Volume 10, Issue 5
October 2021

ITEE Journal
Information Technology & Electrical Engineering

ISSN: - 2306-708X

Plant Diseases Detection and Classification based on Leaf Images using Deep Learning

Manojkumar B. Patel

Department of Information Technology, L. D. College of Engineering, Gujarat, India
E-mail: manojpatel@ldce.ac.in

ABSTRACT

Every country's fundamental need is for agricultural products. Infected plants have a negative influence on the country's agricultural productivity and economic resources. This paper presents an intelligent system that is used to detect and classify plant leaf diseases using deep learning techniques. The PlantVillage dataset, which contains 38 different classes, is used. This dataset contains 54,305 images of plants' leaves and their diseases. In this research work, three pre-trained CNN models (MobileNet, VGG16, and Inception V3) are used to classify plant leaf diseases into 38 different classes. As a result, I obtained excellent accuracy during the training phase and testing phase. I have achieved an accuracy of 93.30% for the proposed VGG16 model, 99.51% for the proposed MobileNet model, and 89.31% for the proposed InceptionV3 model during training. During testing using test data, the accuracy of modals was found to be 94% for the proposed VGG16 model, 99% for the proposed MobileNet model, and 91% for the proposed InceptionV3 model.

Keywords: *Plant Diseases, Deep Learning, CNN (Convolution Neural Network).*

1. INTRODUCTION

Agriculture is the foundation of all human civilizations. Agriculture is the primary source of food, raw materials, and fuel, all of which contribute to a country's economic prosperity. The focus is on increasing production without taking into account the environmental consequences that have resulted in environmental degradation. Plant diseases are extremely significant since they can affect both the quality and quantity of plants in agricultural growth. Plant diseases are caused by fungus, bacteria, viruses, moulds, and other microorganisms. Farmers or specialists are able to recognize various plant diseases with naked eyes. But this approach can be expensive, time-consuming, and incorrect. Therefore, deep learning based methods are used for detection and classification of plant diseases. The images of plant diseases are used for this research.

The PlantVillage dataset, which contains 54,305 images of 14 crop species with 26 diseases, is used for this work. To detect and classify thirty-eight different classes of plant diseases, I propose an intelligent system based on deep learning algorithms with transfer learning. VGG16, MobileNet, and InceptionV3 are three pre-trained deep learning architectures that I chose.

The following is the order of the rest of the paper: In section 2, a literature review is presented. The dataset used in this research work is explained in section 3. Intelligent expert system methodology is explained in Section 4. In section 5, the experiment and results are discussed. Conclusions and future work are presented in section 6.

2. RELATED WORK

In the literature, various image processing and deep learning approaches used to classify numerous plant diseases are discussed. S. S. Sannakki and V. S. Rajpurohit [1] presented work that focuses on the way of segmenting the defective region and using color and texture as characteristics. For the categorization, they employed a neural network classifier. The key benefit is that it converts to L*a*b in order to extract the image's chromaticity layers, and categorization is determined to be 97.30 percent correct. The biggest disadvantage is that it can only be used for a few harvests. P. R. Rothe and R. V. Kshirsagar [2] developed "Cotton Leaf Disease Identification Using Pattern Recognition Techniques," which used snake segmentation and Hu's moments as a distinguishing characteristic. The BPNN classifier addresses the various class difficulties by using an active contour model to restrict the vitality inside the infection area. It was discovered that the average categorization was 85.52 percent. Lee et al. [3] proposed a hybrid model to obtain features of a leaf using CNN and classify the extracted features of the leaf. Durmus et al. [4] used AlexNet and SqueezeNet pre-trained CNN architectures for the detection of diseases of tomato leaves. K.P. Ferentinos, [5] developed a CNN model for the detection and diagnosis of plant disease using simple leaf images of healthy and diseased plants. The final model achieved 99.53% accuracy. Prajwala TM, et al. [6] created a system to identify and classify diseases in tomato leaves using a variant of the CNN architecture known as LeNet. This system has a 94-95 percent overall accuracy rating. Omkar Kulkarni [7] used a transfer learning approach to build the CNN model using InceptionV3 and MobileNet pre-trained models. These models are implemented by using five different types of crops from the PlantVillage dataset. Sammy V. Militante ,et. al. [8] Designed a system to detect and recognize different plant varieties specifically potato, sugarcane, tomato, apple, Corn and grapes. The system can also detect several plant diseases. Marwan Adnan Jasim and Jamal Mustafa AL-Tuwaijari [9] developed a system to detect and classify plant leaf diseases using deep learning techniques. The system can

Volume 10, Issue 5
October 2021

ITEE Journal
Information Technology & Electrical Engineering

ISSN: - 2306-708X

classify plant diseases into 15 different classes of the PlantVillage dataset.

An intelligent system that can perform multi-class categorization of a variety of plant diseases is required. According to recent advances in computational deep learning, CNN-based approaches appear to be a promising strategy for the categorization of plant diseases. I use the concept of transfer learning with CNN models.

3. DATASET

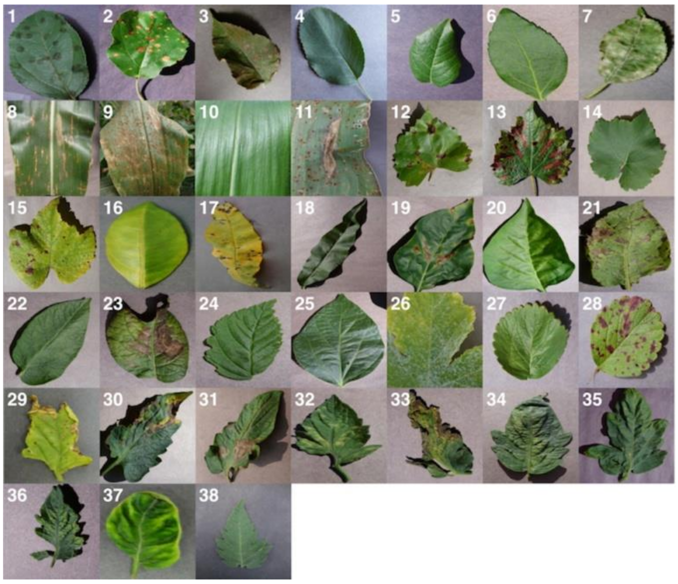

Fig.1. An example of leaf images from the PlantVillage dataset

Here, the PlantVillage Dataset is used, which contains 54,305 images of 14 crop species with 26 diseases. We divide the dataset into three parts: training (80%), validation (15%), and testing (05%). The details of each class are given in Table 1.

Sr. No	Class	Total	Training (80%)	Validation (15%)	Testing (05%)
1	Apple_Apple_scab	630	510	94	26
2	Apple_Black_rot	621	502	93	26
3	Apple_Cedar_apple_rust	275	223	41	11
4	Apple_healthy	1645	1330	246	69
5	Blueberry_healthy	1502	1214	225	63
6	Cherry_(including_sour)_healthy	854	690	128	36
7	Cherry_(including_sour)_Powdery_mildew	1052	851	157	44
8	Corn_(maize)_Cercospora_leaf_spot Gray_leaf_spot	513	416	76	21
9	Corn_(maize)_Common_rust_	1192	964	178	50
10	Corn_(maize)_healthy	1162	939	174	49
11	Corn_(maize)_Northern_Leaf_Blight	985	797	147	41
12	Grape_Black_rot	1180	953	177	50
13	Grape_Esca_(Black_Measles)	1383	1118	207	58
14	Grape_healthy	423	342	63	18
15	Grape_Leaf_blight_(Isariopsis_Leaf_Spot)	1076	870	161	45
16	Orange_Haunglongbing_(Citrus_greening)	5507	4447	826	234
17	Peach_Bacterial_spot	2297	1856	344	97
18	Peach_healthy	360	291	54	15
19	Pepper_bell_Bacterial_spot	997	806	149	42
20	Pepper_bell_healthy	1478	1195	221	62
21	Potato_Early_blight	1000	808	150	42
22	Potato_healthy	152	124	22	6
23	Potato_Late_blight	1000	808	150	42
24	Raspberry_healthy	371	301	55	15
25	Soybean_healthy	5090	4111	763	216
26	Squash_Powdery_mildew	1835	1482	275	78
27	Strawberry_healthy	456	369	68	19
28	Strawberry_Leaf_scorch	1109	896	166	47
29	Tomato_Bacterial_spot	2127	1718	319	90
30	Tomato_Early_blight	1000	808	150	42
31	Tomato_healthy	1591	1286	238	67
32	Tomato_Late_blight	1909	1542	286	81
33	Tomato_Leaf_Mold	952	770	142	40
34	Tomato_Septoria_leaf_spot	1771	1431	265	75
35	Tomato_Spider_mitesTwo-spotted_spider_mite	1676	1354	251	71
36	Tomato_Target_Spot	1404	1135	210	59
37	Tomato_Tomato_mosaic_virus	373	303	55	15
38	Tomato_Tomato_Yellow_Leaf_Curl_Virus	5357	4327	803	227
	Total	54305	43887	8129	2289

Table-1. Dataset Distribution

- **Data Pre-Processing:** The leaf image in the PlantVillage dataset has a size of 256x256 pixels and

ITEE Journal
Information Technology & Electrical Engineering
ISSN: - 2306-708X

RGB values in the range of 0 to 255. The original image is then resized into three different sizes for three different models. The input image size for the VGG16 model, the MobileNet model, and the Inceptionv3 model is 224*224, 224*224, and 150*150, respectively. To normalise RGB values, each pixel's RGB value is divided by 255 to rescale its value from 0 to 1.

- **Data Augmentation:** The deep learning model requires a large amount of data for training to produce good results. To increase the size of the training data, the data augmentation process is required. The number of images for training is 43887, for validation it is 8129, and for testing it is 2289. The data augmentation is only applied to training data. Many geometrical transformations are applied to the image of the training data in the data augmentation process. We employ shear_range, zoom_range, width_shift_range, height_shift_range, and fill_mode to transform the images. The ImageDataGenerator function is used for all these transformations.

4. METHODOLOGY

The methodology of our proposed intelligent system is presented in Fig.2.

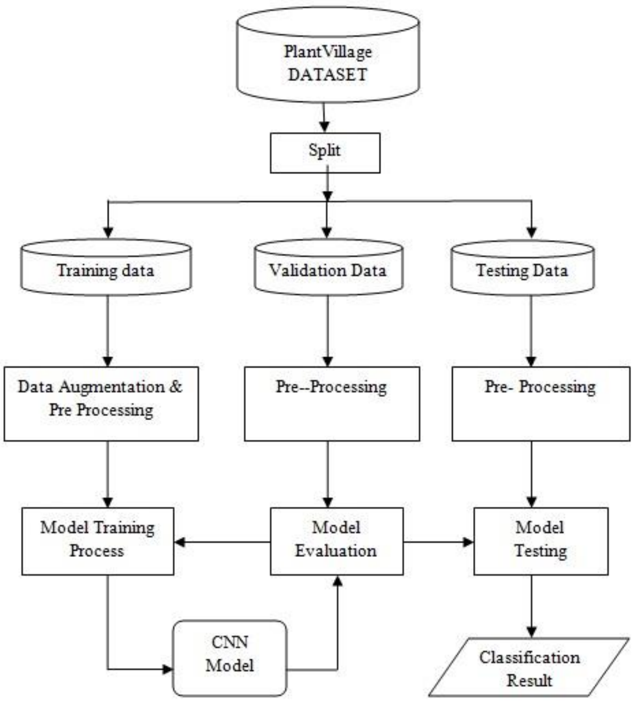

Fig. 2. Methodology of proposed intelligent system

There are three phases to the proposed system: training, validation, and testing. In the training phase, the model needs to be trained using training data that has passed through the data preprocessing and augmentation process. For training, values for batch size, train steps, and epoch are required. For all the pre-trained CNN models used in this research, the training dataset has 43887 images. So the values of batch size, train steps, and epoch are as mentioned below.

- **Batch Size:** The batch size is a hyperparameter that defines the number of samples to work through before updating the internal parameters of the CNN model. For this system, the batch size value is 128.
- **Train step**: The Train Step is defined as the total number of samples in the training data divided by batch size. For this system, the train step value is 343.
- **Epoch:** The Epoch is a hyperparameter that defines the number of times the CNN model will work through the entire training data. For this system, the epoch value is 10.

I must import all three models from the Keras API after determining batch size, epoch, and training steps. This research employed a transfer learning strategy and three pre-trained CNN models. The transfer learning method is a deep learning method in which a previously trained model is utilised as the basis for a new model on a similar problem. For feature extraction from images, the pre-trained CNN models VGG16, MobileNet, and Inception V3 are employed. For feature extraction and classification, we made certain changes to the architecture of these models.

The VGG16 model has a total of 23 layers. We removed the last four layers of the VGG16 model that worked as part of the classification process. Next, we freeze the remaining 19 layers so that the weight does not change during the workout. Then we added a flatten layer and a dense layer with the RELU activation function. Then two new layers were added: the dropout layer and the dense layer. A dropout layer has been added to reduce the overfitting problem. A dense layer is added with 38 output classes and an activation function set to sigmoid. The dense layer is processed as a fully connected layer. We applied a fine-tuning process by removing layers and adding them to a model for classification. To compile the model, we used the Adam optimizer with a learning rate set to 0.0001 and categorical_crossentropy as the loss function.

The number of layers in the pre-trained MobileNet model is 93. We have removed the last five layers from the MobileNet model. Then two new layers were added: the dropout layer and the dense layer. The dense layer activation function is set to the soft max function. Next, we freeze all the layers of the model except the last 23 layers. To compile the model, we use the Adam optimizer with a learning rate of 0.0001 and categorial_crossentropy as a loss function.

There are 313 layers in the InceptionV3 model architecture. By setting the include top parameter to false while loading the model, we were able to remove InceptionV3's fully connected output layer. The model's layers are then frozen. Then we added a dense layer and a flatten layer. We set the activation function of the dense layer to RELU. Then, two new layers, a dropout layer and a dense layer, are added to the model. The dense layer output parameter is set to 38 class and sigmoid as the activation function.

Volume 10, Issue 5
October 2021

ITEE Journal
Information Technology & Electrical Engineering

ISSN: - 2306-708X

5. EXPERIMENTS AND RESULTS

Our main goal in this study is to create an intelligent system for plant disease classification based on a deep learning model. I have implemented all these models in Python using a Jupyter notebook. The model is evaluated and validated using the cross-validation approach. I have passed training data as well as validation data to the model at the time of training. I plotted the learning curve of our model to check its learning process. The learning curve can be used to diagnose underfit, overfit, or well-fit problems in the model. I have achieved an accuracy of 93.30% for the proposed VGG16 model, 99.51% for the proposed MobileNet model, and 89.31% for the proposed InceptionV3 model during training. The graph in Fig. 3 shows the comparison of the accuracy values of the proposed VGG16 model, the proposed MobileNet model, and the proposed InceptionV3 model during the training process.

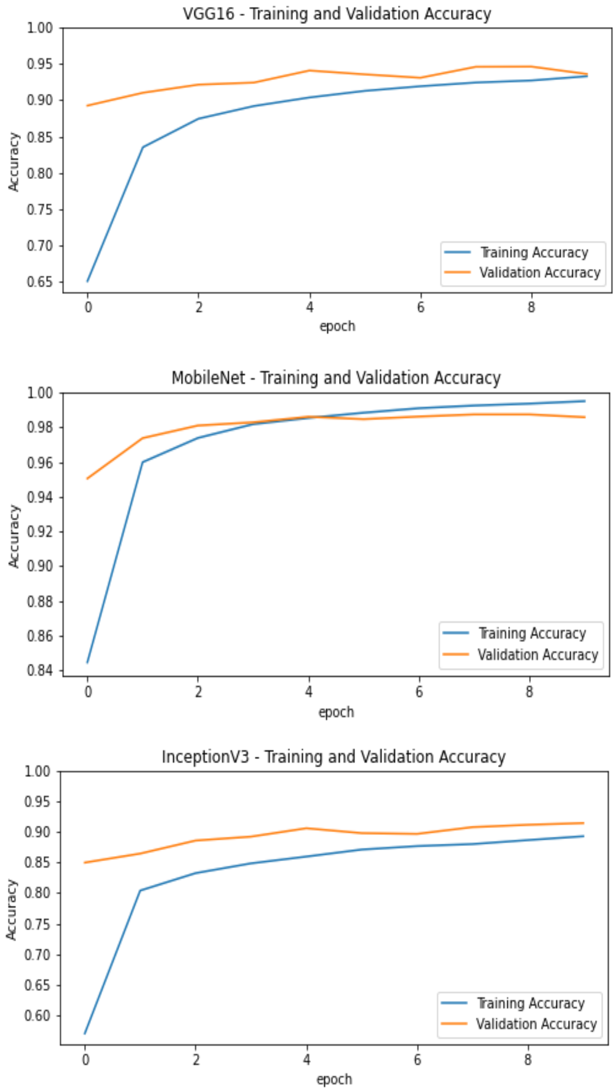

Fig. 3. Comparison of accuracy values of the VGG16 model, the MobileNet model, and the InceptionV3 model

The graph in Fig. 4 shows the comparison of the loss values of the VGG16 model, the MobileNet model, and the InceptionV3 model during the training process. VGG16, MobileNet, and Inception V3 do not have underfitting or overfitting concerns because the value of loss from all models is reduced throughout training.

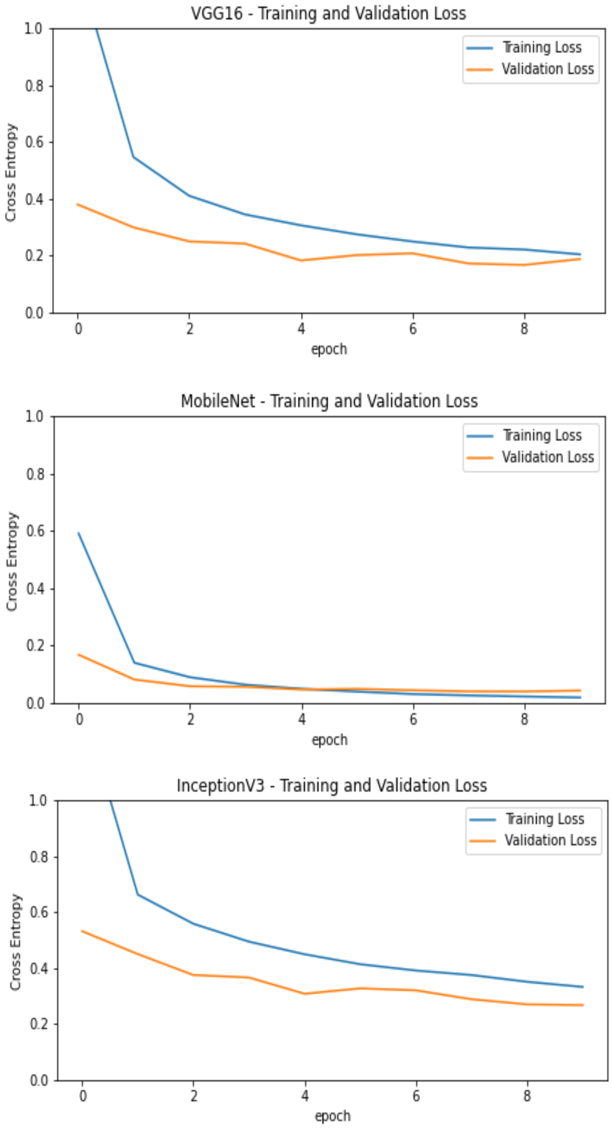

Fig. 4. Comparison of loss values of the VGG16 model, the MobileNet model, and the InceptionV3 model

- **Recall, Precision, and F-1 Score:** The PlantVillage is a dataset with a lot of class imbalance in it. As a result, for each model, I provided a recall, precision, and F1 score to evaluate the proposed architectures. The Testing dataset is used to evaluate our proposed architectures. The recall, precision, f1-score, and support of proposed models are shown in different table numbers 2, 3, and 4.

$$precision = true\ positivetotal/predicted\ positive$$
$$recall = true\ positivetotal/actual\ positive$$
$$f1\ score = 2*((precision*recall)/(precision+recall))$$

Plant Disease Class	precision	recall	f1-score	support

ITEE Journal
Information Technology & Electrical Engineering

ISSN: - 2306-708X

1	0.96	0.88	0.92	26
2	1	0.92	0.96	26
3	1	0.91	0.95	11
4	0.91	1	0.95	69
5	0.98	0.98	0.98	63
6	0.98	0.98	0.98	44
7	1	1	1	36
8	0.78	0.67	0.72	21
9	0.98	1	0.99	50
10	0.84	0.88	0.86	41
11	1	1	1	49
12	1	0.84	0.91	50
13	0.88	1	0.94	58
14	1	0.96	0.98	45
15	1	1	1	18
16	1	0.99	0.99	234
17	1	0.97	0.98	97
18	0.88	0.93	0.9	15
19	0.95	0.95	0.95	42
20	0.94	1	0.97	62
21	0.97	0.81	0.88	42
22	0.82	0.95	0.88	42
23	0.71	0.83	0.77	6
24	0.94	1	0.97	15
25	1	0.99	0.99	216
26	1	1	1	78
27	0.98	0.98	0.98	47
28	1	1	1	19
29	0.9	0.94	0.92	90
30	0.87	0.48	0.62	42
31	0.93	0.8	0.86	81
32	0.89	0.82	0.86	40
33	0.9	0.8	0.85	75
34	0.65	0.99	0.79	71
35	0.74	0.83	0.78	59
36	0.97	0.96	0.97	227
37	0.93	0.87	0.9	15
38	1	0.93	0.96	67
accuracy			0.94	2289
macro avg	0.93	0.92	0.92	2289
weighted avg	0.94	0.94	0.94	2289

Table-2. Classification Report of Proposed VGG16 Model

Plant Disease Class	**precision**	**recall**	**f1-score**	**suppor t**
1	1	0.96	0.98	26
2	1	1	1	26
3	1	1	1	11
4	0.99	1	0.99	69
5	1	1	1	63
6	1	1	1	44
7	1	1	1	36
8	0.87	0.95	0.91	21
9	1	1	1	50
10	0.97	0.93	0.95	41
11	1	1	1	49
12	1	1	1	50
13	1	1	1	58
14	1	1	1	45
15	1	1	1	18
16	1	1	1	234
17	1	1	1	97
18	1	1	1	15
19	1	1	1	42
20	1	1	1	62
21	1	1	1	42
22	0.98	1	0.99	42
23	1	0.83	0.91	6
24	1	1	1	15
25	1	1	1	216
26	1	1	1	78
27	1	1	1	47
28	1	1	1	19
29	1	0.98	0.99	90
30	1	0.88	0.94	42
31	0.98	1	0.99	81
32	1	0.95	0.97	40
33	0.99	1	0.99	75
34	0.99	0.97	0.98	71
35	0.88	0.98	0.93	59
36	1	1	1	227
37	1	1	1	15
38	1	1	1	67
accuracy			0.99	2289
Macro avg	0.99	0.99	0.99	2289
weighted avg	0.99	0.99	0.99	2289

Table-3. Classification Report of Proposed MobileNet Model

Plant Disease Class	**precision**	**recall**	**f1-score**	**support**
1	0.78	0.81	0.79	26
2	0.86	0.96	0.91	26
3	0.91	0.91	0.91	11
4	0.93	0.93	0.93	69
5	0.92	0.94	0.93	63
6	0.97	0.89	0.93	44
7	0.95	0.97	0.96	36
8	0.69	0.86	0.77	21
9	0.98	1	0.99	50
10	0.89	0.76	0.82	41
11	1	0.98	0.99	49
12	0.91	0.96	0.93	50
13	0.97	0.97	0.97	58
14	0.95	0.91	0.93	45
15	0.89	0.94	0.92	18

Volume 10, Issue 5
October 2021

ITEE Journal
Information Technology & Electrical Engineering

ISSN: - 2306-708X

Plant Disease Class	precision	recall	f1-score	support
16	0.96	1	0.98	234
17	0.9	0.99	0.94	97
18	1	0.73	0.85	15
19	0.89	0.81	0.85	42
20	0.78	0.95	0.86	62
21	0.82	1	0.9	42
22	0.92	0.81	0.86	42
23	1	0.5	0.67	6
24	1	0.93	0.97	15
25	0.98	0.97	0.97	216
26	0.99	0.96	0.97	78
27	1	0.91	0.96	47
28	0.94	0.84	0.89	19
29	0.84	0.88	0.86	90
30	0.67	0.67	0.67	42
31	0.86	0.84	0.85	81
32	1	0.5	0.67	40
33	0.76	0.79	0.77	75
34	0.86	0.85	0.85	71
35	0.73	0.69	0.71	59

Plant Disease Class	precision	recall	f1-score	support
36	0.95	0.96	0.95	227
37	0.77	0.67	0.71	15
38	0.95	0.94	0.95	67
accuracy			0.91	2289
macro avg	0.90	0.87	0.88	2289
weighted avg	0.91	0.91	0.91	2289

Table-4.Classification Report of Proposed InceptionV3 Model

- **Confusion Matrix:** Confusion matrix needs to acquire a clear understanding of our proposed models because of the issue of class imbalance. This enables us to determine where our models may be flawed, as well as the confusion matrix used to sketch the architecture's performance. Testing Dataset used to evaluate our proposed models. In Fig. 5,6,and 7, three confusion matrices for three proposed models are shown.

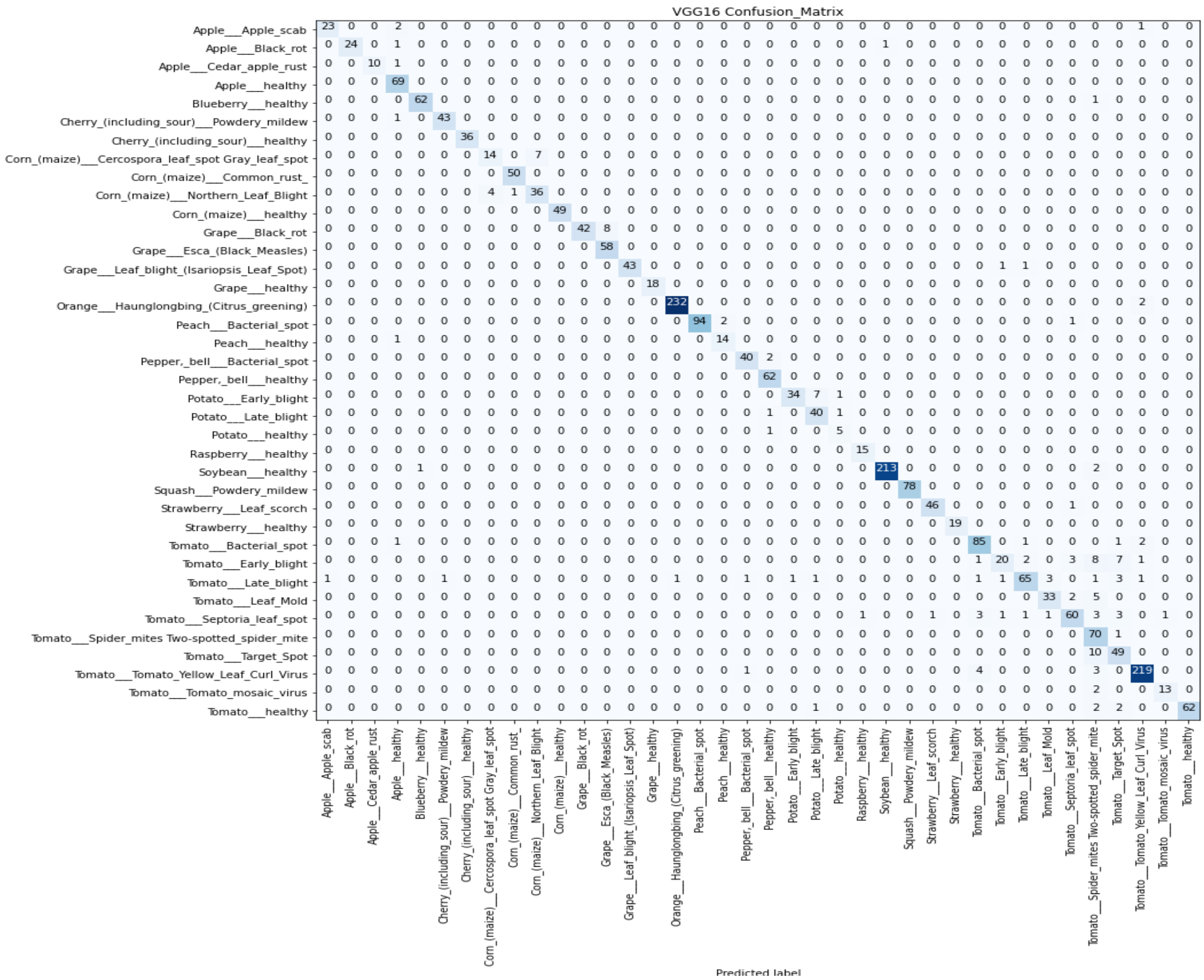

Fig. 5. VGG16 Confusion Matrix

ITEE Journal
Information Technology & Electrical Engineering

ISSN: - 2306-708X

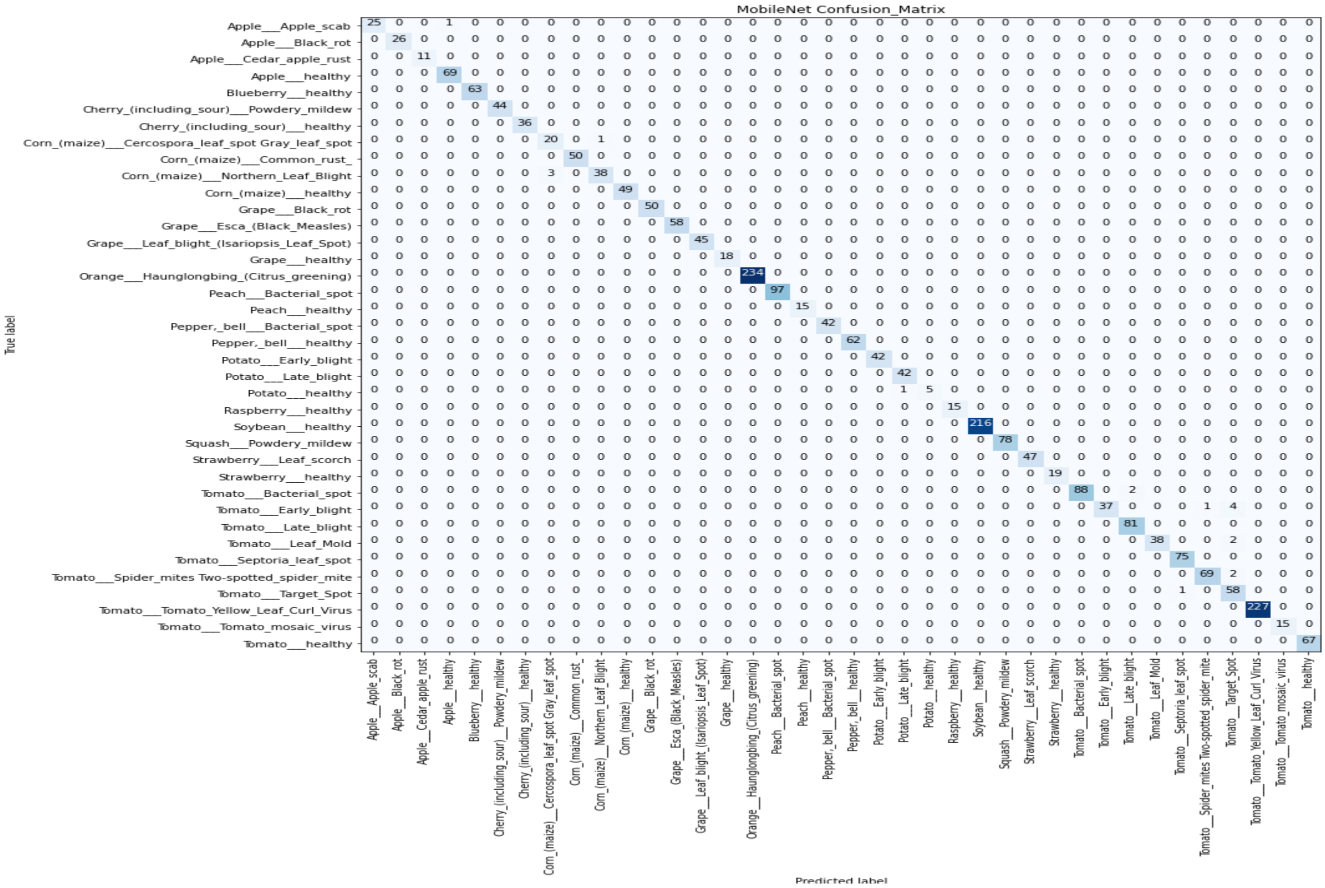

Fig. 6. MobileNet Confusion Matrix

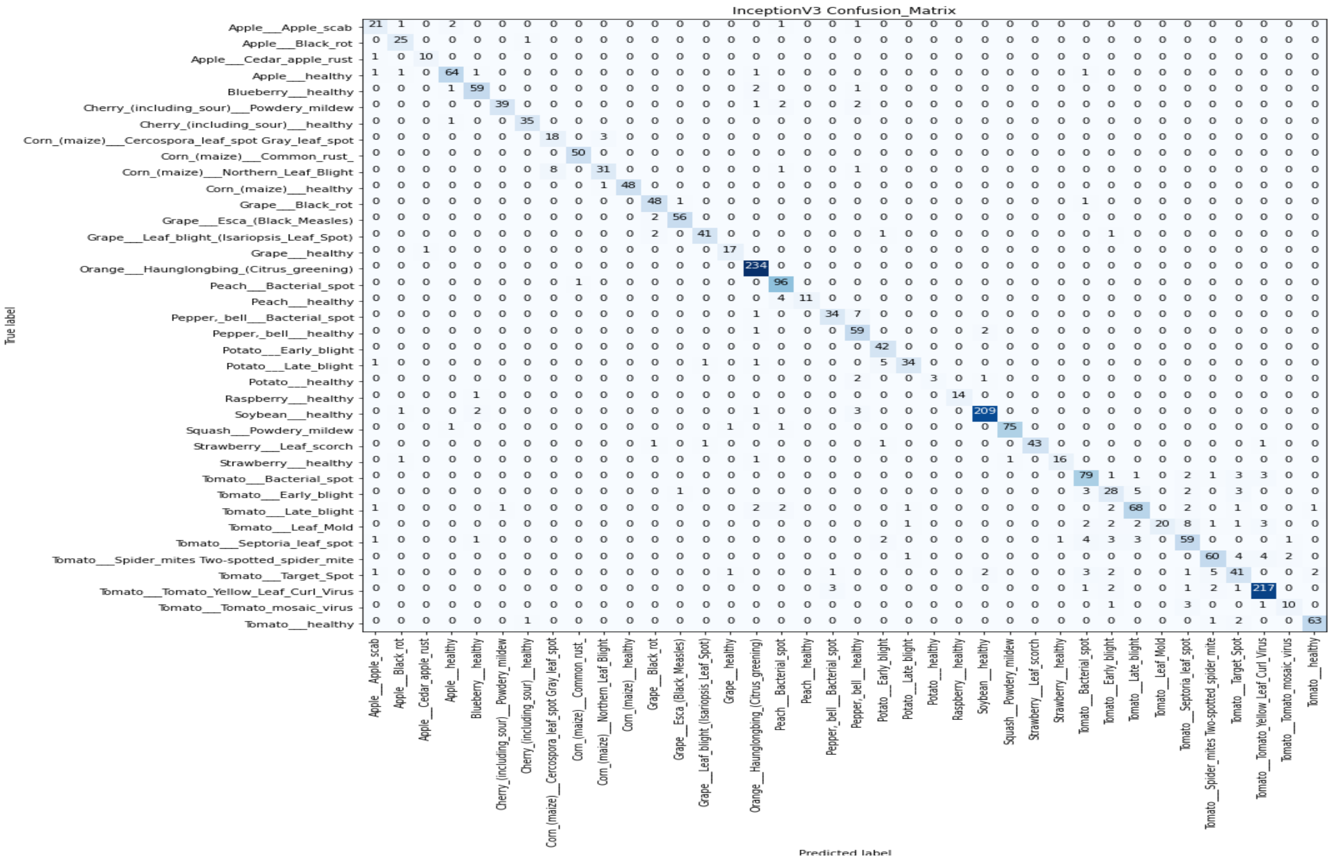

Fig. 7. InceptionV3 Confusion Matrix

ITEE Journal
Information Technology & Electrical Engineering

ISSN: - 2306-708X

During testing using test data, the accuracy of our suggested modals was found to be 94% for the proposed VGG16 model, 99% for the proposed MobileNet model, and 91% for the proposed InceptionV3 model. The models are deployed to the web using Streamlit. Fig. 8 shows the system webapp output.

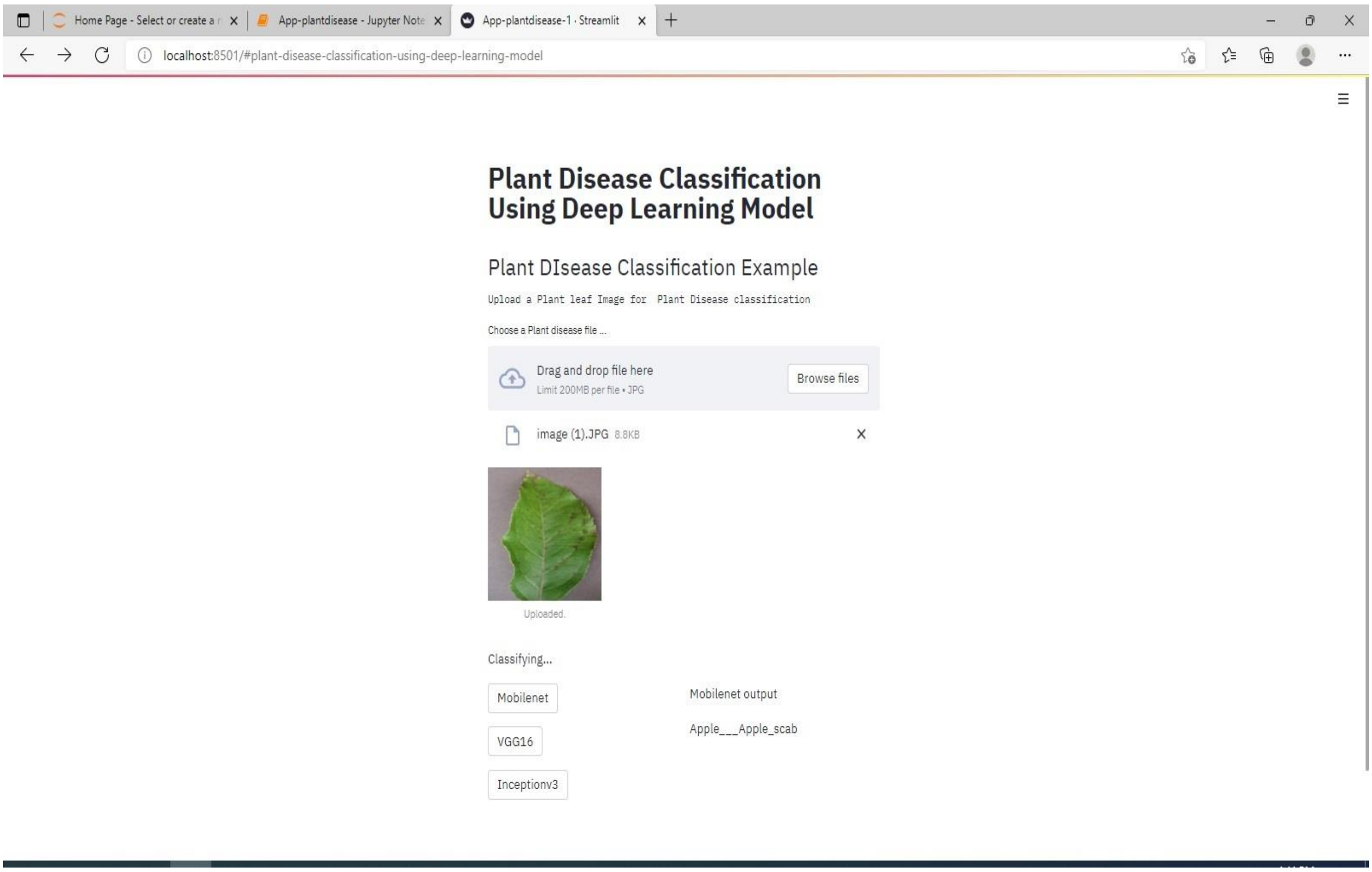

Fig. 8. Webapp output

6. CONCLUSION AND FUTURE WORK

An intelligent system is presented to do multi-class classification of plant diseases using deep learning models. The transfer learning approach has been applied. Three pre-trained CNN models, VGG16, Mobilenet, and InceptionV3, are used in the research work. The proposed system is able to classify plant diseases into 38 different classes of the PlantVillage dataset. The testing accuracy of proposed models VGG16, MobileNet, and InceptionV3 was achieved at 94%, 99%, and 91%, respectively. The models are integrated into the web page using Streamlit.

As part of future research, other learning rates and optimizers might be used to test the suggested system. I will also work on creating a Smartphone-based expert system.

REFERENCES

[1] S. S. Sannakki and V. S. Rajpurohit, "Classification of Pomegranate Diseases Based on Back Propagation Neural Network," International, vol. 2, May-2015.

[2] P. R. Rothe and R. V. Kshirsagar, "Cotton leaf disease identification using pattern recognition techniques," 2015 International Conference on Pervasive Computing (ICPC), pp. 1-6, 2015.

[3] Sue Han Lee and Chee Seng Chan and Simon Joseph Mayo and Paolo Remagnino, "How deep learning extracts and learns leaf features for plant classification," Pattern Recognition, vol. 71, pp. 1-13, 2017.

[4] Halil Durmus and Ece Olcay G, "Disease detection on the leaves of the tomato plants by using deep learning," 2017 6th International Conference on Agro-Geoinformatics, pp. 1-5, 2017.

[5] Konstantinos P.Ferentinos, "Deep learning models for plant disease detection and diagnosis," Computers and Electronics in, vol. 145, pp. 311-318, 2018.

[6] Alla Pranathi, Kandiraju SaiAshritha, Nagaratna B. Chittaragi, Shashidhar G. Koolagudi Prajwala Tm, "Tomato Leaf Disease Detection Using Convolutional Neural Networks," 2018 Eleventh International Conference on Contemporary Computing (IC3), pp. 1-5, 2018.

ITEE Journal
Information Technology & Electrical Engineering

ISSN: - 2306-708X

[7] Omkar Kulkarni, "Crop Disease Detection Using Deep Learning," 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), pp. 1-4, 2018.

[8] Sammy V. Militante and Bobby D. Gerardo and Nanette V. Dionisio, "Plant Leaf Detection and Disease Recognition using Deep Learning," 2019 IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE), pp. 579-582, 2019.

[9] M. A. Jasim and J. M. AL-Tuwaijari, "Plant Leaf Diseases Detection and Classification Using Image Processing and Deep Learning Techniques," 2020 International Conference on Computer Science and Software Engineering (CSASE), pp. 259-265, 2020.

AUTHOR PROFILES

Manojkumar B. Patel is an Assistant Professor at the L. D. College of Engineering, Ahmedabad (India). He has done his M.Tech in computer science and engineering from Guru Gobind Singh Indraprastha University, Delhi. He has more than nine years of teaching experience. He has worked on many research projects during his M.tech program. His research interests are in algorithms, image processing, and deep learning.