Leaf Disease Detection and Remedy Recommendation Using CNN Algorithm

https://doi.org/10.3991/ijoe.v18i07.30383

Prasanna Lakshmi K.[1](✉), Keerthana Reddy Mekala[1], Venkata Sai Rupa Sree Modala[1], Varsha Devalla[1], Avinash Bhargav Kompalli[2]
[1] Gokaraju Rangaraju Institute of Engineering and Technology, Hyderabad, India
[2] SRM IST, Chennai, India
prasannakompalli@gmail.com

Abstract—In many countries, agriculture has an excess impact on life of human beings and economic status. As leaf plays an important role, it gives information about the quantity and quality of agriculture yield depending upon the condition in advance. In this paper, we proposed the system which focuses on the detection of disease in plant leaves using Deep Learning approach. The main two processes that we used in our system are GUL Application and Deep Learning. We used CNN for classification of diseases and Remedy Recommendation upon diseased leaf selected from the Plant Village Dataset. This dataset consists of both healthy and unhealthy leaves. Our results show that the CNN Model achieves 96% accuracy for 8 epochs using Tensor flow.

Keywords—CNN, tensor flow, deep learning

1 Introduction

Plants have been facing many harmful diseases that cause significant reduction in the quantity and quality of agricultural products [2]. Plants can be affected in many ways such as soil fertility, climatic changes etc. Out of these disease plays a major role to affect the plants. Plants are affected by many kinds of diseases that target different parts of the plant body such as leaf, stem, seed, and fruit and so on [3]. A leaf is a plant organ exposed to the external environment. Leaves are the primary way in which plants interact with the atmosphere and take care of their basic needs.

The main function of the plant is photosynthesis which is done by the leaves, so leaves are very important to a plant's overall health and survival. Usually, plant diagnostics are performed with visual inspections by experts and if appropriate, bioassay is a second choice [4]. They are usually expensive and time consuming. Thence, looking for a quick, less costly and précised ways to smartly detect the diseases on the plant leaf. In our study, we are proposing a system which can be used to identify the particular type of disease a plant leaf might have. To do that, we have come up with an automated system using deep learning techniques, a system that will contribute in country' s agricultural development by automatically identifying and classifying diseases from the images of leaves. In this paper, we have focused on the identification of three leaf disease detection (bacterial spot, late blight and yellow leaf curl virus). The reason for choosing these three diseases is the prevalence of these diseases in daily essential plants like tomato, Pepper bell, Peach, Potato.

We have many technologies which used computer-based methodologies which are used to detect the diseases of plant leaves with the help of input given as leaf image. In the field of machine learning and Image Processing [29], object detection and location have recently gained a great deal of attention and many promising methods. Such automated methods have made way to solve the problems, but the toughest challenge is the consistency and robustness of the tests performed. Identifying the disease at early stage and suggesting the remedy to avoid maximum harm caused to the crop yield is important. So, we used CNN algorithm, to classify the disease and suggest remedies.

In Deep Learning, a Convolutional Neural Network (CNN or Convnet) is a class of Artificial Neural Network most commonly applied to analyze visual imagery. It is used for classification. It gives more accurate results. The classifier is evaluated using a mixture of different features. CNN consists of reference layers, the center layers and also Output layers.

This paper proposes such an approach that makes disease prediction and classification of the three mentioned diseases and also suggests remedy to save the plant. This remedy suggestion will be different for different disease (for e.g.: tomato leaves with spot disease to control use spray copper fungicide, whereas tomato leaf with yellow curl virus we grow then under greenhouse conditions). As mentioned in example every diseased leaf is treated in different manner. The novelty of the paper lies in the detection of leaf diseases using deep learning approach with high accuracy.

2 Literature review

In the paper "Deep learning for Image-Based Plant detection" the authors Prasanna Mohanty et al. [1], have introduced an approach to detect disease in plants by training a convolutional neural network. The CNN model [28] is trained to identify healthy and disease affected plants of 14 species. The model achieved an accuracy of 99.35% on test set data. When using the model on images procured from trusted online sources, the model achieves an accuracy of 31.4%, while this is better than a simple model of random selection, a more very different training data set can aid to increase the accuracy. Also, some other variations of model or neural network training may yield higher accuracy, thus paving a path for making plant disease detection easily available to everyone.

Malavika Ranjan et al. in the paper "Detection and Classification of leaf disease using Artificial Neural Network" [2], [27] proposed a vision to detect diseases in plants utilizing the captured image of the diseased leaf. Artificial Neural Network (ANN) is properly trained by choosing feature values to distinguish between diseased plants and healthy ones. The ANN model achieves an accuracy of 80%.

In the paper "Detection of unhealthy regions of plant leaves and classification of plant leaf diseases using texture features" by S. Arivazhagan[3] , disease identification process includes four main steps : first, a color transformation structure is taken for the given input RGB image, and then by means of a particular threshold value, the green pixels are European Journal of Molecular & Clinical Medicine ISSN 2515-8260 Volume 7, Issue 07, 2020 1607 detected and uninvolved, which is followed by segmentation process for obtaining beneficial segments the texture statistics are calculated. To classify the disease, we use the classifier used for the extraction of features.

Emanuel Cortes in his paper "Plant disease detection uses CNN and GAN" [4] proposed an approach to detect plant disease using Generative Adversarial networks. Background segmentation is used to ensure proper feature extraction and output mapping. It is seen that Gan's is used to classify diseases in plants, however segmentation based on background did not improve accuracy.

Kulkarni et al. in the paper "Applying image processing technique to detect plant diseases" [5], a methodology to detect plant diseases at a beginning stage and is very accurate, using artificial neural network (ANN) and diverse image processing techniques. As this particular view is based on ANN classifier for classification and Gabor filter for features extraction, it gives good results with an accuracy of 91%.

3 Proposed methodology

In this proposed system, we had interaction with the end user with a Graphical User Interface that helps a person to experience Automated Disease Detection of leaves. Our model is proposed to classify some of the plant leaf disease for the classification of plants affected by certain specific disease. The most common disease that occur in a plant by examining the leaves of plants are:

1. Bacterial Spot
2. Late Blight
3. Yellow curl virus

(a) (b) (c)

Fig. 1. (a) Bacterial spot, (b) Late blight, (c) Yellow curl virus

Our Model will only concentrate on the picture of leaf affected with these disease while training the model, it will learn specific properties that are existed in the subject of image and will mostly consider the properties to be a class. In this way different properties follow different classes. Therefore, to focus on the plants and diseases they got affected, we trained our model with the most valued images in a segmented manner. This model provides high accuracy in detecting leaf diseases.

3.1 Architecture

Figure 2 represents the workflow of the entire work. The leaf disease dataset was taken from SP Mohanty Repository, et al. [1]. The dataset was created manually by separating infected leaves into three different disease classes and Healthy leaves into other class. There are three diseases: Bacterial spot, Late blight, and yellow leaf curl virus, each having 200 images. The format of each image is .jpg. The size of the all-dataset images is set to same proportion before giving it to model. After that, the images are read and converted in to data array and are passed to CNN model. It gives out feature vector, then by applying some equations on feature vectors the leaf images are classifies into particular disease and suitable remedy is recommended.

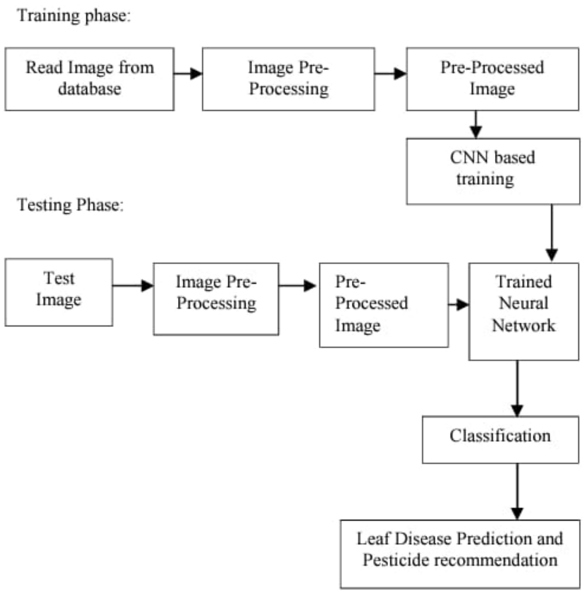

Fig. 2. Architecture of proposed methodology

3.2 Dataset

Data collection. The plant village dataset was taken from SP Mohanty Repository. The dataset used for this approach is manually created by separating the diseased leaves in to three classes as shown in Figure 1 and also a set of healthy leaves are used to make model learn the difference between diseased or not.

- The dataset was then split into two parts: training set, which comprises of the 80% data and test set, which comprises of the remaining 20%.
- The train set consists of 3200 images each class consisting of 800 images, whereas in test set total 800 images i.e., 20% of dataset.

Data preprocessing. In the preprocessing phase various transformations are applied on the training set for avoiding overfitting i.e., if we do not apply these transformations model will be performing well on the training set and will perform poorly on the test set. This process helps us to obtain the images with good content from scratch. The below technologies are used to obtain our Dataset which contains the most valued images.

- *Image acquisition:* It is the preparation and research collection of required type of images. The images captured may have different shapes and proportions, so the images are pre-processed and taken to the same size, removing noise, background and the unnecessary distortions. The output image obtained is given as input for the next module.
- *Segmentation:* Segmentation is done in order to obtain areas of interest, achieved using K-Means Clustering algorithm, Otsu detection, RGB conversion to HSI conversion. Segmentation is performed using Boundary and Spot Detection Algorithms, which help to identify the area of interest. The 8 connectivity pixels are considered for boundary detection and the respective algorithms are used. A collection of features is considered to form clusters using the K-Means clustering algorithm. The Otsu procedure is used to set the threshold. Threshold helps to create binary images from a grey scale, i.e., by setting a threshold value, and pixels below that threshold are set to 0 and above that threshold are set to 1.

After the images are segmented, then the next step is creating training data, we read the image content using imread() function from OpenCV library and the label array is attached to the output of above function. Every image goes through above process. This training data is stored in array format uses NumPy module. After this we load training data and start building the model.

3.3 Building the model

For building model, we had used CNN algorithm on images. Convolutional Neural networks will be taking the leaf images as an input and at the end will be classifying the type of the disease it was affected by as the output.

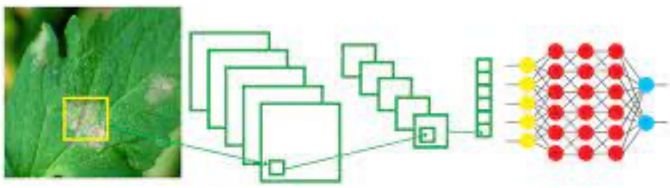

Fig. 3. Architecture of CNN algorithm

Recent advances in deep learning helped to evolve CNN as the best algorithm for Image classification. In the present algorithm two convolutional layers are employed, as more the convolutional layers the better the model performance. Activation function used is ReLU, this function if positive returns the input directly as an output else returns zero, it also helps in imparting non- linearity and helps in increasing model accuracy. As this paper is regarding multi class classification, categorical cross entropy has been utilized as the loss function, these functions are used for determining the error between the output of our model and the target value. To achieve the best possible accuracy, we had operated our model with "Adam" as the optimizer.

CNN consists of layers, the reference layers, central layers and the output layers. These layers perform some operations. They are Convolution, Max Pooling, Flattening and Full Connection.

Convolution. It is core building block of CNN.This layer performs a dot Product between two matrices, one is set of learnable parameters called as filter or kernel and the other is input image matrix. The kernel is spatially smaller than the image, but consists of same number of channels in depth wise.

$$W_{out} = \frac{W-F+2P}{S} + 1 \tag{1}$$

Formulae (1), If we have an input of size W x W x D and D_{out} number of kernels with a spatial size of F with stride S and amount of padding P, then the size of output volume is determined by W_{out}.

Figure 4 represents convolution process where we slide the filter over the next receptive field of the same input image by a Stride and do the same operation again. We will repeat the same process again and again until we go through the whole image. The output will be the input for the next layer. Convo layer also contains ReLU activation to make all negative value to zero.

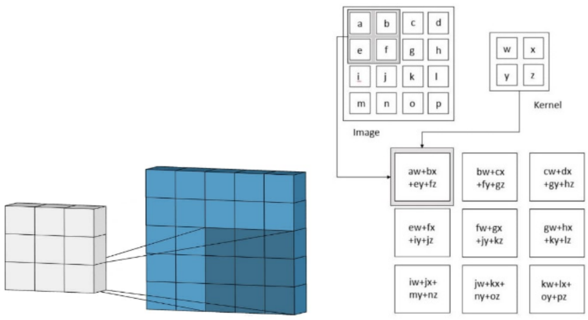

Fig. 4. Convolutions

Max pooling. Pooling layer is used to reduce the spatial volume of input image after convolution. It is used between two convolution layer. If we apply FC after Convo layer without applying pooling or max pooling, then it will be computationally expensive.

$$W_{out} = \frac{W-F}{S} + 1 \tag{2}$$

Formulae (2), If we have an activation map of size W x W x D, a pooling kernel of spatial size F, and stride S, then the size of output volume is determined by W_{out}.

Figure 5 shows First rectangle is the feature map and the second rectangle is the pooled feature map. In this stage we are considering maximum values to not to lose the valuable features of the images. Benefits of Max pooling are reduction in size and introduction of spatial invariance.

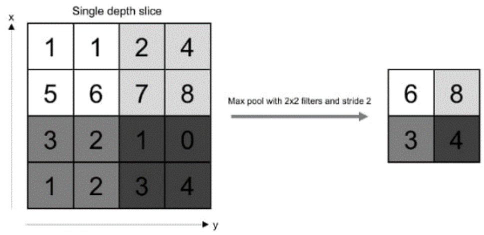

Fig. 5. Pooling

Flattening. In this step Pooled feature map is converted into a 1- dimensional array for providing the 1-dimensional array as input values to the neural network.

Figure 6 describes the flattening process, where above a 3x3 matrix is converted to one dimensional array which is then given as input to FC layer.

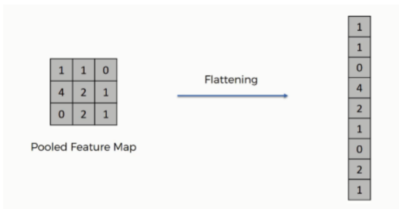

Fig. 6. Flattening

Fully connected layer. In this stage Flattened layer values will be given as input to the neural network, with the addition of fully connected layer neural network will be learning the non- linear combinations of these features and performs classification. This fully connected layer outputs the probability of each class and the class with the highest probability will be the output of the Convolutional Neural network.

4 Experimental analysis

4.1 Dataset

For this experiment, we used public dataset for leaf disease detection called Plant Village curated by Sharada P Mohanty. This dataset consists of healthy and unhealthy images of leaves having 38 classes out of which we have selected leaves from 10 classes that have 3 diseases in common.

The Dataset used for analysis is manually created by separating the diseased leaf images in to three classes namely: yellow curl virus, Bacterial spot and Late blight, which are found to be most common in Tomato, Potato, and Peach and Pepper bell leaves. Healthy Leaves are also included to make the model learn the difference between whether a leaf is diseased or not. The images in the dataset must be preprocessed i.e. Activities like, all leaf images will be of different proportions so they have taken in to same size. After this step the images are segmented which helps to create binary images from grayscale. The last step of preprocessing dataset is to read the image content using imread () i.e., function from OpenCV library and the respective label array is also attached to the above output. This data of every leaf image present in dataset is stored in array format using NumPy module.

In this experiment, a dataset containing total of 4000 leaf images are used. From this the dataset was split in to two parts, where one for training and other for testing. Here 80% of our dataset is used for training and remaining 20% for testing. The ratio size of training and testing datasets is 4:1.

Figure 7 shows the snapshot of leaf images used for this leaf disease detection experiment. It consists of Healthy and Unhealthy leaf images of four types of plants tomato, potato, and peach and pepper bell.

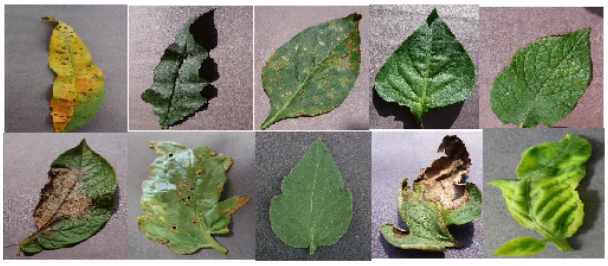

Fig. 7. Leaf images dataset

4.2 Parameters

For evaluating whether a model is best or not we have many metrics like Accuracy, Precision, Recall and F1 score. These all are metrics of classification. A confusion matrix is often used to describe performance of classification model. It consists of four parameters. They are True Positives, True Negatives, False Positives and False Negatives.

Figure 8 represents the confusion matrix where, True positives (TP) are correctly predicted positive values and True Negatives (TN) are correctly predicted negative values. False Positives (FP) and False negatives (FN) occurs when the actual class contradicts predicted class.

		Predicted class	
Actual Class		Class = Yes	Class = No
	Class = Yes	True Positive	False Negative
	Class = No	False Positive	True Negative

Fig. 8. Confusion matrix

Accuracy. It is most intuitive performance measure and it is simply ratio of currently predicted observations to toal observations.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (3)$$

Formulae (3) where (TP+TN) represents predicted observations and (TP+TN+FP+FN) are total observations.

Precision. It tells about how precise the model is out of those predicted positive, how many of them are actual positive. It is a good measure to determine, when the costs of False Positive are high.

$$Precision = \frac{TP}{TP+FP} \quad (4)$$

Formulae (4) where (TP) represents actual positives and (TP+FP) are total predicted positive observations.

Recall. It calculates how many of actual positives our model capture through labelling it as positive i.e., True Positives. It is a good metric to select best model when the cost associated with False Negative are high.

$$Recall = \frac{TP}{TP+FN} \quad (5)$$

Formulae (5) where (TP) represents actual positives and (TP+FN) are predicted positive observations which consists of wrongly predicted positives.

F1 score. It is a function of precision and recall. This is most needed when you want to seek balance between precision and recall. Useful when there is an uneven class distribution i.e., large number of actual negatives.

$$F1\ Score = \frac{Precision*Recall}{Precision+Recall} \quad (6)$$

Formulae (6) includes both previously defined precision and recall values.

4.3 CNN model

This CNN algorithm is implemented in python language with help of Visual Studio Code, using various modules like OpenCV, NumPy, Matplotlib, TensorFlow, sklearn.

Figure 9 describes that in our Model, layers that are added to the neural network for efficient processing, in this algorithm two convolutional layers are deployed, max pooling is chosen to capture the important features of the image, flatten layer is added to the network for transmitting one dimensional array as an input to the neural network two-dimensional array into a one-dimensional array for sending the values of this and finally output layer is added for classifying the type of leaf disease.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_20 (Conv2D)           (None, 62, 62, 32)        896

max_pooling2d_20 (MaxPoolin  (None, 31, 31, 32)        0
g2D)

conv2d_21 (Conv2D)           (None, 29, 29, 64)        18496

max_pooling2d_21 (MaxPoolin  (None, 14, 14, 64)        0
g2D)

flatten_10 (Flatten)         (None, 12544)             0

dense_20 (Dense)             (None, 1024)              12846080

dense_21 (Dense)             (None, 1)                 1025
```

Fig. 9. Layers in CNN model

4.4 Results

For evaluation of the algorithm performance in this paper accuracy had been taken into consideration as evaluation metric and loss values had been captured for finding out the trade-off between the true values and predicted values.

Figure 10 describes the variation between the training and the validation loss. Generally, loss value signifies the difference between the true value and the predicted value in neural networks and common loss function employed is entropy.

```
# Train and validation loss
epochs = range(1, len(acc) + 1)
loss = history.history['loss']
val_loss = history.history['val_loss']
plt.plot(epochs, loss, 'b', label='Training loss')
plt.plot(epochs, val_loss, 'r', label='Validation loss')
plt.title('Training and Validation loss')
plt.legend()
plt.show()
```

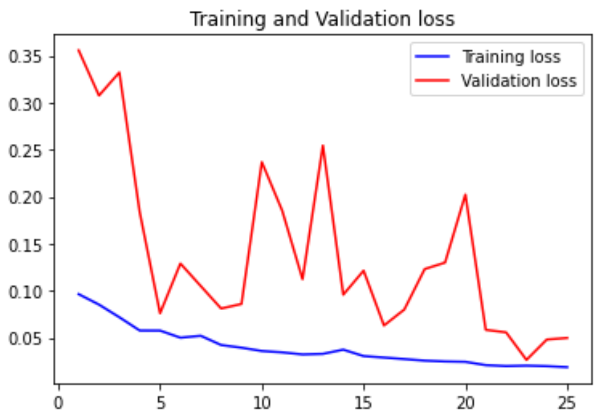

Fig. 10. Training and validation loss comparison graph

Figure 11 depicts the Training and validation accuracies.

Accuracy is the metric used for evaluating the model performance. We can use these learning curves to diagnose whether the model is over fit, under fit or best fit based on the model performance on training and validation sets. A best fit model is that which performs well on both training and validation set. Here our model is close to best fit model.

```
# Train and validation accuracy
epochs = range(1, len(acc) + 1)
acc = history.history['accuracy']
val_acc = history.history['val_accuracy']
plt.plot(epochs, acc, 'b', label='Training accurarcy')
plt.plot(epochs, val_acc, 'r', label='Validation accurarcy')
plt.title('Training and Validation accurarcy')
plt.legend()
plt.show()
```

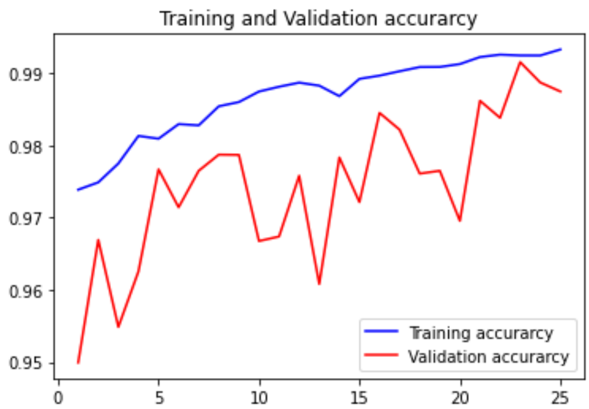

Fig. 11. Training and validation accuracy comparison graph

Table 1 represents scores obtained for different metrics of classification model based on number of test samples given to the model. Metrics included are Accuracy, Precision, Recall, F1 score.

Table 1. Metric values for classification model

No. of Test Samples	Accuracy	Precision	Recall	F1 Score
100	96.0	96.2250	96.0	96.0160
200	96.0	96.1851	96.0	96.0319
300	95.6667	95.9192	95.6667	95.7057
400	95.75	96.0455	95.75	95.7914
500	95.1999	95.5704	95.1999	95.2466
600	94.8334	95.2203	94.8333	94.8622
700	94.5714	94.9565	94.5714	94.5970
800	94.75	95.1347	94.75	94.7865

Figure 12 describes that the Precision, Recall and F1 score graphs helps us to have a finer grained idea on how well a classifier is doing, as opposed to just looking the overall accuracy. Precision metric with higher performance shows how precise our model is.

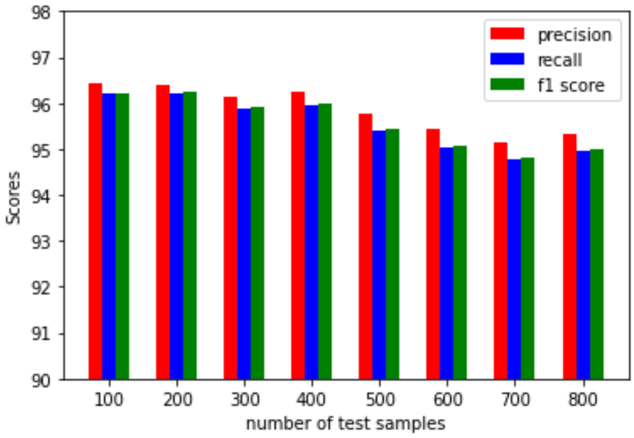

Fig. 12. Comparison between precision recall and F1 score

Figure 13 depicts that precision scores are better than accuracy values. This proves that not always accuracy stands as lead metric. Accuracy and Precision reflect how close a measurement is to an actual value. Accuracy reflects how close a measurement is to a known or accepted value whereas precision reflects how reproducible measurements are.

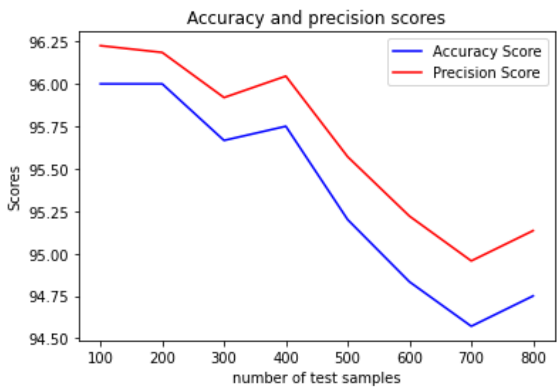

Fig. 13. Precision vs. accuracy graph

Figure 14 represents the level of accuracy for the number of test samples taken. The reason for the decline in graph can be the small data can be linearly separable whereas the large data is not, so the algorithm failed. One more thing is maybe the later are too hard to classify and also the source of the image shows effect on accuracy.

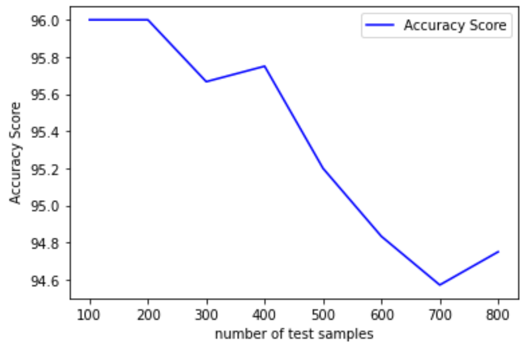

Fig. 14. Number of test samples vs. accuracy

Figure 15 represents the GUI where the user given input is analyzed by our model and the output i.e. Leaf status (Healthy or Unhealthy) and if unhealthy then Disease name is shown.

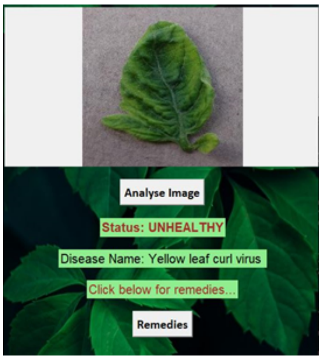

Fig. 15. GUI for leaf disease detection

Figure 16 shows the remedies for the disease the leaf is affected These remedies are vary from disease to disease.

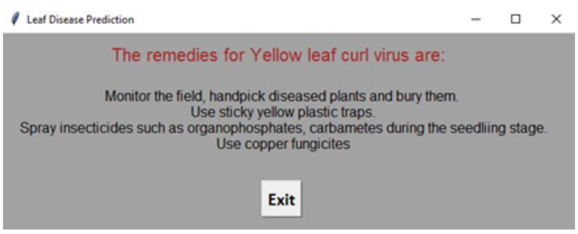

Fig. 16. GUI for remedies

5 Conclusion and future intensifications

This research has contibutes to detecting the diseased leaves,thus solving the problem of agricultural and economic loss .It helps the farmers to have an idea on type of disease the leaf got effected by and situable remedies are provided.It is clear from theresults of the model that the convolutional neural network is capable of acheving results on a dataset using supervised learning considering the accuracy as main quantitative metric where the results of this research achieved high accuracy of up to 96%. Convolution neural network algorithm are accustomed to increase recognition rates within the classification process. The results of this research will be helpful in detecting leaf diseases prior to its damage that inturn protects the crop.

In this research we demonstrated only some sorts of diseases which were commonly known. The recommendations for the future work can be extension for more diseases and also with remedies suggestion. Integration of model with web or Mobile app for easy reach out to the farmers for leaf disease detection without any other human interaction.

6 References

[1] Guo, Yan, et al. "The Study of Locating Diseased Leaves Based on RPN in Complex Environment." Journal of Physics: Conference Series. Vol. 1651. No. 1. IOP Publishing, 2020. https://doi.org/10.1088/1742-6596/1651/1/012089

[2] Swain, Sripada, Sasmita Kumari Nayak, and Swati Sucharita Barik. "A Review on Plant Leaf Diseases Detection and Classification Based on Machine Learning Models." Mukt shabd 9.6 (2020): 5195-5205.

[3] Nisar, Naina, et al. "Image based Recognition of Plant Leaf Diseases: A Review." 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC). IEEE, 2020. https://doi.org/10.1109/I-SMAC49090.2020.9243434

[4] Madana Mohana, R., C. Kishor Kumar Reddy, and P. R. Anisha. "A Study and Early Identificatıon of Leaf Diseases in Plants Using Convolutional Neural Network." Smart Computing Techniques and Applications. Springer, Singapore, 2021. 693-709.

[5] Rao, Anusha, and S. B. Kulkarni. "A hybrid approach for plant leaf disease detection and classification using digital image processing methods." The International Journal of Electrical Engineering & Education (2020): 0020720920953126. https://doi.org/10.1177/0020720920953126

[6] Kumari, Ch Usha, S. Jeevan Prasad, and G. Mounika. "Leaf disease detection: feature extraction with K-means clustering and classification with ANN." 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC). IEEE, 2019. https://doi.org/10.1109/ICCMC.2019.8819750

[7] Kothari, Jubin Dipakkumar. "Plant Disease Identification using Artificial Intelligence: Machine Learning Approach." Jubin Dipakkumar Kothari (2018). Plant Disease Identification using Artificial Intelligence: Machine Learning Approach. International Journal of Innovative Research in Computer and Communication Engineering 7.11 (2018): 11082-11085.

[8] Sujatha, R., et al. "Performance of deep learning vs machine learning in plant leaf disease detection." *Microprocessors and Microsystems* 80 (2021): 103615. https://doi.org/10.1016/j.micpro.2020.103615

[9] Dhingra, Gittaly, Vinay Kumar, and Hem Dutt Joshi. "Study of digital image processing techniques for leaf disease detection and classification." *Multimedia Tools and Applications* 77.15 (2018): 19951-20000. https://doi.org/10.1007/s11042-017-5445-8

[10] Ozguven, Mehmet Metin, and Kemal Adem. "Automatic detection and classification of leaf spot disease in sugar beet using deep learning algorithms." *Physica A: Statistical Mechanics and its Applications* 535 (2019): 122537. https://doi.org/10.1016/j.physa.2019.122537

[11] Thangaraj, Rajasekaran, et al. "Artificial intelligence in tomato leaf disease detection: a comprehensive review and discussion." *Journal of Plant Diseases and Protection* (2021): 1-20. https://doi.org/10.1007/s41348-021-00500-8

[12] Kaur, Sukhvir, Shreelekha Pandey, and Shivani Goel. "Plants disease identification and classification through leaf images: A survey." *Archives of Computational Methods in Engineering* 26.2 (2019): 507-530. https://doi.org/10.1007/s11831-018-9255-6

[13] Atila, Ümit, et al. "Plant leaf disease classification using EfficientNet deep learning model." *Ecological Informatics* 61 (2021): 101182. https://doi.org/10.1016/j.ecoinf.2020.101182

[14] Dhingra, Gittaly, Vinay Kumar, and Hem Dutt Joshi. "A novel computer vision based neutrosophic approach for leaf disease identification and classification." *Measurement* 135 (2019): 782-794. https://doi.org/10.1016/j.measurement.2018.12.027

[15] Zhong, Yong, and Ming Zhao. "Research on deep learning in apple leaf disease recognition." *Computers and Electronics in Agriculture* 168 (2020): 105146. https://doi.org/10.1016/j.compag.2019.105146

[16] Xie, Chuanqi, et al. "Detection of early blight and late blight diseases on tomato leaves using hyperspectral imaging." *Scientific reports* 5.1 (2015): 1-11. https://doi.org/10.1038/srep16564

[17] Kurmi, Yashwant, et al. "Leaf image analysis-based crop diseases classification." *Signal, Image and Video Processing* 15.3 (2021): 589-597.

[18] Bansal, Prakhar, Rahul Kumar, and Somesh Kumar. "Disease Detection in Apple Leaves Using Deep Convolutional Neural Network." *Agriculture* 11.7 (2021): 617. https://doi.org/10.1007/s11760-020-01780-7

[19] Zhao, Shengyi, et al. "Tomato Leaf Disease Diagnosis Based on Improved Convolution Neural Network by Attention Module." *Agriculture* 11.7 (2021): 651. https://doi.org/10.3390/agriculture11070651

[20] Basavaiah, Jagadeesh, and Audre Arlene Anthony. "Tomato Leaf Disease Classification using Multiple Feature Extraction Techniques." *Wireless Personal Communications* 115.1 (2020): 633-651. https://doi.org/10.1007/s11277-020-07590-x

[21] Chowdhury, Muhammad EH, et al. "Automatic and Reliable Leaf Disease Detection Using Deep Learning Techniques." *AgriEngineering* 3.2 (2021): 294-312. https://doi.org/10.3390/agriengineering3020020

[22] Lu, Jinzhu, et al. "Detection of multi-tomato leaf diseases (late blight, target and bacterial spots) in different stages by using a spectral-based sensor." *Scientific reports* 8.1 (2018): 1-11. https://doi.org/10.1038/s41598-018-21191-6

[23] neural Hu, Gensheng, et al. "Identification of tea leaf diseases by using an improved deep convolutional network." *Sustainable Computing: Informatics and Systems* 24 (2019): 100353. https://doi.org/10.1016/j.suscom.2019.100353

[24] Saleem, Muhammad Hammad, Johan Potgieter, and Khalid Mahmood Arif. "Plant disease detection and classification by deep learning." *Plants* 8.11 (2019): 468. https://doi.org/10.3390/plants8110468

[25] Ramcharan, Amanda, et al. "Deep learning for image-based cassava disease detection." *Frontiers in plant science* 8 (2017): 1852. https://doi.org/10.3389/fpls.2017.01852

[26] Mojjada, Ramesh Kumar, et al. "Detection of plant leaf disease using digital image processing." *Materials Today: Proceedings* (2020). https://doi.org/10.1016/j.matpr.2020.11.115
[27] Kalidindi, A., Kompalli, P., Bandi, S., & Anugu, S. (2021). CT Image Classification of Human Brain Using Deep Learning. https://doi.org/10.3991/ijoe.v17i01.18565
[28] El Abkari, S., Jilbab, A., & El Mhamdi, J. (2020). RSS-based Indoor Positioning Using Convolutional Neural Network. International Journal of Online and Biomedical Engineering (iJOE), 16(12), pp. 82–93. https://doi.org/10.3991/ijoe.v16i12.16751
[29] Saudagar, A. K. J. (2020). Biomedical Image Compression Techniques for Clinical Image Processing. International Journal of Online and Biomedical Engineering (iJOE), 16(12), pp. 133–154. https://doi.org/10.3991/ijoe.v16i12.17019

7 Authors

K Prasanna Lakshmi is working as a Professor and Head of Department in Information Technology Department in Gokaraju Rangaraju Institute of Engineering and Technology, Hyderabad. She received the Bachelor's Degree in Mechanical Engineering (2000), Master's Degree in Computer Science (2002) and Ph.D in Computer Science (2016) in the area of Data Stream Mining. She has total 18 years of teaching and 10 years of research experience. She is a reviewer for various conferences, journals and books, Editorial Board Member for IGI Global Publications. Her research interests include Data Science, Artificial Intelligence, Data Stream Mining, Web Mining, Big Data Analysts, Social Networking. Published many research articles in renowned national and international journals and conferences (email: prasanna-kompalli@gmail.com).

Mekala Keerthana Reddy is an UG student finishing her last year at Gokaraju Rangaraju Institute of Engineering and Technology, Hyderabad, India. Currently she is doing an internship at NCR corporation as a Software Engineer. Keerthana's research concentrate on Data Science and Automation (email: keerthanareddy2111@gmail.com).

Modala Venkata Sai Rupa Sree, is an UG student finishing her last year at Gokaraju Rangaraju Institute of Engineering and Technology, Hyderabad, India. Rupa research concentrate on Machine Learning (email: rupasreemodla@gmail.com).

Devalla Varsha is an UG student finishing her last year at Gokaraju Rangaraju Institute of Engineering and Technology, Hyderabad, India. Her research concentrate on Data Science (email: varsha.devella@gmail.com).

Kompalli Avinash Bhargav is an UG student finishing his last year at SRMIST, Chennai, India. He did internships with Code Speedy, High Radius. Currently he is doing an internship at Siemens Healthineers as a Software Engineer. His research interests include Artificial Intelligence, Machine Learning, Deep Learning (email: kompalliavinash@gmail.com).

Article submitted 2022-02-24. Resubmitted 2022-04-10. Final acceptance 2022-04-12. Final version published as submitted by the authors.