A Transfer Learning approach to Plant Disease Detection

[1]Omkar Dalvi, [2]Rummaan Ahmad, [3]Manish Gupta, [4]Rahul Parande, [5]Bhakti Sonawane, [6]Dipti Mukadam

[1234]Students, Department of Computer Engineering, Shah and Anchor Kutchhi Engineering College, Mumbai, Maharashtra, India

[56]Professor, Department of Computer Engineering, Shah and Anchor Kutchhi Engineering College, Mumbai, Maharashtra, India

***Abstract*—Plant disease detection by using different machine learning techniques is a very popular field of study. Many promising results were already obtained but there are still only a few real- life applications that can make farmer's lives easier. The aim of this work is to detect diseases that occur in Guava leaves. For this purpose, various deep learning architectures were used so a comparative analysis is made between Inception V3, Inception- Resnet V2, VGG19 and Resnet 50. As a source for training data, we used the Guava leaves dataset present on Mendeley. During research a problem arose with dataset images as the number of images were insufficient for training. To populate the dataset data augmentation was done with factor 6 which increased the dataset by 6 times per image.**

***Keywords*—Deep Learning, Data Augmentation, Resnet 50, Inception V3, Inception-Resnet V2, VGG 19, Guava Leaves, Disease Detection.**

I. Introduction

Agriculture is one of India's most significant economic industries. It is the country's development's backbone. One of the most serious issues confronting the plant diseases are being discovered in the agriculture industry. In the past, disease detection was done by trained professionals. Farmers in distant locations have a tough time contacting specialists. Plant diseases are caused by a variety of factors, including climate change. There is a substantial loss in agricultural productivity in large farms if the illness is not identified at the correct time.

Machine Learning (ML) methods, Artificial Intelligence (AI), and Digital Image Processing (DIP) approaches have all exploded in popularity in recent years. It is critical for farmers to recognize various illnesses in their crops. During the early stages of development, it is necessary to adapt these new technologies to fit into today's world. Due to the proliferation of plant diseases, agricultural output will be lowered. These diseases can alter the form of the plant, cause harm to the plant, and even kill it, and therefore impact the color and texture of the leaves and fruits. Some research has suggested several machine learning algorithms and visual processing approaches for illness detection and classification. Those diseases in plants are classified in several ways. For image processing, machine learning algorithms are created and pattern extraction, which can help with categorization and accuracy. This study uses the Convolutional Neural Networks (CNN) Algorithm to build an image processing method. To train, algorithms employ a variety of activation functions and then categorize the results. The key to minimising losses in agricultural product output and quantity is early detection of plant diseases. The study of plant diseases involves the assessing of the visually noticeable patterns on the plant. For long-term agriculture, assessing the health of plants is of utmost importance, which cannot be done manually as it takes a great deal of effort, knowledge of plant diseases, and an inordinate length of time. TO solve this issue, image processing techniques have been developed which can be utilised to detect diseases just by using an image of the plant. The detection involves a number of steps like Acquisition, Pre-processing, Segmentation, Feature extraction and Classification. It is easier and less expensive to identify illnesses automatically by just looking at the signs on the plant leaves. Machine vision is also supported, allowing for image-based automatic process control and inspection.

II. Related Work

In paper [7] the authors have proposed a CNN model implemented in the TensorFlow backend system to classify the plant diseases. The same was implemented for real time data using Raspberry Pi in OpenCV. Their dataset consisted of 100 leaf images of various leaves infected with disease. Similarly separate datasets of 100 images each are created for each disease. Features from the image are extracted using small squares of input data to learn the features and relationship among pixels through this layer. Optimized activation function is designed to improve the accuracy. For image segmentation, pooling and flattening layers along with K – means clustering algorithm is used.

In paper [8] the author used a CNN model to classify the different plant diseases obtained from the Plant Village dataset. AlexNet architecture which distinguishes the different types of diseases of the plant into 38 various unique classes. PyTorch AlexNet model is divided into two stages namely, feature extraction and classification. Initially, convolution layer extracts basic features like edges, lines, and corners. High level features are extracted by higher level layers. Hyperparameters that are used in the AlexNet architecture are SGD, base learning rate, momentum, and batch size. Classification methods used were CNN, Pre-trained AlexNet and SGD algorithm. Their dataset contained 54323 plant leaves images of 38 different categories, which includes images of common diseases of the plant.

In paper [10] this paper reviews and summarizes image processing techniques for various plant species. The author discussed various feature extraction and classification methods in detail. Feature extraction methods discussed were Texture Analysis Methods: Statistical, Structural, Fractals, Signal Pro- cessing. Textures feature extraction methods: Gray Level Co- occurrence Method (GLCM), Spatial Gray Level Dependence Matrix (SGDM), Gabor Filter, Wavelets Transform. Classification methods highlighted were K nearest neighbour (KNN), Probabilistic Neural Network (PNN), Support Vector Machine (SVM), Back Propagation Network (BPN), Radial Basis Function (RBF).

In the proposed work of paper [12], diseases of pomegranate plants are detected using digital Image Processing technique. Diseases like, Fruit Spot, Bacterial Blight, Fruit Rot and Leaf Spot are diagnosed. Their system used hidden layer output visualization where it tends to pass the image to CNN where the extraction only takes positive values to the continuous layer, the corrected linear layer is applied to Relu, that gives a rough idea about the part of the image that was important for the process. Their dataset included healthy or pathologic leaf pictures classified into 38 labels (54306 images, 26 diseases) of Pomegranate fruit. K nearest neighbour is used for image pre-processing and segmentation.

In paper [1] the framework controls the resizing method for image pre - handling. To acquire the feature, the histogram thresholding is used for segmentation followed by the element extraction which is finished by utilizing the GLCM and Radon change method. Towards the last stage, SVM order method is used to detect the leaf disease. Dataset included 70 images of Guava leaf provided by Shri Mata Vaishno Devi University, Katra, J&K, India. Key features are extracted using GLCM (Gray Level Co-occurrence Matrix), a factual strategy for surface investigation. Relevant outputs are fed to the SVM classifier, it basically makes a hyperplane of informational collections.

In [13] the author proposed an ensemble of pre-trained deep learning models namely, EfficientNetB7, DenseNet12, and EfficientNet Noisy Student to analyse each of their performances on the apple leaves dataset. Their dataset included 3642 images of apple leaves, distributed among 4 classes. Image Augmentation is used to extract various features, by using various techniques namely Blurring, Flipping, Shearing, and Convolution. Relevant outputs are fed to the SVM classifier.

In paper [9], for feature extraction networks, the Faster RCNN model uses VGG-16 to extract features from input images. In this work, ResNet-50 is selected as the feature extraction network, and then optimized the structure of ResNet-50.

In [16] The approach proposed in this paper employs three distinct models built with AlexNet as the foundational Convolutional Neural Network. The first model is based on the basic AlexNet architecture. The optimizer returns the best result for this model. Adam has a validation accuracy of 96.34 percent. AlexNet and SVM layers were combined to get a close-to-perfect result with a validation accuracy of 99.98 percent.

In paper [17], the author has proposed how much percent the fruit is affected and recognize the fruit in the given image. This feature is very useful for the farmers and useful for different purposes. To get better results in the classification and identification of fruit diseases Inception v3 model and Transfer Learning are used.

In [18], In the proposed work, the authors have used various images for detecting leaf diseases. They have used segmentation techniques like k-means clustering, for extracting various features Gray Level Co-occurrence Matrix (GLCM) is used and Support Vector Machine (SVM) classifier to classify different types of diseases.

In paper [19], the concept of transfer learning was implemented to develop an automated system for detecting different diseases in the potato leaves i.e., late and early blight and healthy leaves of potato plant.

III. PROPOSED METHODOLOGY

Application uses a pre-trained model to predict the possible disease of the guava leaf. It takes an image as input and uses the dataset used to train the model to classify the type of disease and predict the same. The pre-trained model used is of Inception-V3 from the TensorFlow package and Keras sub package. In an Inception-V3 model, various techniques for optimizing the network have been used which accounts for easier model adaptation. The techniques consist of factorized convolution, dimension reduction, regularization and parallelized computation. The architecture of the Inception-V3 model is built as:

1. Factorized Convolutions: This reduces the number of parameters in the Network and thus helps to reduce computations.
2. Smaller Convolutions: Convolutions of smaller size result in faster training of the model. For example, a 5x5 filter consisting of 25 parameters is replaced to reduce training time by replacing it by 2 3x3 filters thus having only 18 parameters.
3. Asymmetrical Convolutions: A single symmetrical layers is found to be slower than combinations of asymmetrical convolution layers.
4. Auxiliary classifier: An auxiliary classifier is a smaller CNN placed between layers during training, and the loss acquired is included in the main network loss. In Inception-V3 an auxiliary classifier act as a regularizer.
5. Grid Size Reduction: The grid size is reduced towards the end during Pooling operations. Before going through an average pooling layer, the grid size is reduced by adding another filter to increase efficiency of the average pooling layer.

IV. DATASET DISCUSSION

The Dataset required for training and testing the models have been taken from the following source:

Rauf, Hafiz Tayyab; Lali, Muhammad Ikram Ullah (2021), “A Guava Fruits and Leaves Dataset for Detection and Classification of Guava Diseases through Machine

Learning”, Mendeley Data, V1, doi: 10.17632/s8x6jn5cvr.1.

The dataset contained both healthy and diseased images of Guava fruit, leaves and stems. Diseased images were classified into 4 diseases namely - Canker, Rust, Mummification and Dot.

The total images of leaves from the above given source were 440 (After removing the fruit and stem images). Other details related to the images of the dataset are –

- File type: JPEG file
- Dimensions: 300 x 300 pixels
- Horizontal resolution: 96 dpi
- Vertical resolution: 96 dpi

TABLE I
DATASET DETAILS

Type	Quantity (Before Augmentation)	Quantity (After Augmentation)
Healthy	277	1289
Canker	42	278
Dot	36	216
Mummification	52	275
Rust	33	279

Since the dataset is of size 440, it is not ideal enough to get accurate results. Data Augmentation technique was used to augment images of every category. Images were augmented with a factor of 6 and total images after augmentation were found to be 2337 images.

The entire dataset is partitioned into training, testing and validation sets for both healthy and diseases in particular. The sets are partitioned in the ratio of 60% - 20% - 20% (.6.2.2) using the split function Fig 1 and Fig 2 show illustration of dataset before and after augmentation.

Fig. 1. Dataset before augmentation

Fig. 2. Dataset after augmentation

V. IMPLEMENTATION

This section defines the implementation details of the entire process i.e., data collection, pre-processing, feature extraction, and the model building. The model will be validated using performance evaluation metrics.

A. *Dataset Collection*

This paper uses Guava leaves dataset present on Mendeley Data associated with the paper named “A Guava Fruits and Leaves Dataset for Detection and Classification of Guava Diseases through Machine Learning” written by Rauf, Hafiz Tayyab; Lali and Muhammad Ikram Ullah. This dataset contained both healthy and diseased images of Guava fruit, leaves and stems. Diseased images were classified into 4 diseases namely - Canker, Rust, Mummification and Dot. This dataset is in an unprocessed form. The dataset needs to be pre- processed.

B. *Dataset Pre-processing*

Dataset Pre-processing is a process of preparing raw data and making it suitable for machine learning. All the training and testing images should be pre-processed before sending them to the network. The PyTorch library is used in this work which is shown in Fig.1. This library includes a transform module that implements the common transformations, including normalization, used in pre-processing. Before going to apply our data to the model, it should be resized to the input size of the network. Next, converting the dataset into tensor data type means, convert a NumPy array in the range of 0 to 255 to a float tensor in the range from 0 to 1. Finally, do all this transformation to each image in the dataset. Data augmentation is also applied to the dataset to increase the size of it and to introduce slight distortion to the images. During the training phase, this data augmentation reduces overfitting.

C. *Comparative Analysis*

The major design and implementation constraint has been the accuracy of the model trained, loss incurred by the model and thus the architecture of the CNN to be used. A detailed study of various architectures of CNN have been undertaken in order to generate the model which would not only serve the purpose but also give accurate results. The architectures trained are namely- ResNet50, Inception-V3, Inception-ResNet-V2 and VGG-19. Fig 3, Fig 4, Fig 5, Fig 6 show comparative graphs and Table 2 shows comparative table.

D. *Model Building*

In this work, an Inception V3 network is used for feature extraction as well as classification. Here the Inception V3 is trained by the augmented dataset which contains 2337 images. The structure of Inception V3 are described in section III. The software architecture is illustrated in Fig 7.

TABLE II
COMPARATIVE ANALYSIS ON VARIOUS ARCHITECTURES

Characteristic	ResNet50	Inception-V3	Inception-ResNet-V2	VGG-19
Layers	50 (48, 1, 1)	48	164	19 (16, 3, 5, 1)
Input Size	(224, 224, 3)	(224, 224, 3)	(224, 224, 3)	(224, 224, 3)
Accuracy (Epoch:10, BS:16)	81.82	93.18	90.91	88.64
Accuracy (Epoch:10, BS:32)	81.82	90.91	91.18	88.64
Accuracy (Epoch:11, BS: 16)	64.72	96.15	89.72	93.54
Accuracy (Epoch:11, BS:32)	84.09	93.18	92.45	90.91
Accuracy (Epoch:25, BS:32)	79.55	94.11	93.18	90.91
Output Nodes	5	5	5	5

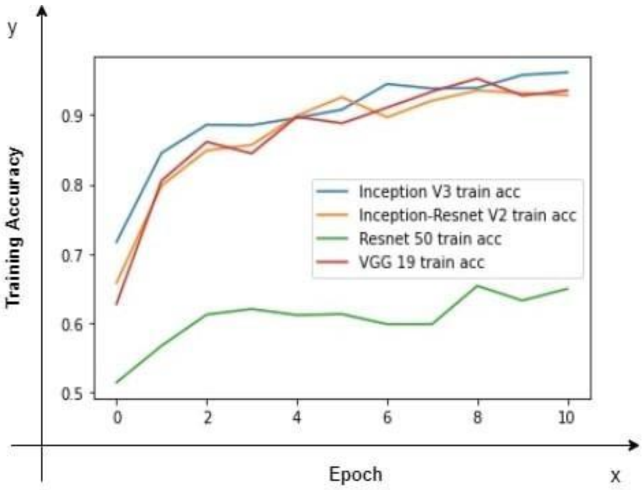

Fig. 3. Training accuracy graph

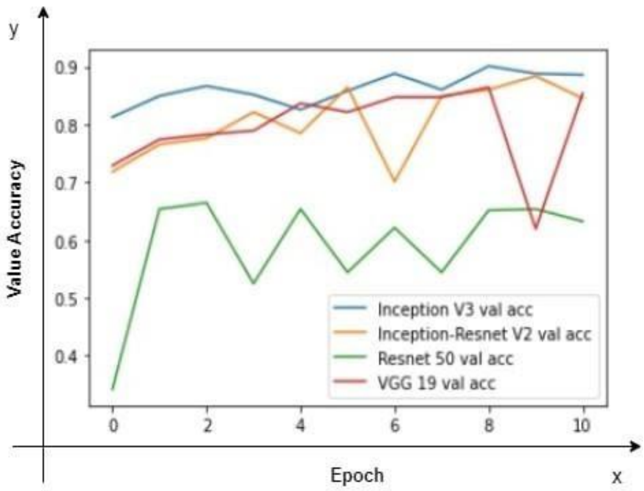

Fig. 5. Value accuracy graph

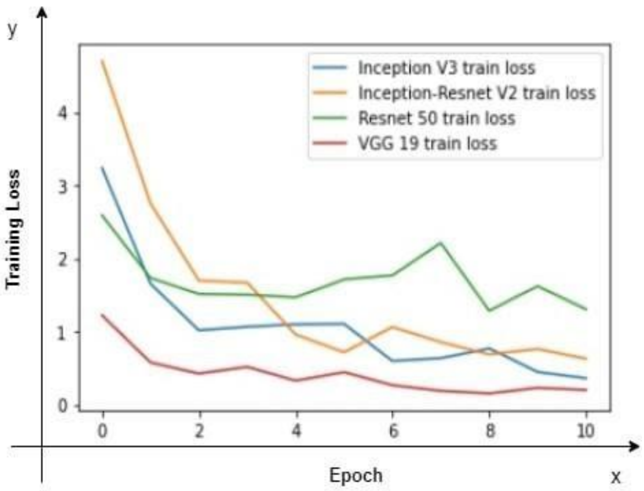

Fig. 4. Training loss graph

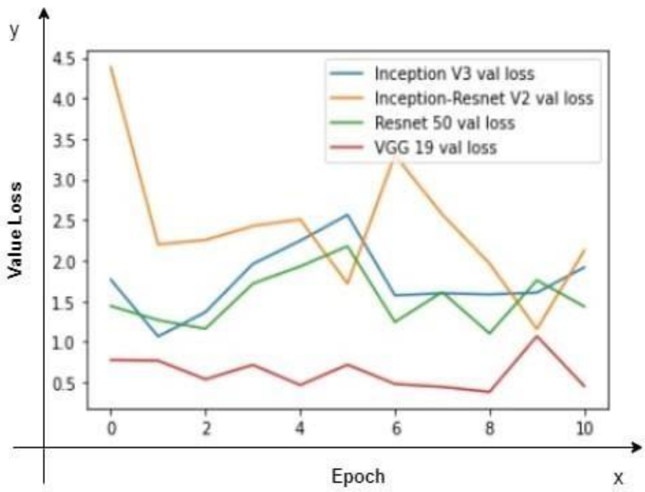

Fig. 6. Value loss graph

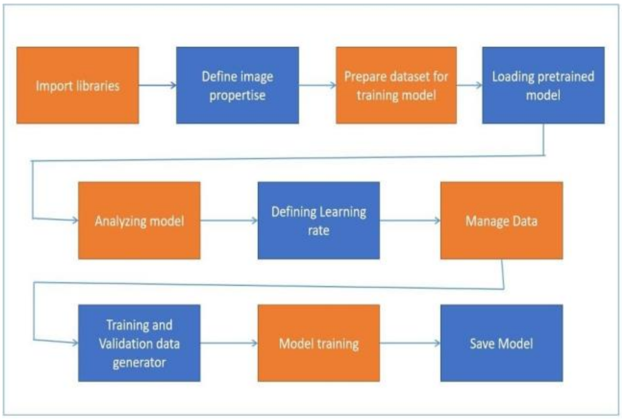

Fig. 7. Software Architecture Flowchart

E. *Feature Extraction*

For extracting the features of leaves from the dataset, transfer learning is used. A pre-trained model is used to classify leaves in various classes. The last layer of pre-trained model is eliminated and added a last layer which classify the leaves. The model consists of five output nodes.

Transfer learning is basically, retraining the final layer of a deep network. Not only does it help when you don't have adequate computing resources to train a network from scratch but also for solving problems with limited training examples.

Some models have hundreds of millions of parameters, which might take weeks to train on modest equipment. However, adapting weights via transfer learning is not preferable if you have sufficient data, because the features that were extracted from the original training process are not likely to be suitable for another dataset.

Feature extraction refers to the portion of the training process using which a CNN learns to map input space to a latent space. This is later used for classification via the final layer.

In other words, the hidden layers learn discriminatory features in the form of weight-adjusted (usually by back propagating the error) convolutional filters. Thus, feature extraction is used to define the portion of the training process that occurs before the final layer. By performing deep learning feature extraction, we consider the pre-trained network (i.e., Inception V3) as an arbitrary feature extractor, allowing the input image to propagate forward, stopping at the pre-specified layer, and taking the outputs of that layer as our features.

By doing so, we can still utilize the robust, discriminative features learned by CNN. We have eliminated the last layer of the pre trained Model and added our own layer which does the classification on leaves in different classes.

F. *Classification*

After completion of the training process, the model is ready for the classification of any unlabelled images of plants. The model takes the image as an input and the comparison is done between the training and testing images and predicts whether the leaf is diseased or healthy as the output along with the disease type.

VI. Test Results and Discussion

Based on a comparison of multiple CNN architectures, Inception v3 had the best accuracy for the Guava dataset, as it employs a pre-trained model to detect the possible leaf disease. It takes an image as input and uses the dataset used to train the model to classify the type of disease and predict the same. The pre-trained model used is of Inception-V3 from the TensorFlow package and Keras sub package.

The batch size is set to 16 and 11 was the number of epochs. 60% of images from the augmented Guava leaves dataset were used to train the accuracy of this model. In every class, 20% of the images were selected for testing and 20% of the images were selected for validation. The testing dataset gives more than 95% accuracy. It means if 100 images are inputted then 95 images were classified correctly.

The accuracy and the loss for both training and validation graphs generated by the model are shown below. When the training dataset is increased and epoch the accuracy is also increased. At the 9th epoch, the model gives the highest accuracy of 95.01%. Fig 8 and 9 illustrates graphs of training accuracy and validation accuracy, and training losses and validation losses respectively.

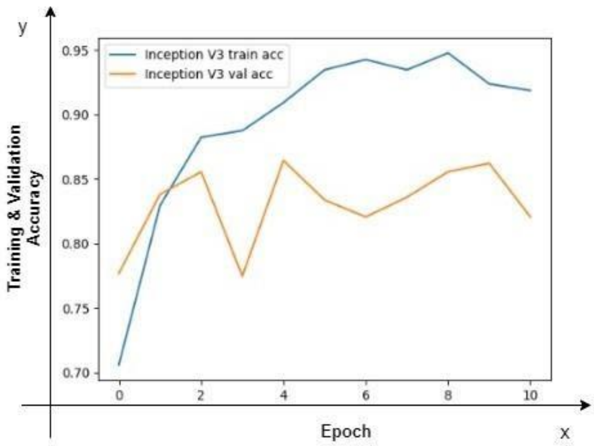

Fig. 8. Training and validation accuracy

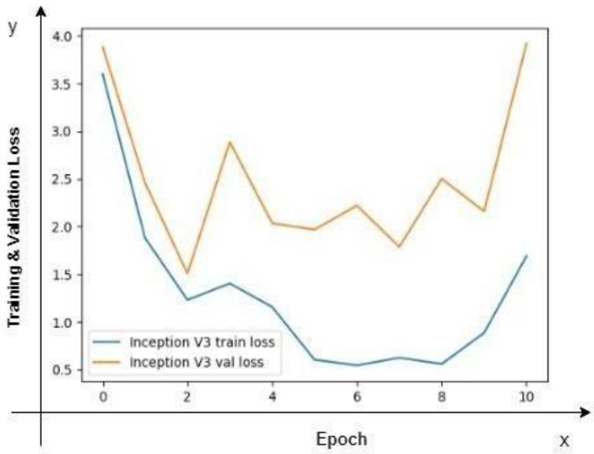

Fig. 9. Training and validation loss

The tkinter gui application framework provides different tools to create a gui application in python. The created gui application by using tkinter shown below.

This application contains two buttons such as upload and classify buttons. The images can be downloaded from the web or the images can be captured by phone camera or normal camera and uploaded on the gui. The color images are considered and the images should be in the .jpg format. After uploading the images, it will display the predictions for the uploaded image on the same window.

It shows the image along with whether the leaf is diseased or healthy, if diseased it will display the disease name. Fig 10 onwards show the application screenshots.

Fig. 10. First Screen

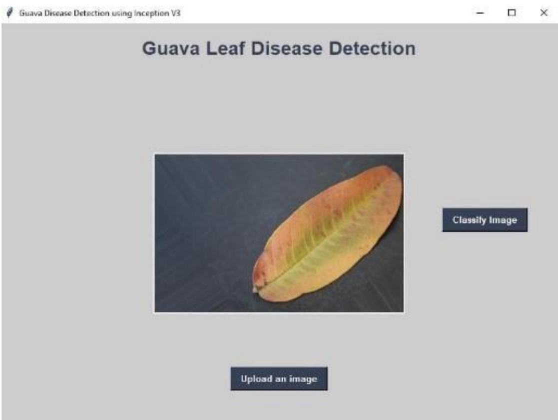

Fig. 11. Second Screen

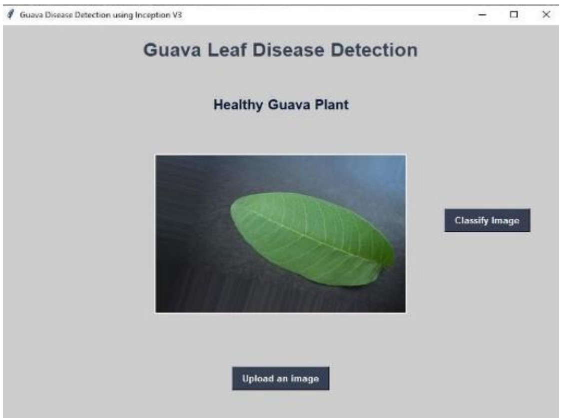

Fig. 12. After Classification Screen (Healthy)

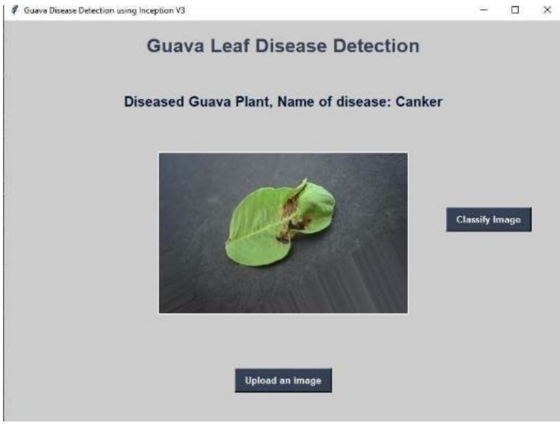

Fig. 13. After Classification Screen (Canker)

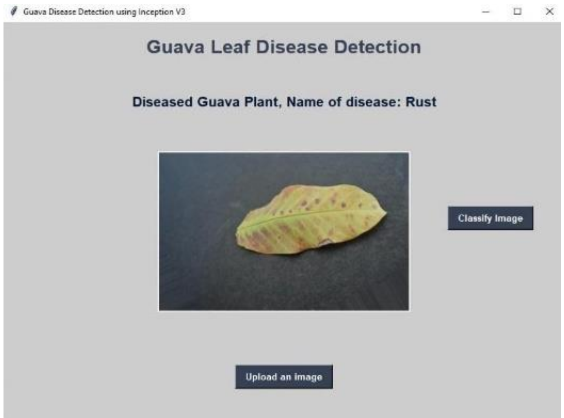

Fig. 14. After Classification Screen (Rust)

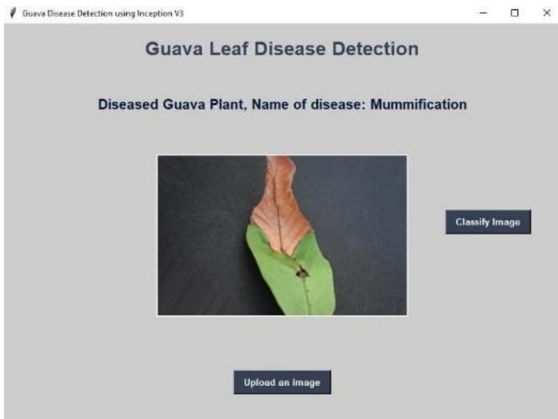

Fig. 15. After Classification Screen (Mummification)

Fig. 16. After Classification Screen (Dot)

VII. CONCLUSION

Agriculture is one of the most important sectors in the Indian economy. To have a raise in the economy of any country, there must be a prediction of diseases in the crops is very much important. The proposed method uses a CNN model to classify the different plant diseases obtained from Guava leaves a dataset. The Inception V3 architecture which will distinguish the different types of diseases of the plant into 4 unique classes. Also, our proposed system gives a good solution to predict the diseases in the plant and can help in early identification of them. In the future, it is possible to work on different learning rates on our proposed system.

ACKNOWLEDGMENT

First and foremost, we would like to express our sincere gratitude towards the faculty of ‘Shah & Anchor Kutchhi Engineering College’, Mumbai for the encouragement and support that helped us achieve what we aimed for. And a greatest thanks to our entire team involved in this project.

We would like to extend our deepest thanks to our project guides, Mrs Dipti Mukadam and Mrs Bhakti Sonawane for providing us with their valuable support and contribution to this project. Their consistent support and Co - operation showed the way towards the successful completion of the project.

REFERENCES

[1] Rauf, Hafiz Tayyab; Lali, Muhammad Ikram Ullah (2021), “A Guava Fruits and Leaves Dataset for Detection and Classification of Guava Diseases through Machine Learning”, Mendeley Data, V1, doi: 10.17632/s8x6jn5cvr.1

[2] P. Revathi and M. Hemalatha, ”Classification of cotton leaf spot diseases using image processing edge detection techniques,” 2012 International Conference on Emerging Trends in Science, Engineering and Technology (INCOSET), 2012, pp. 169-173, doi: 10.1109/INCOSET.2012.6513900.

[3] Mrs.Disha Sushant Wankhede, Dr. Selvarani Rangasamy, December 22, 2020, ”Leaves: India’s Most Famous Basil Plant Leaves Quality Dataset”,

[4] M. Sardogan, A. Tuncer and Y. Ozen, ”Plant Leaf Disease Detection and Classification Based on CNN with LVQ Algorithm,” 2018 3rd International Conference on Computer Science and Engineering (UBMK), 2018, pp. 382-385, doi: 10.1109/UBMK.2018.8566635

[5] Shantanu Kumbhar, Amita Nilawar, Shruti Patil, Bodireddy Mahalakshmi, Manasi Nipane, “Farmer Buddy-Web Based Cotton Leaf Disease Detec- tion Using CNN” International Journal of Applied Engineering Research ISSN 0973-4562 Volume 14, Number 11 (2019) pp. 2662-2666.

[6] Tanmay A. Wagh, R. M. Samant, Sharvil V. Gujarathi, Snehal B. Gaikwad, “Grapes Leaf Disease Detection using Convolutional Neural Network” International Journal of Computer Applications (0975 – 8887) Volume 178 – No. 20, June 2019.

[7] S.Yagneshwar Yadhav, T.Senthilkumar, S.Jayanthy, J.Judeson Antony Kovilpillai, “Plant Disease Detection and Classification using CNN Model with Optimized Activation Function” Proceedings of the International Conference on Electronics and Sustainable Communication Systems (ICESC 2020), IEEE Xplore Part Number: CFP20V66-ART; ISBN: 978- 1-7281-4108-4.

[8] G. Madhulatha, O. Ramadevi, “Recognition of Plant Diseases using Convolutional Neural Network” Proceedings of the Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I- SMAC) IEEE Xplore Part Number: CFP20OSV-ART; ISBN: 978-1-7281- 5464-0.

[9] Keke Zhang, Qiufeng Wu, Yiping Chen, “Detecting soybean leaf disease from synthetic image using multi-feature fusion faster R-CNN”, Comput- ers and Electronics in Agriculture 183 (2021) 106064.

[10] Gavhale, Ms & Gawande, Ujwalla. (2014). An Overview of the Re- search on Plant Leaves Disease detection using Image Processing Tech- niques. IOSR Journal of Computer Engineering. 16. 10-16. 10.9790/0661- 16151016.

[11] S. S. Chouhan, U. P. Singh, A. Kaul and S. Jain, ”A Data Repos- itory of Leaf Images: Practice towards Plant Conservation with Plant Pathology,” 2019 4th International Conference on Information Systems and Computer Networks (ISCON), 2019, pp. 700-707, doi: 10.1109/IS- CON47742.2019.9036158.

[12] Vasumathi MT, Kamarasan M. (2021) An Effective Pomegranate Fruit Classification Based On CNN-LSTM Deep Learning Mod- els. Indian Journal of Science and Technology. 14(16): 1310-1319. https://doi.org/10.17485/IJST/v14i16.432

[13] P. Bansal, R. Kumar, and S. Kumar, “Disease Detection in Apple Leaves Using Deep Convolutional Neural Network,” Agriculture, vol. 11, no. 7, p. 617, Jun. 2021

[14] V. Feng, “An overview of resnet and its variants,” *Medium*, 17-Jul-2017. [Online]. Available: https://towardsdatascience.com/an-overview-of-resnet-and-its-variants-5281e2f56035. [Accessed: 24-Aug-2021].

[15] “freecodecamp.org,” *Certification | freeCodeCamp.org*. [Online]. Available: https://www.freecodecamp.org/learn/machine-learning-with-python/#tensorflow. [Accessed: 24-Aug-2021].

[16] M. Kawatra, S. Agarwal and R. Kapur, "Leaf Disease Detection using Neural Network Hybrid Models," 2020 IEEE 5th International Conference on Computing Communication and Automation (ICCCA), 2020, pp. 225-230, doi: 10.1109/ICCCA49541.2020.9250885.

[17] M. Nikhitha, S. Roopa Sri and B. Uma Maheswari, "Fruit Recognition and Grade of Disease Detection using Inception V3 Model," 2019 3rd International conference on Electronics, Communication and

Aerospace Technology (ICECA), 2019, pp. 1040-1043, doi: 10.1109/ICECA.2019.8822095.

[18] S. Kumar, K. Prasad, A. Srilekha, T. Suman, B. P. Rao and J. N. Vamshi Krishna, "Leaf Disease Detection and Classification based on Machine Learning," 2020 International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE), 2020, pp. 361-365, doi: 10.1109/ICSTCEE49637.2020.9277379.

[19] D. Tiwari, M. Ashish, N. Gangwar, A. Sharma, S. Patel and S. Bhardwaj, "Potato Leaf Diseases Detection Using Deep Learning," 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS), 2020, pp. 461-466, doi: 10.1109/ICICCS48265.2020.9121067.