JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Plant Disease Detection using CNN Techniques

[1]S Sivakama Sundari, [2] Dr. Sampath Kumar AK, [3]Dr. M. Islabudeen

[1] M. Tech Artificial Intelligence, Department of CSE, School of Engineering, Presidency University, Bangalore

[2] Associate Professor CSE, School of Engineering, Presidency University, Bangalore

[3] Associate Professor CSE, School of Engineering, Presidency University, Bangalore

***Abstract*:** *Plants are the meals supply of the earth. Plant infections and illnesses are consequently a first-rate threat, however the maximum not un-usual place prognosis is basically to study flora for the presence or absence of visible symptoms. The agricultural production of the country gets affected majorly due to pests as they affect the plants and crops. The detection and identification of disease is been observed by farmers and experts through their naked eyes. Based on the leaf image classification, an approach of plant disease recognition model is being developed with the help of deep convolutional networks. Early detection of diseases to which plants are exposed is very important, especially in a country like India with a large population. The diseases caused by bacteria, virus and fungus results on lowering the crop yield in a huge aspect. The loss can be prevented by predicting the plant disease at the earliest. With the help of Deep Learning concepts, the performance and accuracy of disease detection can be improved. It uses image processing concepts for noise reduction, ML and DL concepts i.e., CNN for Problem Solving. This project captures plants and leaf disease and helps farmers to identify and detect the solutions for the problem that is being infected.*

Keywords: *Plant Disease Recognition Model, Image Classification, Plant Disease Recognition, Image Processing, Machine Learning Concepts, Deep Learning Concepts*

1. INTRODUCTION

According to April 2020, Country Like has a population of around 1.38 billion, with an estimated 95.8 million working as farmers in India. Note that 18% of India's GDP comes via the agriculture category. If agriculture turned into revolutionized, the situation of the neighborhood farmers might create quite a few employment and enlargement possibilities withinside the agricultural sectors. In India, the improvement and studies on pesticides, fungicides and herbicides has been stepped forward very well. Agriculture has always been the backbone of developing countries. In order to make the people of such countries economically wealthy and strong, they must harvest sufficient quality and quantity from agriculture. Many crops are damaged each year by bad weather, viruses and various plant diseases. Identification of new diseases on plants and crops have been failed by farmers. So, plant does now no longer get precise remedy for precise ailment or viruses. Usually many farmers can't manage to pay for the specialists advices because of loss of cash and different instances like traveling lengthy distance to get the assist, and the time-ingesting strategies. Today's higher generation has enabled human beings to offer the ok vitamins and nourishment had to meet the wishes of the world's developing populace. The illness of vegetation haven't any inclination withinside the improvement which might also additionally waken hardships greater basically than half. The signs exhibited via way of means of the vegetation result in an wrong diagnosis, as non-expert gardeners might also additionally have a more difficult time recognizing them than plant pathologists. An automated gadget designed to help perceive plant sicknesses via way of means of the plant's look and visible signs is probably of superb assist to amateurs withinside the gardening procedure and skilled specialists as a affirmation gadget in ailment diagnostics Vegetables and culmination are not unusual place gadgets and the main agricultural things. Powerful dependence on engineered pesticides achieves the immoderate substance content material fabric which creates withinside the earth, air, water, and shockingly in our bodies antagonistically have an impact at the environment. Farmers can take advantage of the information age of precision farming to gather facts and make gratifying choices over excess agricultural performance. Precision farming can be used for batch packages including crop pest detection, weed exposure, crop yielding, and crop disease detection. Farmers use pesticides to control pests, prevent disease, and increase yields. Plant ailment detection is paramount to a hit agricultural gadget. Farmers commonly understand plant ailment signs with the bare eye and this calls for steady monitoring.

Using a normal camera, virtually click on a photo of the affected vicinity and add it to the gadget, and the ailment can be diagnosed and an appropriate remedy and insecticides can be supplied if needed. Exponential populace growth, climatic situations additionally motive plant sicknesses. To come across ailment, leaves should be cautiously monitored. Most vegetation are stricken by numerous fungal and bacterial sicknesses. In preprocessing, the cropped snapshots are scaled to 256*256 resolution to estimate computation time. A background reduction technique was performed to shift the past history about the image. Deep studying-primarily based totally strategies, specifically CNNs, are the maximum promising procedures for routinely studying essential and identifiable features.

Deep studying (DL) includes numerous convolutional layers that constitute residences found out from statistics. Plant ailment detection may be carried out the use of deep studying models. Deep studying additionally has a few drawbacks because it calls for a huge quantity of statistics to teach the network. The creation of those deep studying strategies into agriculture and mainly withinside the subject of plant ailment diagnosis , has simplest all started to take area withinside the In current years, on a as a substitute confined scale. The simple deep learning device used in these paintings is CNN. CNN is alone a most effective strategies for modeling complex strategies. and appearing sample matching on huge-statistics packages along with picture sample matching. Their outcomes have been very encouraging, with automated identity fulfillment quotes as much as 99.35%. For the carried out DL architecture, we evaluated overall performance the use of numerous batch sizes from 32 to 180. Various dropout values and studying quotes have been extensively utilized for overall performance studies. Several epochs have been used to run the model. Evaluations confirmed that the carried out Deep CNN completed brilliant outcomes and advanced overall performance in comparison to modern gadget studying strategies.

2. PROBLEM STATEMENT

A Disease Recognition Model based on leaf image classification is the objective of this paper. We use image processing with a Convolution neural network (CNN) to find diseases in plants. A type of artificial neural network called a convolutional neural network (CNN) is used in image recognition and is designed to process pixel input.

3. RELATED SURVEYS

In an article, Liu recommends a advanced copy of deep convolutional networks for detailed forecast and recognition of apple leaves. The classic proposed in this document can undoubtedly detect various frame transactions with very high efficiency. An absolute of 13,689 photographs were put together using image processing techniques that are equivalent as PCA vibration. Apart from this new his AlexNet-based neural network, it is also proposed to implement the NAG algorithm to optimize the network. Future work on the prediction of apple leaf perturbations may implement alternative deep learning models alike F-CNN, R-CNN and SSD.

Paper [1] also characterize different approach for extracting infected leaf seeds and classifying plant diseases. Here we use a convolutional neural network (CNN) that dwell of different layers used for prediction. The entire procedure is so-called stationed on the images used for coaching and preprocessing examining and image improvement, followed by CNN deep and optimizer training procedures. Based on the particular images, treatment methods can be accurately determined and different plant diseases can be identified.

The paper [2] proposed a path to identify plant diseases adopting GAN. Background analysis is utilised to ensure features extracted are apt along with the mapping of output. We can see a certain geese that could potentially classify the disease faced by the crops, but background-based segmentation did not improve accuracy.

In articles [3-6], Researchers present sound knowledge strategies to explain highly complex undertakings in various research areas in science, e-science, medicine, fixed autonomy, and 3D approaches.

Plant disorder is being detected via the usage of network coding schemes [7-10]. Various data stream approaches, ANNs, and data analysis are also some of the approaches that might being used for detection [11-14].

CNN technology is utilised for diabetes and cancer detection [15,16]. The work uses deep awareness strategies for plant disease detection, directed by the emergence of deep awareness systems and its exploitation

The aim of Syafiqah Ishakais and colleagues' research on leaf disease classification by Artificial Neural Networks is to collect and evaluate data from leaf photographs and use image processing methods to take a decision whether the leaves of medicinal plants are healthy or sick.

A new crop disease detection system stationed on DCN along with Plant Image Classification was published by Srdjan Sladojevic and fellowship as Deep CNN reinforced on recognition of Crop Infection by Plant depiction Classification. The methodology used along with the initial ability to create new processes represent a generally simple framework that continues to be built.

An exposed group of 87,848 photos was intended for the models, which included 25 distinct plant species in 58 distinct [plant, disease] pair classes and non-diseased plants. Various model designs were created, one of which had a 99.53 percent success rate. The model is a useful tool for early detection due to its high success rate. [17]

With 20 ages, the settled characteristics are molded into the cerebrum association. With a significant accuracy of 98.59 percent, abundant neural network-based topologies are used to predict the plant disease. [18]

The technique recommends that banana plant leaf infections can be identified naturally. It is impossible to express the orientation and posture of objects because of the CNN max pooling layer's inherent limitations. In light of the drawbacks, a new model known as the Capsule Network (CapsNet) has been implemented. The fabricated model accurately recognized the banana bacterial shrink, dark sigatoka, and solid leaves with a test exactness of 95%. [19]

We figured out how well each model worked by looking at how damaged or discolored the leaves were. The recognition rate is greater than 94 percent, despite the fact that thirty percent of the leaves are damaged. In sequence to evolve a visual organisation that is comparable to how humans recognise plant species, we intend to identify the leaves that are attached to the branches in subsequent research.[20]

4. ALGORITHMS USED

CNN: CNN is a more advanced version of simple ANN that generates superior images. because specific things appear in recurring patterns in every image. CNN's two most important features are pooling and convolution. Convolution is used to locate the edges of patterns in an image and pooling is used to shrink an image.

RESIDUAL NETWORK (ResNet): To address the issue of the shattering gradient, here planning initiates the idea of residual blocks. It connects a layer's actuate to those of other thickness by passing over some thickness at intervals. A block remains as a result of this. By stacking these lingering blocks together, ResNets are created.

Instead of accommodating the layer determine the underlying map, the indicated structure adapts the system to the residual map.

Initial mapping, allowing the system to fit, say $H(x)$.

$F(x) = H(x) – x$ which gives $H(x) = F(x) + x$.

XCEPTION MODEL: The term "extreme inception" refers to Xception, which takes the principles of Inception to an extreme. Inception applied distinct filters to each depth space derived from the various input spaces and compressed the initial input using 1x1 convolutions. Xception simply reverses this step. Instead, it compresses the input space by applying 1X1 convolution across the depth after filtering each depth map first. A depth-wise separable convolution, which was first used in brain network planning in 2014, is virtually identical to this strategy.

DENSETNET121 MODEL: DenseNet, or the Thick Convolutional Organization, is an engineering that utilizes more limited associations in the middle of layers to make profound studying networks further additionally making them simpler to prepare. DenseNet is a CNN in every layer communicates with the network's deeper layers.

5. RELATED WORK

Seizures and a variety of disease-related conditions can affect plants. There are different causes which can be portrayed by their impact on plants, unsettling influences because of ecological circumstances like temperature, mugginess, exorbitant or lacking food, light and the most well-known infections like bacterial, viral and contagious sicknesses.

The main objective of this project:

- ✓ To determine the disease faced by the plant leaf.
- ✓ To prevent farmers from guessing the disease faced by the plant.

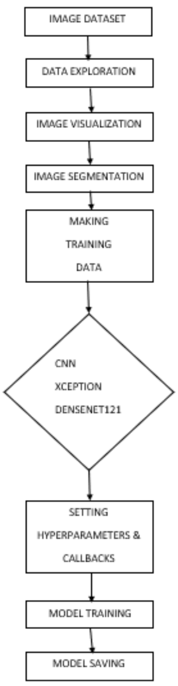

Fig. 5.1 System Flowchart

Many models and algorithms is being made use for the project title namely, ResNet, VGG-16, MobileNetV2, Support Vector Machines, ANN Classifier, KNN Classifier etc.

6. DATASET

PLANT VILLAGE DATASET

The dataset contains images of various plants with various disease types along with healthy plant leaf images. In this we have 4 types of leave images namely, Healthy, Scab, Rust and Multiple Diseases.

Fig. 6.1 Multiple Diseases Image

Fig. 6.2 Rust Image

Fig. 6.3 Scab Image

Fig. 6.4 Healthy Image

Attributes	**Information**
Source	https://www.kaggle.com/datasets/emmarex/plantdisease
No. of images	87700
Classification	CNN and ResNet
Used for	Training the model

Table 6.1: Information regarding Plant Village dataset

PLANT PATHOLOGY DATASET

The "Plant Pathology Challenge" aims to train a model using images from the training dataset in order to 1) precisely classify an image from the testing dataset into a different category of diseased leaf or a healthy leaf; and 2) Identify numerous diseases accurately, sometimes multiple diseases on a single leaf; 3) Address unusual classes and symptoms; 4) Focus on depth perception—angle, light, shade, and the leaf's physiological age; 5) When searching for relevant features during learning, incorporate expert knowledge of identification, annotation, quantification, and computer vision.

Fig. 6.5 Healthy Image

Fig. 6.6 Scab Image

Fig. 6.7 Rust Image

Attributes	**Information**
Source	https://www.kaggle.com/datasets/emmarex/plantdisease
No. of images	2000
Classification	CNN and ResNet
Used for	Training the model

Table 6.2: Information regarding Plant Pathology dataset

7. SECTION 01

METHODOLOGY

There are various steps in this work, such as dataset, exploring the data, visualising the data, model building, evaluation and prediction. The proposed work mainly focuses on predicting the disease that the plant faces by training the images with some online datasets as mentioned above.

7.1 Module 1- Dataset

As mentioned above table 1 and 2, we have made use of both the datasets along with some images that are clicked using mobile camera.

7.2 Module 2- Exploring the data

Data exploration take place in every code which ive made use to predict the diseases. Data exploration in other words means loading the data and reading the data.

7.3 Module 3- Visualizing the data

Data that is present in the dataset is represented in the form of graphs using python programming in which the graph depicted is shown for the explored data.

7.4 Module 4- Model building

In this we will present the details on how the model is being proposed.

Step 1: Importing libraries

A step in every code that help execute the code with required libraries has to be imported. Using the libraries written in the code represents the model building and prediction on how the code executes effectively. Libraries like Numpy, Pandas, Sklearn, Keras, Matplotlib, TensorFlow, Tensor Dash.

Step 2: A glimpse of the dataset

```
[INFO]: Current File is ['Strawberry__healthy', 'Strawberry__Leaf_scorch', 'Blueberry__healthy', 'Potato__Late_blight', 'Pe
pper,_bell__healthy', 'Apple__Black_rot', 'Tomato__Tomato_mosaic_virus', 'Peach__Bacterial_spot', 'Soybean__healthy', 'Tom
ato__Early_blight', 'Grape__Black_rot', 'Tomato__Septoria_leaf_spot', 'Squash__Powdery_mildew', 'Corn_(maize)__healthy',
'Tomato__Bacterial_spot', 'Tomato__Target_Spot', 'Apple__Apple_scab', 'Tomato__Spider_mites Two-spotted_spider_mite', 'Cher
ry_(including_sour)__Powdery_mildew', 'Corn_(maize)__Northern_Leaf_Blight', 'Tomato__Tomato_Yellow_Leaf_Curl_Virus', 'Corn_
(maize)__Cercospora_leaf_spot Gray_leaf_spot', 'Corn_(maize)__Common_rust_', 'Potato__Early_blight', 'Raspberry__healthy',
'Grape__healthy', 'Apple__healthy', 'Apple__Cedar_apple_rust', 'Grape__Leaf_blight_(Isariopsis_Leaf_Spot)', 'Cherry_(includ
ing_sour)__healthy', 'Tomato__Late_blight', 'Tomato__healthy', 'Peach__healthy', 'Orange__Haunglongbing_(Citrus_greenin
g)', 'Pepper,_bell__Bacterial_spot', 'Grape__Esca_(Black_Measles)', 'Potato__healthy', 'Tomato__Leaf_Mold'] ...
```

Fig. 7.1 unique plants present in the dataset

Step 3: Loading the data into the model

The below Fig. 7.2 depicts the output of how the data is getting loaded into the model for training and validation. This is one amongst the folder which is getting loaded into the model.

```
[INFO]: Load picture from folder Strawberry___healthy
[INFO]: Start loading picture b6c7ab0a-7c78-4c75-aa71-878972b5a4b4___RS_HL 4569.JPG ...
[INFO]: Start loading picture c1ccdd49-6389-40eb-b841-c77ab6568014___RS_HL 2056.JPG ...
[INFO]: Start loading picture d454c3f8-ca2e-4935-9c09-3b6a480ac097___RS_HL 1870.JPG ...
[INFO]: Start loading picture 57c57efe-e32c-447b-b60d-cdedf0f86b28___RS_HL 4673.JPG ...
[INFO]: Start loading picture 243609c1-0a35-4474-8f7c-02dc34044735___RS_HL 1663.JPG ...
[INFO]: Start loading picture 4abe583a-841b-4173-98d5-548dcbc16275___RS_HL 2129.JPG ...
[INFO]: Start loading picture 75d24cc5-b37f-42de-947f-dc7ee673674e___RS_HL 4571.JPG ...
[INFO]: Start loading picture 2b349e9d-0131-444a-acda-9b4154073cb5___RS_HL 4507.JPG ...
[INFO]: Start loading picture 6829e297-83c4-4032-9fb2-34baf6bb64e7___RS_HL 4576.JPG ...
[INFO]: Start loading picture f36cfb30-4d03-46e2-8899-d372bf05212f___RS_HL 2112.JPG ...
[INFO]: Start loading picture 26d742d2-76d6-41bb-ba8e-ea068793d74c___RS_HL 4334.JPG ...
[INFO]: Start loading picture c00b240e-63b9-4b16-b7ba-958f0be960e4___RS_HL 1684.JPG ...
[INFO]: Start loading picture 4596027f-5ac4-49f9-bcbd-604d3f5a437a___RS_HL 4790.JPG ...
[INFO]: Start loading picture f58ee24e-67e0-4883-a6b9-9591634e535c___RS_HL 1949.JPG ...
[INFO]: Start loading picture 33d585b6-736d-4767-9004-2444f4916fd9___RS_HL 4360.JPG ...
[INFO]: Start loading picture c051460a-5c1c-4a9e-8921-0150225545fa___RS_HL 4849.JPG ...
[INFO]: Start loading picture 99140b91-93f1-4ccf-9761-e0370717baf6___RS_HL 4552.JPG ...
[INFO]: Start loading picture 057e51f5-f5c1-40b2-930b-a2346c138969___RS_HL 4596.JPG ...
[INFO]: Start loading picture b127a511-d0ce-4fdc-bbe6-6c2a2e2dcf1f___RS_HL 1816.JPG ...
[INFO]: Start loading picture 1e007f9a-6e90-4dd6-9be9-f0ce3775128f___RS_HL 2076.JPG ...
[INFO]: Start loading picture 1eb7dc9e-fbbb-4a62-8a71-eed5f2cb6845___RS_HL 4614.JPG ...
[INFO]: Start loading picture 2fbe071f-efd0-4a93-97b5-38160dafa0c5___RS_HL 2146.JPG ...
[INFO]: Start loading picture a34f4043-6213-42b5-a837-b372ef04c959___RS_HL 4454.JPG ...
[INFO]: Start loading picture d1aee44a-b6bb-45b9-b7b6-5d553add8fd1___RS_HL 2163.JPG ...
[INFO]: Start loading picture 741e834f-a63a-4efd-b961-d5f7e047abdc___RS_HL 2085.JPG ...
```

Fig. 7.2 Data Loading into the model

Step 4: CNN Model

```python
model = keras.Sequential()

model.add(keras.layers.Conv2D(32,(3,3),activation="relu",padding="same",input_shape=(256,256,3)))
model.add(keras.layers.Conv2D(32,(3,3),activation="relu",padding="same"))
model.add(keras.layers.MaxPooling2D(3,3))

model.add(keras.layers.Conv2D(64,(3,3),activation="relu",padding="same"))
model.add(keras.layers.Conv2D(64,(3,3),activation="relu",padding="same"))
model.add(keras.layers.MaxPooling2D(3,3))

model.add(keras.layers.Conv2D(128,(3,3),activation="relu",padding="same"))
model.add(keras.layers.Conv2D(128,(3,3),activation="relu",padding="same"))
model.add(keras.layers.MaxPooling2D(3,3))

model.add(keras.layers.Conv2D(256,(3,3),activation="relu",padding="same"))
model.add(keras.layers.Conv2D(256,(3,3),activation="relu",padding="same"))

model.add(keras.layers.Conv2D(512,(5,5),activation="relu",padding="same"))
model.add(keras.layers.Conv2D(512,(5,5),activation="relu",padding="same"))

model.add(keras.layers.Flatten())

model.add(keras.layers.Dense(1568,activation="relu"))
model.add(keras.layers.Dropout(0.5))

model.add(keras.layers.Dense(38,activation="softmax"))

opt = keras.optimizers.Adam(learning_rate=0.0001)
model.compile(optimizer=opt,loss="sparse_categorical_crossentropy",metrics=['accuracy'])
model.summary()
```

Fig. 7.3 CNN implementation

Step 5: Model Summary

As mentioned above, we have made use of the CNN model for building and evaluating the model. This is being compared with two other models namely, ResNet and Xception model along with DenseNet121 model to check which one amongst the three models predicts the best results with highest accuracy.

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 256, 256, 32)      896
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 128, 128, 32)      0
_________________________________________________________________
dropout_1 (Dropout)          (None, 128, 128, 32)      0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 128, 128, 64)      18496
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 128, 128, 64)      36928
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 64, 64, 64)        0
_________________________________________________________________
dropout_2 (Dropout)          (None, 64, 64, 64)        0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 64, 64, 128)       73856
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 64, 64, 128)       147584
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 32, 32, 128)       0
_________________________________________________________________
dropout_3 (Dropout)          (None, 32, 32, 128)       0
_________________________________________________________________
flatten_1 (Flatten)          (None, 131072)            0
_________________________________________________________________
dense_1 (Dense)              (None, 1024)              134218752
_________________________________________________________________
dropout_4 (Dropout)          (None, 1024)              0
_________________________________________________________________
dense_2 (Dense)              (None, 38)                38950
=================================================================
Total params: 134,535,462
Trainable params: 134,535,462
Non-trainable params: 0
```

Fig. 7.3 model summary of CNN

Step 6: Training the neural network.

The model is being training with images that is being separated into two namely, training and validation data.

```
history = classifier.fit_generator(
    aug.flow(xTrain, yTrain, batch_size=BS),
    validation_data=(xTest, yTest),
    steps_per_epoch=len(xTrain) // BS,
    epochs=EPOCHS)
```

Fig. 7.4 Training the Neural Network

Step 7: Results

The results that is provided after executing the CNN model is depicted below.

Epoch	Loss	Accuracy	MSE	Validation Loss	Validation Accuracy	Validation MSE
1	0.1234	0.9727	0.0265	0.1217	0.9735	0.0258
2	0.1217	0.9736	0.0257	0.1218	0.9736	0.0257
3	0.1217	0.9736	0.0257	0.1218	0.9736	0.0257
4	0.1217	0.9736	0.0257	0.1218	0.9736	0.0257
5	0.1217	0.9737	0.0257	0.1218	0.9737	0.0257
6	0.1217	0.9737	0.0256	0.1218	0.9737	0.0256
7	0.1217	0.9737	0.0256	0.1218	0.9737	0.0256
8	0.1217	0.9737	0.0256	0.1219	0.9737	0.0256
9	0.1217	0.9737	0.0256	0.1219	0.9737	0.0256
10	0.1217	0.9737	0.0256	0.1219	0.9737	0.0256
11	0.1217	0.9737	0.0256	0.1218	0.9737	0.0256
12	0.1217	0.9736	0.0256	0.1219	0.9736	0.0256
13	0.1217	0.9736	0.0256	0.1219	0.9736	0.0256
14	0.1217	0.9736	0.0256	0.1219	0.9736	0.0256
15	0.1217	0.9736	0.0256	0.1219	0.9736	0.0256
16	0.1217	0.9736	0.0256	0.1219	0.9736	0.0256
17	0.1217	0.9736	0.0256	0.1219	0.9736	0.0256
18	0.1217	0.9736	0.0256	0.1219	0.9736	0.0256
19	0.1217	0.9736	0.0256	0.1219	0.9736	0.0256
20	0.1217	0.9736	0.0256	0.1219	0.9736	0.0256

Fig. 7.5 Result values after CNN code implementation

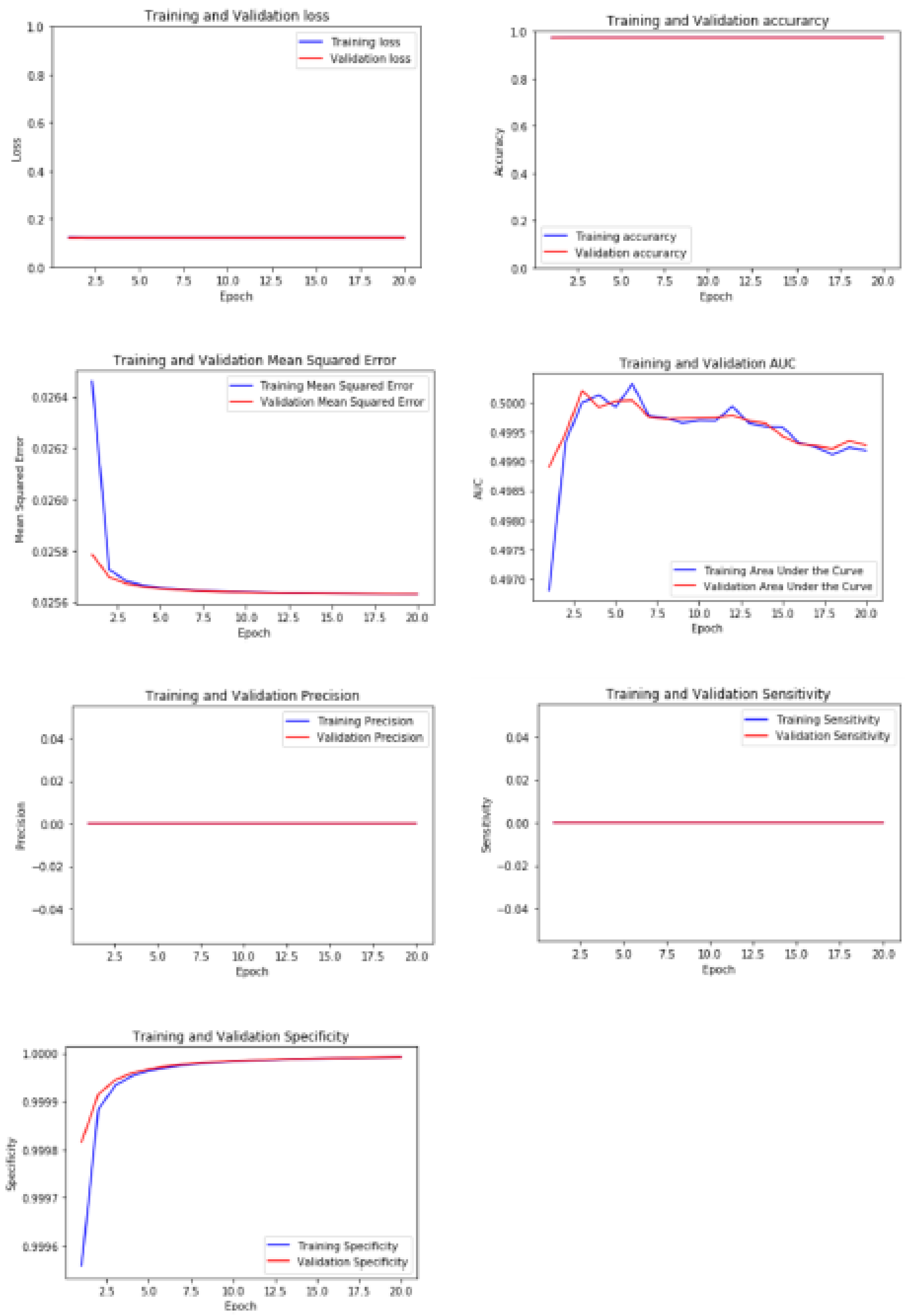

Fig. 7.6 Graphs of Accuracy, Loss, MSE and its other components

8. SECTION 2

METHODOLOGY

As mentioned, we will be comparing 2 models. In the previous section we had discussed about the working of the CNN model. In this section, let us go through the steps followed by the Xception model and DenseNet121 model.

The term "extreme inception" refers to Xception, which takes the principles of Inception to an extreme. Inception applied distinct filters to each depth space derived from the various input spaces and compressed the initial input using 1x1 convolutions. Xception simply reverses this step. Instead, it compresses the input space by applying 1X1 convolution across the depth after filtering each depth map first.

DenseNet is a CNN in every layer communicates with the network's deeper layers. It is made up of two important blocks in addition to the pooling and convolutional layers that are fundamental to the system. These are Transition and Dense Block layers.

Step 1: Model Building

```
Downloading data from https://github.com/fchollet/deep-learning-models/releases/download/v0.4/xception_weights_tf_dim_ordering_tf_kernels_notop.h5
83689472/83683744 [==============================] - 1s 0us/step
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
xception (Model)             (None, 16, 16, 2048)      20861480
_________________________________________________________________
global_average_pooling2d (Gl (None, 2048)              0
_________________________________________________________________
dense (Dense)                (None, 4)                 8196
=================================================================
Total params: 20,869,676
Trainable params: 20,815,148
Non-trainable params: 54,528
_________________________________________________________________
```

Fig. 8.1 Model Summary of Xception Model

```
Downloading data from https://github.com/keras-team/keras-applications/releases/download/densenet/densenet121_weights_tf_dim_ordering_tf_kernels_notop.h5
29089792/29084464 [==============================] - 0s 0us/step
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
densenet121 (Model)          (None, 16, 16, 1024)      7037504
_________________________________________________________________
global_average_pooling2d_1 ( (None, 1024)              0
_________________________________________________________________
dense_1 (Dense)              (None, 4)                 4100
=================================================================
Total params: 7,041,604
Trainable params: 6,957,956
Non-trainable params: 83,648
_________________________________________________________________
```

Fig. 8.2 Model Summary of DenseNet121 Model

Step 2: Ensembling the model

```
Model: "model"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_3 (InputLayer)            [(None, 512, 512, 3) 0

__________________________________________________________________________________________________
sequential_1 (Sequential)       (None, 4)            7041604     input_3[0][0]

__________________________________________________________________________________________________
sequential (Sequential)         (None, 4)            20869676    input_3[0][0]

__________________________________________________________________________________________________
average (Average)               (None, 4)            0           sequential_1[1][0]
                                                                 sequential[1][0]
==================================================================================================
Total params: 27,911,280
Trainable params: 27,773,104
Non-trainable params: 138,176

__________________________________________________________________________________________________
```

Fig. 8.3 Model Summary after ensembling the models

Step 2: Evaluation

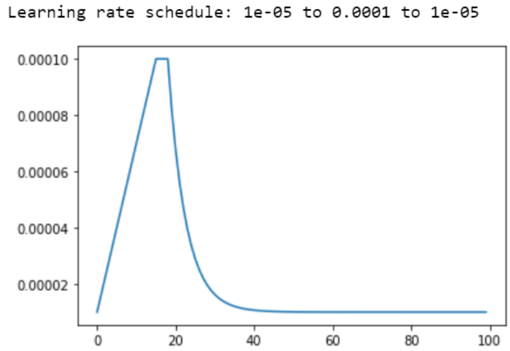

Fig. 8.4 Learning Rate of the model

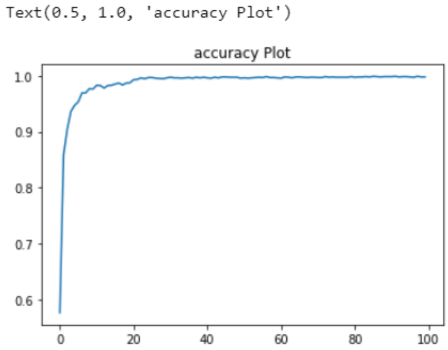

Fig. 8.5 Accuracy of the models

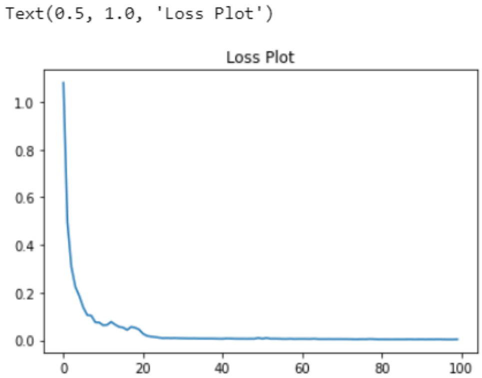

Fig. 8.6 Loss plot of the models

The model is being trained with images from two datasets along with randomly clicked images of plant leaves. After the model is being trained with these images for both the models i.e., Xception model and DenseNet121 model. Evaluation is represented in the form of graphs as shown above and these are some of the evaluation outputs for the models.

Step 3: Testing

The model is being trained and evaluated with 2 datas i.e., Training and Validation dataset. The model is executed with the testing and submission data before saving the model which helps the model to execute with other images that is absent in the datasets with which the model is trained.

	image_id
0	Test_0
1	Test_1
2	Test_2
3	Test_3
4	Test_4
...	...
1816	Test_1816
1817	Test_1817
1818	Test_1818
1819	Test_1819
1820	Test_1820

1821 rows × 1 columns

Fig. 8.7 Testing.csv

Step 4: Submission/Saving the model

The model is saved along with the information of the previous method results.

	image_id	healthy	multiple_diseases	rust	scab
0	Test_0	1.293066e-07	4.960460e-07	9.999994e-01	1.957720e-08
1	Test_1	2.531218e-06	2.072042e-06	9.999950e-01	3.478465e-07
2	Test_2	1.343778e-09	1.222801e-08	3.227613e-08	9.999999e-01
3	Test_3	9.999887e-01	6.636069e-07	8.456799e-06	2.162048e-06
4	Test_4	6.187040e-09	1.515457e-08	9.999974e-01	2.641845e-06
...	...	...	...	...	...
1816	Test_1816	1.256253e-09	6.237798e-06	9.999937e-01	3.099693e-09
1817	Test_1817	5.721491e-06	8.123470e-06	1.602107e-05	9.999701e-01
1818	Test_1818	6.402610e-08	1.077220e-07	9.999998e-01	8.000031e-09
1819	Test_1819	9.999983e-01	5.557402e-08	4.996133e-07	1.152112e-06
1820	Test_1820	1.818624e-06	9.468650e-05	3.533049e-06	9.999000e-01

1821 rows × 5 columns

Fig. 8.9 Submission.csv

Step 4: Prediction

The prediction after the code is being implemented is shown in an application names Streamlit that is a web application through which the images are browsed in the local system and the model predicts the results.

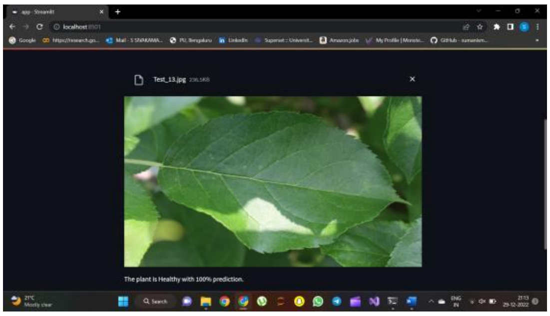

Fig. 8.10 Healthy Image Prediction

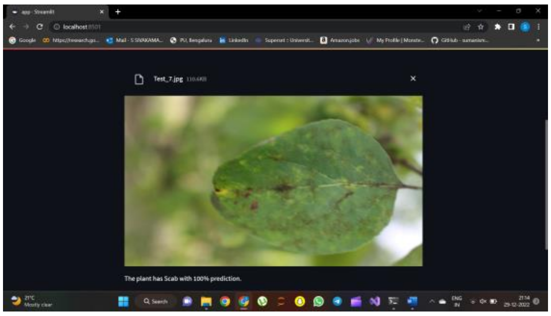

Fig. 8.11 Scab Image Prediction

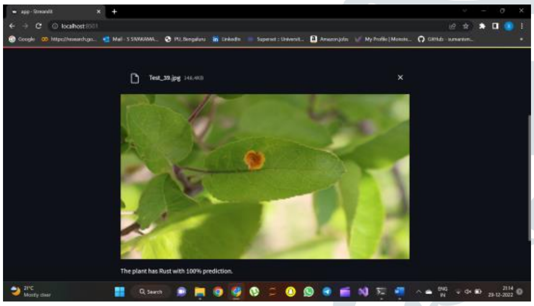

Fig. 8.12 Rust Image Prediction

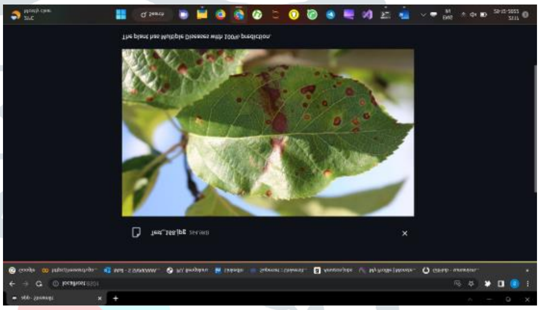

Fig. 8.13 Multiple Diseases Prediction

The model reads images which has RCG characteristics with pixel values that falls between the range 256*256. The images below this pixel range is read in a blurred form which is inappropriate for predicting the disease face by the plant leaf.

9. CONCLUSION

Crop protection in essential cultivation is not an accessible duty. This requires thorough ability of the crop being grown and potential pests, pathogens, and weeds. DL methods have been widely applied to the detection and classification of plant diseases. Solves or partially solves problems with traditional machine learning techniques. Based on a certain architectural convolutional network, a special deep learning model was developed to analyse plant diseases from images of active or infected plant leaves. Results of the advised system show that the CNN classifier perceive other diseases with high certainty.

SERIAL NUMBER	METHODS	ACCURACY PERCENTAGE
1	CNN	96.5%
2	RESNET	99.2%
3	XCEPTION AND DENSENET121	100%

10. REFERENCES

[1] K.Padmavathi, and K.Thangadurai, “Implementation of RGB and Gray scale images in plant leaves disease detection – comparative study”, 2016, Indian Journal of Science and Technology

[2] Santhosh Kumar S, and B. K. Raghavendra “Diseases Detection of Various Plant Leaf Using Image Processing Techniques: A Review”, 2019, 5th International Conference on Advanced Computing & Communication Systems (ICACCS)

[3] I. Lenz, H. Lee, and A. Saxena, “Deep learning for detecting robotic grasps,” The International Journal of Robotics Research, vol. 34, no. 4-5, pp. 705–724, 2015.

[4] B. Alipanahi, A. Delong, M. T. Weirauch, and B. J. Frey, “Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning,” Nature Biotechnology, 2015.

[5] L. Zhang, G.-S. Xia, T.Wu, L. Lin, and X. C. Tai, “Deep learning for remote sensing image understanding,” Journal of Sensors, 2016.

[6] J. Arevalo, F. A. Gonzalez, R. Ramos-Pollan, J. L. Oliveira, and M. A. G. Lopez, “Convolutional neural networks for mammography mass lesion classification,” in Proceedings of the 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC ’15), August 2015.

[7] Prashanthi, V. & Devik, P. & Kanakala, Srinivas, “Identification of opportunities for coding in a network”, International Journal of Recent Technology and Engineering, 2019.

[8] Prashanthi, V., D. Suresh Babu, and CV Guru Rao, "Network Coding aware Routing for Efficient Communication in Mobile Ad-hoc Networks”, International Journal of Engineering & Technology(UAE), 2018.

[9] Kanakala, S., Ananthula, V. R., & Vempaty, P, "Energy-efficient cluster based routing protocol in mobile ad hoc networks using network coding", Journal of Computer Networks and Communications, 2014.

[10] K.Srinivas, A.Nagaraju, S.Ramachandram and G.Narsimha, "Performance Evaluation of Routing Protocols in Static and Dynamic Ad-hoc Networks based on Energy Consumption", International Conference on Information Technology for Real World Problems, 2010.

[11] Lakshmi, K. Prasanna, and C. R. K. Reddy. "A survey on different trends in data streams." 2010 International Conference on Networking and Information Technology, IEEE, 2010.

[12] Dhanalaxmi, B., G. Apparao Naidu, and K. Anuradha, "Adaptive PSO based association rule mining technique for software defect classification using ANN", Procedia Computer Science 46, 2015.

[13] Prashanthi, V. & Kanakala, Srinivas " Comprehensive data analysis of aarogyasri scheme progress" in Telangana. Journal of Advanced Research in Dynamical and Control Systems, 2018.

[14] V.Prashanthi, Srinivas Kanakala," Generating analytics from web log ", International Journal of Engineering and Advanced Technology, 2020.

[15] P. Santhi , N. Deeban , N. Jeyapunitha , B. Muthukumaran and R. Ravikumar, "Prediction of Diabetes using Neural Networks" International Journal of Advanced Trends in Computer Science and Engineering, 2020.

[16] KollaBhanu Prakash , Lakshmi Kalyani. N, Pradeep Kumar Vadla and Naga Pawan YVR, "Analysis of Mammography for Identifying Cancer Cells using Convolution Neural Networks, International Journal of Advanced Trends in Computer Science and Engineering, 2020.

[17] Wallelign, Serawork, Mihai Polceanu, and Cedric Buche. "Soybean plant disease identification using convolutional neural network." The thirty-first international flairs conference, 2018.

[18] Ramcharan, Amanda, et al. "Deep learning for image-based cassava disease detection." Frontiers in plant science 8, 2017.

[19] Bolanle F. Oladejo, Oladejo Olajide Ademola, "Automated Classification of Banana leaf disease using an Optimized Capsule Network Model", Department of Computer Science, University of Ibadan, Ibadan, Nigeria, pp. 119-130, CSCP 2020.

[20]. Jeon, Wang-Su, and Sang-Yong Rhee. "Plant leaf recognition using a convolution neural network." International Journal of Fuzzy Logic and Intelligent Systems 17.1 (2017): 26-34.

[21]. Srdjan Sladojevic, Marko Arsenovic, Andras Anderla, Dubravko Culibrk and Darko Stefa-novic, "Deep Neural Networks Based Recognition of Plant Diseases by Leaf Image Classification", Computational Intelligence and Neuroscience, Article ID 3289801, 2016.

[22]. Sharada P. Mohanty, David P. Hughes and Marcel Salathe, " Using Deep Learning for Image- Based Plant Disease Detection", Frontiers in Plant Science, Vol. 7, Article 1419, 2016.

[23]. Kiran R. Gavhale, Ujwalla Gawande and Kamal O. Hajari, "Unhealthy region of citrus leaf detection using image processing techniques", IEEE International Conference on Convergence of Technology (I2CT), Pune 2014, pp. 1-6.

[24]. Pranjali B. Padol; Anjali A. Yadav, "SVM Classifier Based Grape Leaf Disease Detection", IEEE Conference on Advances in Signal Processing (CASP), Pune 2016, pp. 175-179.

[25]. Ahmad Nor Ikhwan Masazhar and Mahanijah Md Kamal, "Digital Image Processing Technique for Palm Oil Leaf Disease Detection using Multiclass SVM", IEEE 4th International Conference on Smart Instrumentation, Measurement and Applications (ICSIMA), Malaysia 2017, pp. 1-6.

[26]. Monzurul Islam, Anh Dinh and Khan Wahid, “Detection of potato Diseases Using Image Segmentation and Multiclass Support Vector Machine”, IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE), Canada 2017, pp. 1-4.

[27]. Nithesh Agarwal, Jyothi Singhai and Dheeraj K. Agarwal, “Grape Leaf Disease Detection and Classification Using Multi- Class Support Vector Machine”, proceeding of IEEE International conference on Recent Innovations in Signal Processing and Embedded Systems (RISE), Bhopal 2017, pp. 238-244.

[28]. Md. Selim Hossain, Rokeya Mumtahana Mou, Mohammed Mahedi Hasan, Sajib Chakraborty and M. Abdur Razzak, “Recognition and Detection of Tea Leaf’s Diseases Using Support Vector Machine”, IEEE 14th International Colloquium on Signal Processing & its Applications (CSPA), Malaysia 2018, pp. 150-154.

[29] Sukhvir Kaur , Shreelekha Pandey and Shivani Goel, “Semi-automatic leaf disease detection and classification system for soybean culture”, journal on IET Image processing, Vol. 12, Issue 6, 2018, pp. 1038- 1048.

[30]. Dheeb Al Bashish, Malik Braik and Sulieman Bani- Ahmad, “A Framework for Detection and Classification of Plant Leaf and Stem Diseases”, IEEE International Conference on Signal and Image Processing (ICSIP), Chennai 2010, pp. 113-118.

[31] Keyvan Asefpour Vakilian and Jafar Massah, “An artificial neural network approach to identify fungal diseases of cucumber (Cucumis sativus L.) Plants using digital image processing”, Archives of Phytopathologyand Plant Protection, Vol. 46, Issue 13, Taylor & Francis 2013, pp. 1580-1588.

[32] Ramakrishnan M. and Sahaya Anselin Nisha A., “Groundnut Leaf Disease Detection and Classification by using Back Probagation Algorithm”. IEEE International Conference on Communications and Signal Processing (ICCSP), Melmaruvathur 2015, pp. 0964 – 0968.

[33] Rashmi Pawar and Ambaji Jadhav, “Pomegranate Disease Detection and classification”, IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI), Chennai 2017, pp. 2475-2479.

[34] Umapathy Eaganathan, Jothi Sophia, Vinukumar Lackose, Feroze Jacob Benjamin, “Identification of Sugarcane Leaf Scorch Disease using K-means Clustering Segmentation and KNN based Classification”, International Journal of Advances in Computer Science and Technology (IJACST), Vol. 3, No. 12, Special Issue of ICCEeT, Dubai , 2014, pp. 11- 16.

[35] Aditya Parikh, Mehul S. Raval, Chandrasinh Parmar and Sanjay Chaudhry, “Disease Detection and Severity Estimation in Cotton Plant from Unconstrained Images”, IEEE International Conference on Data Science and Advanced Analytic, Canada 2016, pp. 594- 601.

[36] Gautham Kaushal, Rajini Bala, “GLCM and KNN based Algorithm for Plant Disease Detec-tion”, International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, Vol. 6, Issue 7, 2017, pp. 5845-5852.