions. Basically, the convolutional layer has two parts: a linear convolution process and a non-linear activation function.

The convolution process works on images with multiple channels like RGB images [51] and is defined by (1):

$$conv(I,K)_{x,y} = \sum_{i=1}^{n_H}\sum_{j=1}^{n_W}\sum_{k=1}^{n_C} K_{i,j,k} I_{x+i-1,y+j-1,k} \qquad (1)$$

where the kernel K (f_h , f_w , n_C) combines with the image I (n_H , n_W , n_C) of different size but with similar no. of channels n_C and initiates a feature map O (o_H , o_W , z). The f_h represents the height and fw represents the width of the kernel.n_H denotes the height whereas n_W denotes the width of the given image.Usually, the kernel is a square matrix with an odd number of rows and columns, i.e., f_H=f_W = f .The formula for calculating the dimensions of the feature map[9] is given by equation (2):

$$Feature\_map\,(o_H, o_W, z) = \left(\lfloor\frac{n_H+2p-f}{s}+1\rfloor, \lfloor\frac{n_W+2p-f}{s}+1\rfloor, z\right) \qquad (2)$$

The most common activation function is the rectified linear unit (ReLU). ReLU doesn't activate all neurons, only activates some neurons at a given time. The neurons are activated when the output of the convolution unit or other linear transform is zero or positive. It is given by the formula (3):

$$f(z) = \max(0, z) \qquad (3)$$

2. POOLING LAYER

The pooling layer is responsible for reducing the resolution of the feature maps produced by the convolutional layers. It shrinks the size of activation maps that have many parameters. This way, it lowers the computational cost, prevents overfitting and speeds up the training process. The main pooling operations are max, min, average. However, max pooling is the most popular and selects the highest value from each input patch.The pooling operation only modifies the dimensions n_H and n_W ,n_C remains affected.The max pooling operation is expressed by the equation (4):

$$\text{Max Pooling} : yj = \max_{i \in Rj}(Pi) \qquad (4)$$

Where R indicates a receptive field containing p pixels the dimension of the generated feature map formula is shown by equation (5).

$$Feature\_map\,(o_H, o_W, n_C) = \left(\lfloor\frac{n_H+2p-f}{s}+1\rfloor, \lfloor\frac{n_W+2p-f}{s}+1\rfloor, n_C\right) \qquad (5)$$

3. FULLY CONNECTED LAYER

The convolutional neural network (CNN) has fully connected (FC) or dense layers that are similar to the layers in regular neural networks and usually connected at the end of a CNN to build output layers with a specific number of outputs for the better result. The Fully-Connected layers work on 1-D data. Flatten layer converts the 2-D output of previous layers into a 1-D format. The FC layers perform two kinds of functions: linear and non-linear transformations. These transformations can be written as in (6)-(7):

$$Z = W^T * X + b \qquad (6)$$
$$O = f(Z) \qquad (7)$$

here, X is the input feature map ,W is weight and B denotes bias terms and O denotes the output of the fully connected layer

To improve the prediction, the optimal weights are required to minimize the loss function. The most common method to obtain the best weights is gradient descent. Adam algorithm is an alternative method to achieve a more stable and smooth path while optimizing the gradients. It adjusts the learning rate based on the adaptive estimates of the first and second moments or lower order moments.

IV. Proposed Methodology

The proposed Conv-4 DCNN model is a deep CNN with 4 convolutional layers and two dense layers after four max-pooling layers. Each convolution layer and the first dense layer use ReLU as a non-linear activation function. The output layer uses softmax function to classify plant diseases into different classes. The softmax function assumes that each sample belongs to only one class. Figure 4 shows the flow of different layers and activation functions in the deep CNN model. To prevent overfitting, a dropout layer is added after the third max-pooling layer.

The dropout layer randomly removes some neurons from the network. This way, the network does not depend on any single feature and distributes the weights more evenly for better generalization.The first convolution layer, 32 filters having size 3 × 3 with valid padding and stride of 1 are applied to RGB images of size 256 × 256. This produces 32 feature maps of size 254 × 254. The number of channels in the feature maps is equal to the number of filters used. The first pooling layer, the feature maps are reduced by a factor of 2 using a kernel of size 2 × 2. This results in 32 feature maps of size 127 × 127. The same kernels are used at the higher layers. The convolution layers are increased with 128 filters of size 3 × 3 with valid padding. This creates 128 feature maps of size 28 × 28. The number of channels in the feature maps is the same as the number of filters used. The feature maps are then reduced by a factor of 2 using a kernel of size 2 × 2. This gives 128 feature maps of size 14 × 14. This improves the training accuracy to 99% and the test accuracy to 99.31%. The structure of the deep CNN model is illustrated in Fig.(5). The model is trained on about 35 lakh parameters. Table 2 summarizes the values and parameters of the deep CNN model.

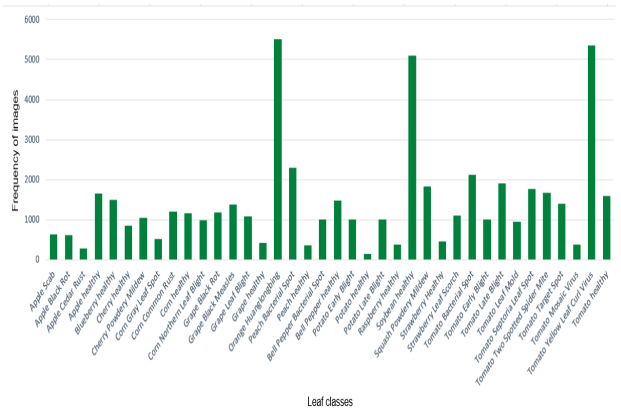


Fig. (3) (leaf classes)

A. *Block/Architecture Diagram*

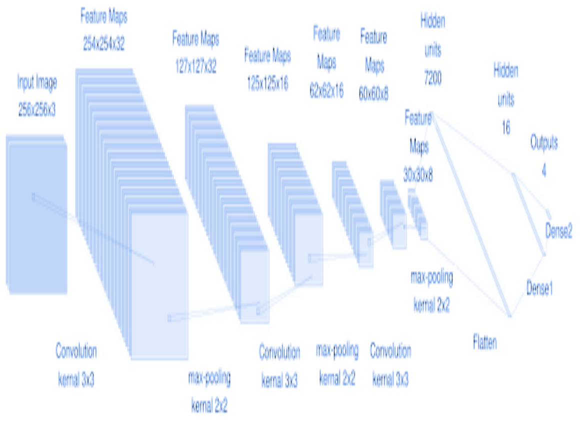
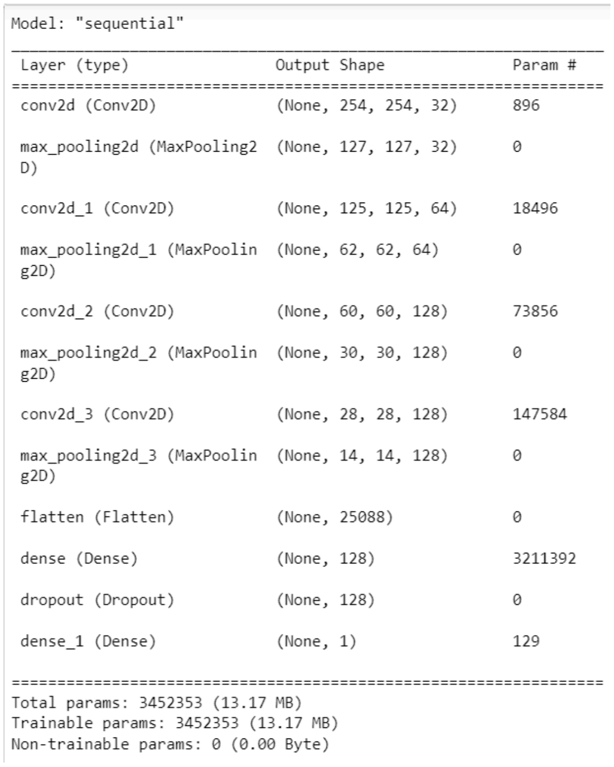


Fig.(4) (Layered architecture of CNN model)



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

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

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

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

 conv2d_2 (Conv2D)           (None, 60, 60, 128)       73856

 max_pooling2d_2 (MaxPoolin  (None, 30, 30, 128)       0
 g2D)

 conv2d_3 (Conv2D)           (None, 28, 28, 128)       147584

 max_pooling2d_3 (MaxPoolin  (None, 14, 14, 128)       0
 g2D)

 flatten (Flatten)           (None, 25088)             0

 dense (Dense)               (None, 128)               3211392

 dropout (Dropout)           (None, 128)               0

 dense_1 (Dense)             (None, 1)                 129

=================================================================
Total params: 3452353 (13.17 MB)
Trainable params: 3452353 (13.17 MB)
Non-trainable params: 0 (0.00 Byte)
```

Fig. (5) (Details about parameters)

Sr.No	filtersize	Layer	Dimension
1		Input	256 x 256 x 32
2	3 x 3	conv2d (ReLu)	254 x 254 x 32
3	2 x 2	maxpooling2d	127 x 127 x 32
4	3 x 3	conv2d (ReLu)	125 x 125 x 64
5	2 x 2	maxpooling2d	62 x 62 x 64
6	3 x 3	conv2d (ReLu)	60 x 60 x128
7	2 x 2	maxpooling2d	30 x 30 x 128
8	3 x 3	conv2d (ReLu)	28 x 28 x 128
9	2 x 2	maxpooling2d	14 x 14 x 128
10		Flatten	25088
11		Dense (ReLu)	128
12		Dropout-Rate	0.4
13		Dense (Softmax)	5

Table. 2

B. *Algorithms*

In this model we have used a deep CNN model which is best suited for image based classification. It is a class of deep learning neural networks. For our model we included four-Convolutional layers containing 4 Max-pooling layers. Each Convolutional layer applies filters to inputs thus extracting spatial features in the dataset. After every Convolutional layer, we applied ReLU activation function to make the model non-linear, which enables the model to capture and learn complex patterns.Max-pooling layers reduces the spatial dimensionality of the input feature map while retaining all the important information within some local neighborhood.

Flatten layers reduce the result to 1-Dimensional output to be further fed to fully connected layers. Fully connected layers learn the global patterns and relationships in the 1-dimensional result passed onto by a flatten layer. Fully connected layers are often called Dense layers, they perform global reasoning on the features extracted by convolutional and pooling layers.

This allows the network to capture complex relationships between features from different spatial locations. Dense layers enable complex decision making based on the extracted features.The details about the model has been shown in fig.5 and Table 1.

Once the machine learning model has been trained it is beneficial to save the model with all its parameters. Once it has been saved we can load the pre-trained model as and when needed and make classification without having the need to train the model from scratch a ga in. This saves both time and system resources which leads to faster classifications. Saving everything by creating a single archive file that contains everything using the TensorFlow , Save Model format (H5 format) and Creating a JSON file that only contains the architecture / configuration.

C. Flowchart

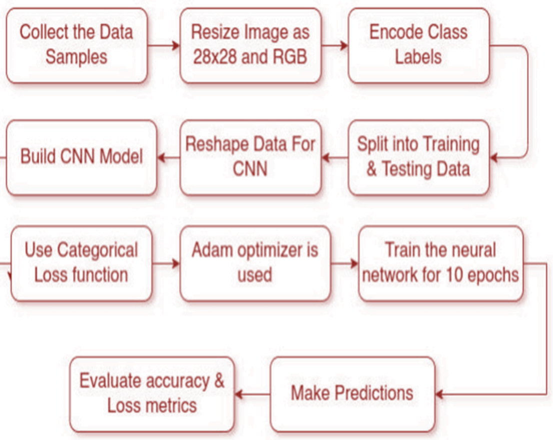

Collect the Data Samples
Resize Image as 28x28 and RGB
Encode Class Labels
Split into Training & Testing Data
Reshape Data For CNN
Build CNN Model
Use Categorical Loss function
Adam optimizer is used
Train the neural network for 10 epochs
Make Predictions
Evaluate accuracy & Loss metrics

Fig.(6)

V. Results

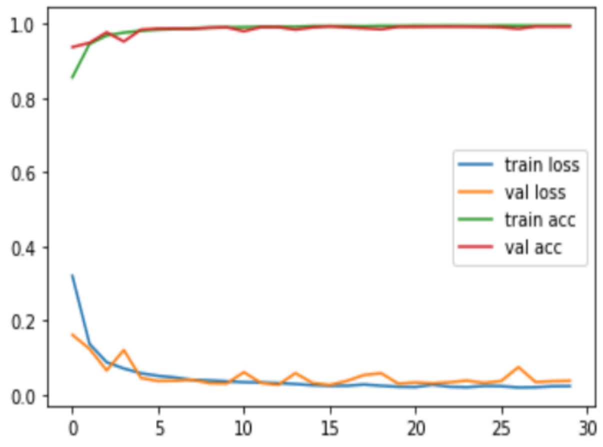
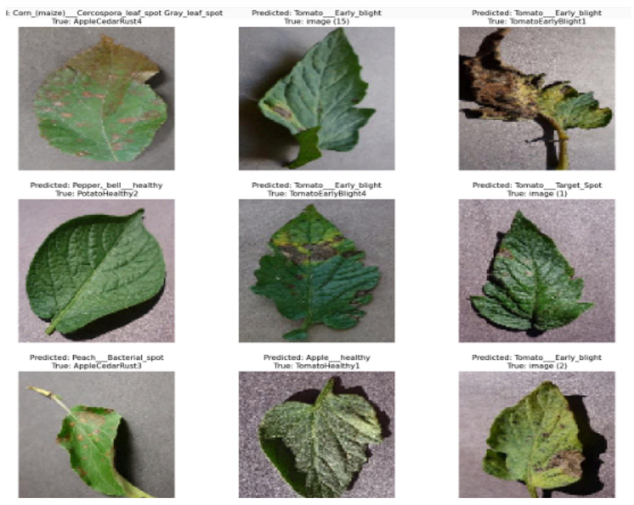


Fig.(7)
Here Y-axis represents training and validation accuracy and loss, X-axis represents epochs


: Corn_(maize)___Cercospora_leaf_spot Gray_leaf_spot
True: AppleCedarRust4
Predicted: Tomato___Early_blight
True: image (15)
Predicted: Tomato___Early_blight
True: TomatoEarlyBlight1
Predicted: Pepper,_bell___healthy
True: PotatoHealthy2
Predicted: Tomato___Early_blight
True: TomatoEarlyBlight4
Predicted: Tomato___Target_Spot
True: image (1)
Predicted: Peach___Bacterial_spot
True: AppleCedarRust3
Predicted: Apple___healthy
True: TomatoHealthy1
Predicted: Tomato___Early_blight
True: image (2)

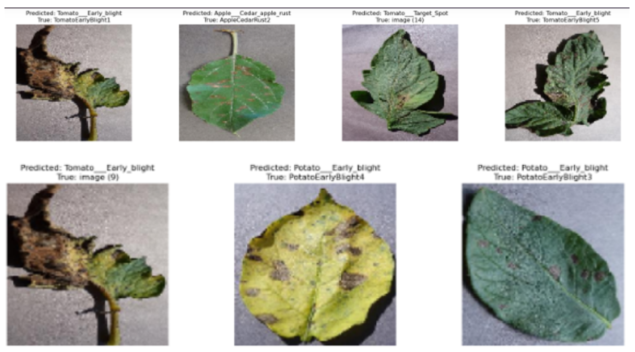
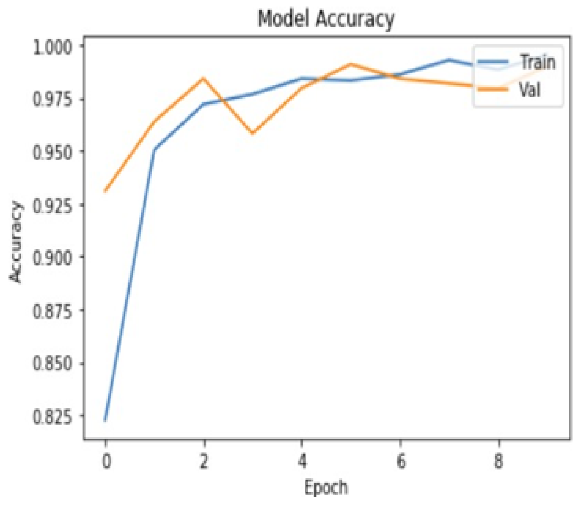

Predicted: Tomato___Early_blight
True: TomatoEarlyBlight1
Predicted: Apple___Cedar_apple_rust
True: AppleCedarRust2
Predicted: Tomato___Target_Spot
True: image (14)
Predicted: Tomato___Early_blight
True: TomatoEarlyBlight5
Predicted: Tomato___Early_blight
True: image (9)
Predicted: Potato___Early_blight
True: PotatoEarlyBlight4
Predicted: Potato___Early_blight
True: PotatoEarlyBlight3

Fig.(8)

(Fig.8) shows that the image given to the model for testing is predicted accurately, There is less error in the model .The predicted value and the actual value is approximately the same.

VI. Analysis and Discussion

1) Figures and Tables



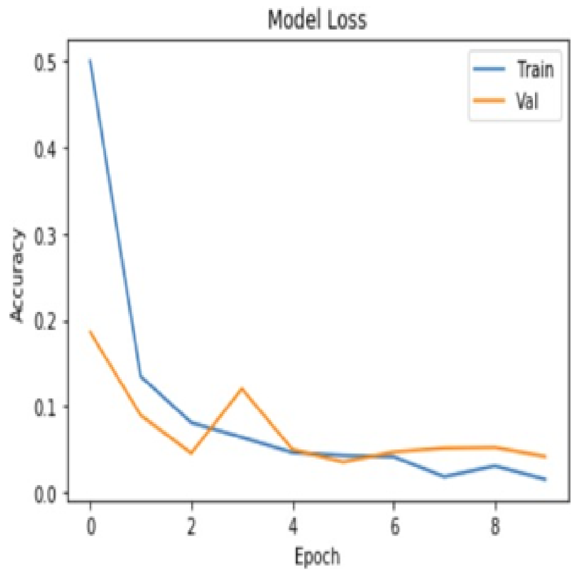
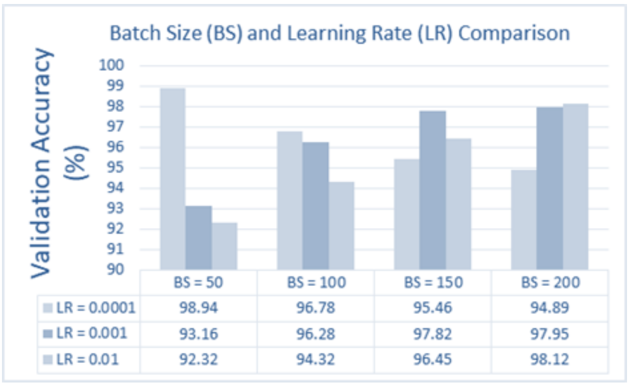


Fig.(9)

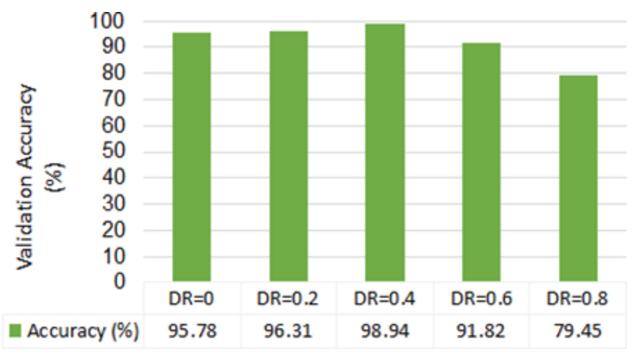
The analysis of the other models which was earlier made (Figure 9) indicates that the accuracy of the validation data is very low as they have used less number of images for training there model, also the parameter tuning was not good which resulted in loss of features or drop of parameters while training the model which may have played crucial role in enhancing the models accuracy. They have trained their model on 10 epochs.



	BS = 50	BS = 100	BS = 150	BS = 200
▨ LR = 0.0001	98.94	96.78	95.46	94.89
▨ LR = 0.001	93.16	96.28	97.82	97.95
▨ LR = 0.01	92.32	94.32	96.45	98.12

Fig.(10) (Validation accuracy reached at different learning rates and mini-batch sizes.)

In this experiment, we analyze the behavioral model under different small variables and learning rates. The results were obtained with dimensions of 50, 100, 150 and 200 and learning values of 0.0001, 0.001 and 0.01, as shown in Figure 10. The proposed model achieved the highest acceptance with a learning of 0.001 and a sample size of 50. It has been observed that small work is mostly related to small items.



	DR=0	DR=0.2	DR=0.4	DR=0.6	DR=0.8
🟩 Accuracy (%)	95.78	96.31	98.94	91.82	79.45

Fig.11(Validation accuracy achieved at different dropout rates.)

Dropout is a regularization technique, Dropout is an ongoing process that allows certain units to be ignored during model training, minimizing overwork. Check the effect of release rate (DR) by training the model on the release difference from 0.0 to 0.8 in steps of 0.2. Figure 11 shows the actual accuracy of the model for different output values. At 0.4 output, accuracy reaches a maximum of 99%. It can also be seen from the figure that higher output is the effect of exposure..

VII. Conclusion and Future Scope

A Deep CNN model can help in identifying image based classification. It can be used by a non-technical person as well as it's easy to use via an application. This proposed model is based on a similar neural network model. It's trained on 3 million plant leaf image datasets for 30 epochs over a batch size of 50. This model achieves an accuracy of 99% with training datasets and precision of about 98%.This model is optimized than the previous present models in terms of accuracy and the model trained on a large number of datasets as well as it has more layers compared to previous models.

Potential extensions of this work include collecting more leaf images of different plants from different regions, different sizes, with different diseases and organisms, and different cultivation methods. Large-scale data, along with the development of different images, will enable more rigorous testing and help develop standards for detecting diseases at different stages of various crops.

Acknowledgment

The authors are extremely thankful to the department of Computer Engineering, DR. D.Y. Patil Institute of Engineering Management and Research, Pune, Maharastra.

References

1. Fuentes, A., Yoon, S., Kim, S. C., & Park, D. S. (2016). Fuentes et al. (2016) developed a deep learning model that can detect and recognize various diseases and pests affecting tomato plants in real time using sensor data. Sensors, 16(11), 2022. (https://www.mdpi.com/1424-8220/17/9/2022)

2. Jansen, M., Wentzell, P. D., & Fjeld, T. (2019). reviewed the applications of machine learning techniques for measuring and analyzing plant stress responses in high-throughput phenotyping platforms. Trends in Plant Science, 24(2), 143-159. https://www.cell.com/trends/plant-science/fulltext/S1360-1385(15)00263-0

3. Lee, S. H., Chan, C. S., Wilkin, P., Remagnino, P., & Nelson, J. D. (2015).presented Deep-plant, a system that uses convolutional neural networks to identify plant species from images. In Proceedings of the British Machine Vision (BMVC)119.1-119.12). https://ieeexplore.ieee.org/abstract/document/735083927

4. Minervini, M., Scharr, H., & Tsaftaris, S. A. (2015). discussed the challenges and opportunities of image analysis methods for plant phenotyping, and proposed a framework for integrating them with machine learning models. IEEE Signal Processing Magazine, 32(4), 126-131. .https://ieeexplore.ieee.org/abstract/document/7123050

5. Pound, M. P., Atkinson, J. A., Townsend, A. J., Wilson, M. H., Griffiths, M., Jackson, A. S., ... & Murchie, E. H. (2017). demonstrated that deep machine learning can achieve state-of-the-art results in image-based plant phenotyping, and provided a benchmark dataset and a web-based tool for evaluation. GigaScience,6(10),1-10. https://academic.oup.com/gigascience/article/6/10/gix083/4091592

6. Singh, A., Ganapathysubramanian, B., Singh, A. K., & Sarkar, S. (2016). Machine learning for high-throughput stress phenotyping in plants. Trends in Plant Science, 21(2), 110-124. https://www.cell.com/trends/plant-science/fulltext/S1360-1385(15)00263-0

7. IEEE Paper: Machine Learning for Plant Leaf Disease Detection and Classification – A Review, L. Sherly Puspha Annabel ; T. Annapoorani ; P. Deepalakshmi, Published in: International Conference on Communication and Signal Processing (ICCSP)- 2019

8. Emmanuel Mourjou Apollinaire Tagne A Dataset of Field Plant Images for Plant Disease Detection and Classification With Deep Learning IEEE Access (29/03/2023)

9. G. Geetha, S.Samundeeswari, G.Saranya ,K.Meenakshi and M. Nithya Plant Leaf Disease Classification and Detection System Using Machine Learning(17/12/2020) ICCPET2020

Books:

1. Digital Image Processing" by Rafael C. Gonzalez and Richard E. Woods introduced the fundamentals and applications of digital image processing, covering topics such as image enhancement, restoration, segmentation, compression, and recognition.

2. Pattern Recognition and Machine Learning" by Christopher M.Bishop provided a comprehensive and accessible introduction to the theory and practice of pattern recognition and machine learning, covering topics such as Bayesian methods, neural networks, kernel machines, graphical models, and reinforcement learning.

3. Deep Learning" by Ian Goodfellow, Yoshua Bengio, and Aaron Courville offered a comprehensive and in-depth overview of deep learning, covering topics such as linear algebra, optimization, convolutional networks, recurrent networks, generative models, and natural language processing.

4. Computer Vision: Algorithms and Applications" by Richard Szeliski presented a modern and practical guide to computer vision, covering topics such as image formation, feature detection, segmentation, stereo, motion, 3D reconstruction, recognition, and vision systems