International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 09 Issue: 05 | May 2022 www.irjet.net p-ISSN: 2395-0072

# Plant Disease Detection using Convolution Neural Network (CNN)

**Nita patil[1], Alpesh Tandel[2], Rushika Gawade[3], Arati kamble[4]**

*[1]Assistant professor, Computer Engineering*
*Datta Meghe College of Engineering, Airoli, New Mumbai – 400708(Maharashtra) (India)*
*[2,3,4]Students, Computer Engineering*
*Datta Meghe College of Engineering, Airoli, New Mumbai – 400708(Maharashtra) (India)*

---

***Abstract*— When crop plant is suffering from pests it attacks the agricultural production of the world. As usual farmers and experts focus the plants by eye for detect and notice of disease. But this manual process is time processing, high-cost and inexact. Therefore, there is need for accurate and efficient automatic detection of the plant diseases. The main aim of this project is to find a solution to the problem of 38 different classes of plant diseases detection using the simplest approach while making use of minimal computing resources to achieve better results and high accuracy compared to the traditional models. Convolution Neural Network (CNN) training model is deployed for detection of plant diseases. CNN model employs automatic feature extraction to help in the classification of the input image into respective disease classes. This proposed system has achieved an average accuracy of 92% indicating the feasibility of the neural network approach even under unfavorable conditions.**

***Key Words***: CNN, ***Disease detection, Confusion Matrix***

## 1. INTRODUCTION

The agriculture production of the farmer is much reduced if crops are having pests. Manual process of identification and recognition of diseses by experts is cumbersome and takes lot of time. Previous technologies based on feature extraction from the images of the plant have less accuracy as compared to deep learning approaches. l All steps are required for implementing this disease recognition model are fully expressed in this project, starts from collecting images and information to make a database, evaluate by agricultural experts, a deep learning substructure to achieve the deep CNN learn. This project may be a new perspective in detecting plant diseases using the deep convolutional neural network trained and refine to become accurately to the database of a plant's leaves that was converged unaccompanied for diverse plant diseases.

The proposed model is detecting the disease from the given plant leaf image. This can be achieved using CNN. In the proposed model we have used Convolution Neural Network (CNN) for the classification of plant disease detection. Convolutional Neural Network (CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. Using CNN for plant disease detection is not a new domain. A lot of work has been already done on this domain. But we have found out that most of the research used very small dataset as well as the activation functions that they used can be replaced with large dataset and various optimization functions. So, the proposed model will be having improved the accuracy of plant disease detection using CNN over the previous work by applying newer technique. We have also created a simple website giving the image as an input and the website will classify the Plant disease detection using the proposed CNN model. The dataset that we have used for this project consists of plant leaf images which were taken from Kaggle dataset.

## 2. LITERATURE SURVEY

Prasanna Mohanty et.al, 2016 [1], detected disease in plants by training a convolutional neural network. CNN model is trained to classify healthy and unhealthy plant of 14 crops. This model achieved an accuracy of 99.35% on test dataset. Malvika Ranjan et.al, 2017 [2], proposed use HSV features for feature extraction on cotton plant and used Artificial Neural Network (ANN) to classify disease crops and healthy samples. The ANN model achieved an accuracy of 80%.

S. Arivazhagan et.al, 2013 [3], proposed model process involves four main process as follows first, a color transformation structure is take as input RGB picture, and then it means of a specific threshold value then green pixels are detected , which is followed by segmentation steps, and for obtaining advantages of segments the texture statistics are calculated. At the end, classifiers are used for features that are taken to identify the disease. Kulkarni et.al, 2017 [4], applied image processing methods to identify the plant diseases for right and accurate detection of plant diseases using artificial neural network (ANN) and diverse image processing methods. As the proposed approach is based on ANN classifier and Gabor filter for feature extraction, it got better output with a rate of recognition is 91%. R.P Narmadha et.al, 2017 [5], applied Image processing techniques which can be defined as the technical analysis of an image by detecting the disease of plants leaf using complex algorithms of machine learning. B R , Jagdesh et.al, 2019 [6], proposed methodology in which histogram matches and their histogram specification is the transformation image so that their histogram matches the

specified histogram in image Processing, which is well-known by histogram equalization method is a special case in which the specified histogram is uniformly distributed. Dhiman Mondal et.al, 2018 [7] , proposed methodology in which Naive Bayes is a kind of classifier which uses the Bayes Theorem that predicts membership probabilities for each class such as the probability that giving the record and data points belongs to a particular class. Prof. Anjali A Yadav et.al. 2016 [8], used support vector machine (SVM) is a type of deep learning algorithm that activate process of supervised learning which using for classification and regression of data groups of SVM methods. An SVM builds a learning model process, which assigns new examples to one group to another groups.

## 3. METHODOLOGY

We used convolution Neural Network (CNN) and it is type of artificial neural network used in image recognition and processing pixel data. The CNN model is built using Python language with TensorFlow and Keras library. The TensorFlow has the used to handle deep learning networks and uses many backend software like GUI and ASIC. Kearas is a high-level neural network that run on top of (works as an interface) for TensorFlow library.

### 3.1 WORKING OF CNN MODEL

CNN consists of 4 different types of layers:

1. Input Layer
2. Convolution Layers
3. Max-Pooling Layers
4. Fully Connected Layers

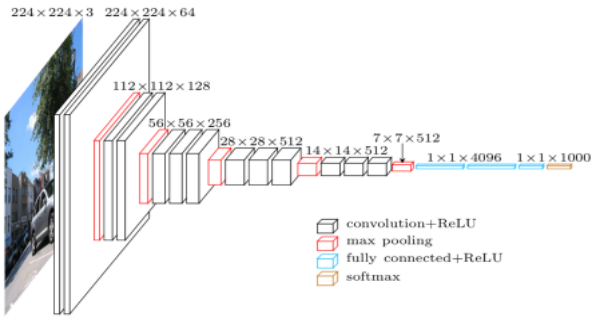


Fig. 1. Layered view of sample CNN Model

In the above example shows how different layers in CNN model are stacked together in order to classify the input image.

#### 2.1.1 INPUT LAYER

The input layer is the very beginning of the workflow for the convolution neural network. In Convolution neural Network input layer should contain image data. Image data is represented by three-dimensional matrix.

#### 3.1.2 CONVOLUTION LAYER

The First layer of CNN is Convolution Layer. Convolution layer is usually called as feature extractor layer because features of the image are get extracted within this layer. CNN uses filters to extract input image features. The element of image is connected to Convolution layer to perform convolution operation. Results of the operation is single integer of the output volume. The output are going to be the input for the subsequent layer. Convolution layer also contains ReLU activation to create all negative value to zero. Rectified Linear Unit can be applied to take in account the non-linearities in the output. Basically, it is just a function that transforms its inputs into outputs that have a certain range.

#### 3.1.3 RECTIFIED LINEAR UNIT (ReLU)

A Rectified Linear Unit (ReLU) is a non-linear activation function that performs on multi-layer neural networks.

$$f(x) = \max(0, x) \quad (1)$$

Where x = an input value

In this layer we remove every negative value from the filtered image and replace it with zero. This function only activates when the node input is above a certain quantity. So, when the input is below zero the output is zero.

#### 3.1.4 MAX–POOLING LAYER

Max pooling is a type of operation that is added to CNN's following individual convolutional layers. Max-pooling reduces the dimensionality of images by reducing the number of pixels in the output from the preceding convolutional layer. The product of Convolution layer and pooling layer is then feed into fully connected layer which drives the final decision on the classification.

#### 3.1.5 FULLY CONNECTED LAYER (DENSE LAYER)

Fully connected layer nothing but the feed forward neural network. Output of the final pooling and convolution layer will be the input of a fully connected layer and that is flatten. Flatten means it unroll the 3-dimension matrix of the final convolution layer into a vector. But the output of convolution and pooling layers both are 2D volumes. The input image from the previous layers are flattened and fed to the Dense. The flattened vector then undergoes few more dense layers where the mathematical functions operations are done to generate classification probabilities. The last layer the Fully Connected layer is used for final classification i.e., in our case it will provide the final classification of the plant disease detection. It can be done by using activation function like Softmax.

### 3.1.6 SOFTMAX

The last layer in the Fully Connected layer is used for final classification. It can be done by using Softmax activation function. The softmax function turns a vector of K real values into a vector of K real values that sum to 1. The input values can be positive, negative, zero, or greater than one, but the softmax transforms them into values between 0 and 1, so that they can be classify as probabilities. If one of the inputs is small or negative, the softmax turns it into a small probability, and if an input is large, then it turns it into a large probability, but it will always between 0 and 1.

The Softmax formula is as follows:

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \quad (2)$$

$\sigma$ = softmax

$\vec{z}$ = input vector

$e^{z_i}$ = standard exponential function for input vector

$K$ = number of classes in the multi-class classifier

$e^{z_j}$ = standard exponential function for output vector

$e^{z_j}$ = standard exponential function for output vector

### 3.1.7 ADAM OPTIMIZER

Adam is an optimization solver for the Neural Network algorithm. Adam is a popular extension to stochastic gradient descent. It uses mini-batch optimization and can make progress faster while seeing less data than the other Neural Network optimization solver.

### 3.2 OUR CNN MODEL

For implementation of this project we have used Jupiter Notebook.

The proposed CNN model consists of 4 Convolution Layers, 4 Max-Polling layers and 2 Fully Connected layers.

Activation functions used for convolution layers and 1 fully connected layers are 'ReLu', while 2nd the fully connected layer uses Softmax for the final classification into 38 classes. Input Size for the input layer of the model is 256 x 256. All the images are resized to 256 x 256 before feeding then to the Input Layer.

Other parameters of our CNN model are as follows:

1. Kernel Size or Filter Size that we have used :3x3
2. For Optimizer we have selected: Adam optimizer
3. Learning rate: Starts from 0.001 then we have used cross entropy function from Tensorflow Keras for monitoring improvement in Validation Loss and changing Learning Rate automatically

The output size of each Convolution layer is calculated using following formula:

Output size = Input_size – (filter_size – 1)

Size of the input Image will be 256 x 256 with 128 filters. And by the end of 4th Convolution layer our output size is 3 x 3 after applying 64 filters. The Output size of the final Dense Layer is 38 because we are classifying given Plant Village Images into 38 classes. Figure 2 shows summary of the finalized CNN model after experimenting with the various kernel sizes and optimization functions.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 256, 256, 128)     3584

max_pooling2d (MaxPooling2D  (None, 128, 128, 128)    0
)

conv2d_1 (Conv2D)            (None, 128, 128, 32)      36896

max_pooling2d_1 (MaxPooling  (None, 64, 64, 32)       0
2D)

conv2d_2 (Conv2D)            (None, 64, 64, 64)        18496

max_pooling2d_2 (MaxPooling  (None, 32, 32, 64)       0
2D)

conv2d_3 (Conv2D)            (None, 32, 32, 64)        36928

max_pooling2d_3 (MaxPooling  (None, 16, 16, 64)       0
2D)

dropout (Dropout)            (None, 16, 16, 64)        0

flatten (Flatten)            (None, 16384)             0

dense (Dense)                (None, 256)               4194560

dense_1 (Dense)              (None, 38)                9766

=================================================================
Total params: 4,300,230
Trainable params: 4,300,230
Non-trainable params: 0
_________________________________________________________________
```

Fig. 2. Summary of the Finalized Model

### 3.3 DATASET USED

The Dataset was taken from Kaggle of Plant Village dataset. We have included vegetable plants of Apple, Blueberry, Cherry, Corn, Grape, Orange, Peach, Paper Bell, Potato, Raspberry, soybean, Squash, Strawberry, Tomato vegetable detection. Dataset have number of images of each plant. Our Plant Village dataset contain 38 classes of plant disease types.

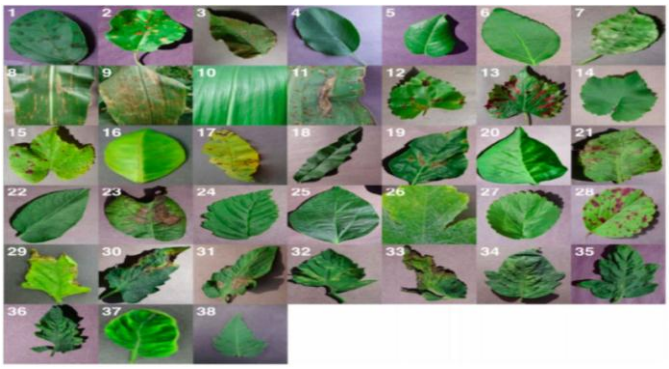

Fig. 3. Sample images of Plant Village dataset for 38 classes.

Our dataset contains a total of 78534 images of disease and healthy plant leaves. Dataset contain 14 plant crops and 38 classes. Some of the images will be used for training of our CNN model while rest of the images will be used for testing. Our training dataset is divided into 80:20 training and validation split, i.e. 80% for training set and 20% for validation set.

```
{'Apple___Apple_scab': 0,
 'Apple___Black_rot': 1,
 'Apple___Cedar_apple_rust': 2,
 'Apple___healthy': 3,
 'Blueberry___healthy': 4,
 'Cherry_(including_sour)___Powdery_mildew': 5,
 'Cherry_(including_sour)___healthy': 6,
 'Corn_(maize)___Cercospora_leaf_spot Gray_leaf_spot': 7,
 'Corn_(maize)___Common_rust_': 8,
 'Corn_(maize)___Northern_Leaf_Blight': 9,
 'Corn_(maize)___healthy': 10,
 'Grape___Black_rot': 11,
 'Grape___Esca_(Black_Measles)': 12,
 'Grape___Leaf_blight_(Isariopsis_Leaf_Spot)': 13,
 'Grape___healthy': 14,
 'Orange___Haunglongbing_(Citrus_greening)': 15,
 'Peach___Bacterial_spot': 16,
 'Peach___healthy': 17,
 'Pepper,_bell___Bacterial_spot': 18,
 'Pepper,_bell___healthy': 19,
 'Potato___Early_blight': 20,
 'Potato___Late_blight': 21,
 'Potato___healthy': 22,
 'Raspberry___healthy': 23,
 'Soybean___healthy': 24,
 'Squash___Powdery_mildew': 25,
 'Strawberry___Leaf_scorch': 26,
 'Strawberry___healthy': 27,
 'Tomato___Bacterial_spot': 28,
 'Tomato___Early_blight': 29,
 'Tomato___Late_blight': 30,
 'Tomato___Leaf_Mold': 31,
 'Tomato___Septoria_leaf_spot': 32,
 'Tomato___Spider_mites Two-spotted_spider_mite': 33,
 'Tomato___Target_Spot': 34,
 'Tomato___Tomato_Yellow_Leaf_Curl_Virus': 35,
 'Tomato___Tomato_mosaic_virus': 36,
 'Tomato___healthy': 37}
```

Fig. 4. Classes of plant leaf diseases

### 3.4 TESTING METHODOLOGY

Testing of our Model is done on two datasets as follows:

1. Validation Set used during training
2. Test Set which contains images that our CNN model has never seen

#### 3.4.1 EVALUATION METRICS

The most important task in building CNN model is to evaluate its performance. Following parameters are used as evaluation metrics.

##### 3.4.1.1 PRECISION

The precision is calculated as the ratio between the numbers of positive samples correctly classified to the total number of samples classified as positive (either correctly or incorrectly). It basically shows how often the prediction is actually correct.

$$\text{Precision} = \frac{tp}{tp + fp} \tag{3}$$

##### 3.4.1.2 RECALL

It quantifies the number positive class prediction made out of all positive example in the dataset.

$$\text{Recall} = \frac{tp}{tp + fn} \tag{4}$$

##### 3.4.1.3 F1-SCORE

It is weighted average between precision and recall. It provides a single score that balances both the concerns of precision and recall in one number.

$$F_1 = \left( \frac{2}{\text{recall}^{-1} + \text{precision}^{-1}} \right) = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{5}$$

##### 3.4.1.4 ACCURACY

It shows the overall accuracy of our model. It is calculated using sum of true positives (tp) and true negatives (tn) divided by the total number of samples i.e. (tp + tn + fp + fn).

$$\text{Accuracy} = \frac{tp + tn}{tp + tn + fp + fn} \tag{6}$$

Where in equation (3), (4), (6) following terms are used as

tn= true positive

tp= true negetive

fp= false positive

fn= false negative.

### 3.5 CONFUSION MATRIX

A confusion matrix is a summarized table of the number of correct and incorrect predictions yielded by a classification model. A confusion matrix is a performance measurement for our proposed model. It basically generates a Matrix Table which compares Predicted Labels with True Labels.

## 4. RESULTS AND ANALYSIS

The objective of this project is to detect the plant disease using CNN on 38 classes like Apple black rot, Apple cedar rust, cherry powdery mildew, Grape leaf blight, tomato early bright and potato late blight. Results of this project allowed us to determine which parameters such as kernel size and optimizer are best suited in order to accomplish our objective on validation dataset. The experiments are performed on Dell laptop with Ryzen 3 processor with 4 GB RAM. First we have tested the performance of our model using different parameters and after selecting the parameters with best performance, we have tested our finalized model on the test datasets.

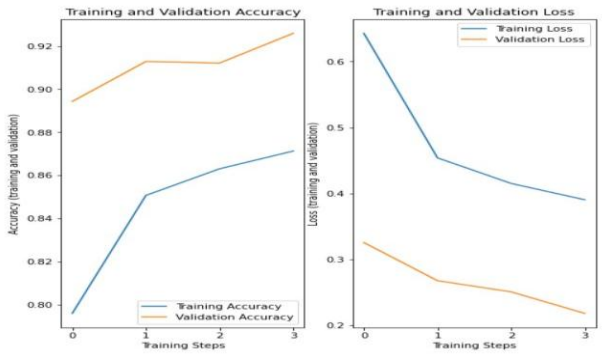


Fig. 5. Graph of Training Loss vs Validation Loss and Training Accuracy vs Validation Accuracy

After testing the proposed model on test Dataset following Confusion Matrix is generated using Sklearn Matrics shown by Figure 6.

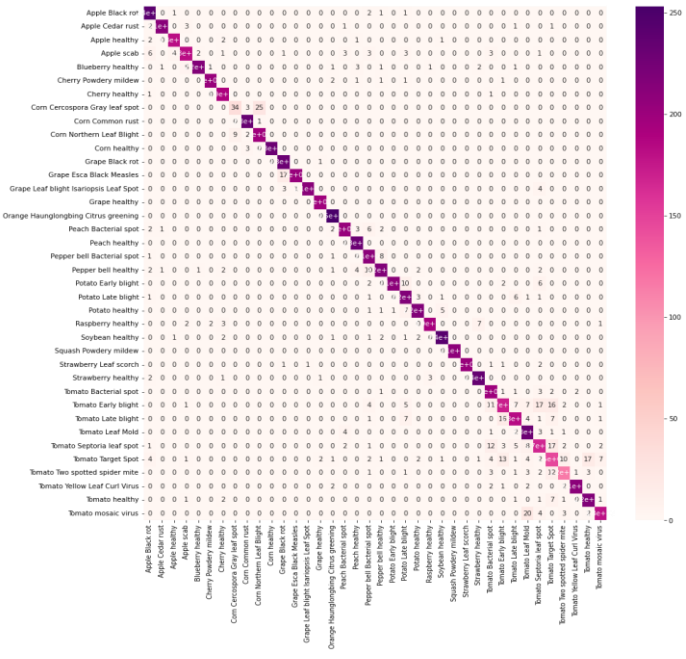

Fig. 6. Confusion Matrix using finalized model on Test Dataset.

We have calculated the accuracy of our finalized model on Test Dataset as 92 %. Fig. 7 shows the class wise performance for evaluation parameters on test Dataset.

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| Apple Black rot | 0.91 | 0.98 | 0.94 | 246 |
| Apple Cedar rust | 0.99 | 0.96 | 0.97 | 220 |
| Apple healthy | 0.97 | 0.97 | 0.97 | 187 |
| Apple scab | 0.93 | 0.87 | 0.90 | 209 |
| Blueberry healthy | 0.99 | 0.93 | 0.96 | 232 |
| Cherry Powdery mildew | 0.99 | 0.97 | 0.98 | 209 |
| Cherry healthy | 0.94 | 0.99 | 0.96 | 192 |
| Corn Cercospora Gray leaf spot | 0.79 | 0.55 | 0.65 | 62 |
| Corn Common rust | 0.97 | 1.00 | 0.98 | 234 |
| Corn Northern Leaf Blight | 0.88 | 0.95 | 0.91 | 209 |
| Corn healthy | 1.00 | 0.99 | 0.99 | 233 |
| Grape Black rot | 0.91 | 1.00 | 0.95 | 231 |
| Grape Esca Black Measles | 1.00 | 0.92 | 0.96 | 220 |
| Grape Leaf blight Isariopsis Leaf Spot | 1.00 | 0.96 | 0.98 | 220 |
| Grape healthy | 0.98 | 1.00 | 0.99 | 198 |
| Orange Haunglongbing Citrus greening | 0.96 | 1.00 | 0.98 | 253 |
| Peach Bacterial spot | 0.95 | 0.92 | 0.94 | 220 |
| Peach healthy | 0.95 | 1.00 | 0.97 | 231 |
| Pepper bell Bacterial spot | 0.86 | 0.95 | 0.90 | 220 |
| Pepper bell healthy | 0.92 | 0.90 | 0.91 | 242 |
| Potato Early blight | 1.00 | 0.91 | 0.95 | 231 |
| Potato Late blight | 0.86 | 0.94 | 0.90 | 231 |
| Potato healthy | 0.96 | 0.94 | 0.95 | 231 |
| Raspberry healthy | 0.98 | 0.93 | 0.95 | 209 |
| Soybean healthy | 0.97 | 0.96 | 0.96 | 253 |
| Squash Powdery mildew | 1.00 | 1.00 | 1.00 | 209 |
| Strawberry Leaf scorch | 1.00 | 0.97 | 0.99 | 209 |
| Strawberry healthy | 0.96 | 0.97 | 0.97 | 242 |
| Tomato Bacterial spot | 0.84 | 0.95 | 0.89 | 209 |
| Tomato Early blight | 0.82 | 0.71 | 0.76 | 242 |
| Tomato Late blight | 0.88 | 0.83 | 0.85 | 220 |
| Tomato Leaf Mold | 0.82 | 0.95 | 0.88 | 242 |
| Tomato Septoria leaf spot | 0.77 | 0.76 | 0.76 | 220 |
| Tomato Target Spot | 0.70 | 0.67 | 0.68 | 220 |
| Tomato Two spotted spider mite | 0.85 | 0.81 | 0.83 | 143 |
| Tomato Yellow Leaf Curl Virus | 0.99 | 0.96 | 0.97 | 220 |
| Tomato healthy | 0.91 | 0.94 | 0.92 | 231 |
| Tomato mosaic virus | 0.93 | 0.86 | 0.90 | 209 |

Fig. 7. Performance measure on each class.

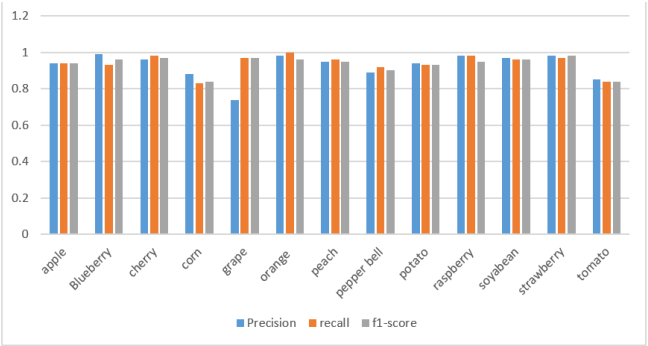


Fig. 8. Precision, recall and f1-score of each crop.

The experiments are performed on Plant Village dataset which consist of healthy and disease plant leaf image. For input data disease, samples of plant leaves like apple with black rot disease, tomato leaf with early blight disease, corn leaf with northing leaf blight, potato with early blight disease etc. are considered. CNN model is trained to identify diseases of each plant class. The performance of our model is shown by the specific error table known as confusion matrix. It shows correctly and incorrectly prediction for each class.

The performance of the model is evaluated by average value of diagonal entries which represents the number of correctly predicted disease. The accuracy for our CNN model is 82%. Figure 8 shows the precision recall and f1-score for each class. The average precision, average recall and average f1-score for all crops disease are found to be 0.91,0.93 and 0.92 respectively.

## 5. CONCLUSION

A large part of the Indian population depends on agriculture, hence it becomes very needful to detect the plant diseases that results in losses, since agriculture is critical to the growth of our economy. This project based on deep learning approach called CNN is utilized to build 14 different plant leaf disease detection system. This approach utilized a minimum set of layers such as convolution layer, ReLu layer, Max- pooling, fully connected layer to identify the diseases of 38 classes. The neural network is trained with PlantVillage dataset. A GUI is designed for this system. This GUI permits the user to choose the images from the dataset. User can select any image from the dataset and the image gets loaded, following which the prediction of the disease will be shown on the User Interface. Convolutional neural network, trained for detecting the plant leaf disease, predicts the disease of the test image correctly for almost all the images with few anomalies. We obtained 92% accuracy on test dataset of Kaggle. The average precision, average recall and average f1-score for all crops disease are found to be 0.91,0.93 and 0.92 respectively.

## 6. FUTURE WORK

Our project is deployed into the web application. It can be extended to use as an embedded application. More number of the images can be added to improve accuracy along with the testing of transfer learning. For the large scale open field cultivation we can use real time monitoring using drones and other autonomous agriculture vehicles.

## REFERENCES

[1] Prasanna Mohanty, David Hughes and Marcel Salathe, "Using Deep Learning for Image-Based Plant Disease Detection", 2016, Frontiers in Plant Science,7(September),[1419]. https://doi.org/10.3389/fpls.2016.01419.

[2] Malvika Ranjan1, Manasi Rajiv Weginwar, NehaJoshi, Prof.A.B. Ingole, detection and classification of leaf disease using artificial neural net-work, International Journal of Technical Research and Applications e-ISSN: 2320-8163, Volume 3, Issue 3 (May-June 2017), PP. 331-333

[3] S.Arivazhagan, R. Newlin Shebiah, S.Ananthi, S.Vishnu Varthini. 2013. "Detection of unhealthy region of plant leaves and classification of plant leaf diseases using texture features". Agric Eng Int: CIGR Journal

[4] Pranesh Kulkarni, Atharva Karwande and Tejas Kolhe, "Plant Disease Detection Using Image Processing and Machine Learning", 2021.

[5] R.P Narmada, G Arulvadivu, "Detection And Measurement of Paddy Leaf Disease Symptoms using Image Processing" in: International Conference on Computer Communication and Informatics, 2017.

[6]Nanjesh B.R, Jagadeesh, Ashwin GeetD'sa "Plant Disease Analysis Using Histogram Matching Based on Bhattacharya's Distance Calculation" International Conference on Electrical, Electronics and Optimization Techniques (ICEEOT)-20164 Vector Machine", 2019 IEEE.

[7] Dhiman Mondal and Dipak Kumar Kole, "Detection and Classification Technique of Yellow Vein Mosaic Virus Disease in Okra Leaf Images using Leaf Vein Extraction and Naive Bayesian Classifier", 2015 International Conference on Soft Computing Techniques and Implementations- (ICSCTI), Oct 8- 10, 2015pp 166-171.

[8]Pranjali B. Padol and Anjali A. Yadav, " SVM classifier based grape leaf disease detection", 2016,Conference on Advances in Signal Processing (CASP).