

Agribot-Plant Disease Predictor

M. Thanjaivadivel^a, and Dr.R.Suguna^b

^a

Research Scholar, Department of Computer Science and Engineering, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai, Tamilnadu, India

^bProfessor, Department of Computer Science and Engineering, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai, Tamilnadu, India

Article History: Received: 10 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 20 April 2021

Abstract: India is a diversified country, of that more than 70 % of the population's main income depends on agriculture and farming. The large range of diversified crops provides farmers with an opportunity to select from and find suitable pesticides for the plants[11]. But the various plant diseases will cause a significant reduction in the quality and productivity of agricultural products. Plant productivity, health, and disease monitoring play an important role in agriculture and farming. The advanced image processing techniques have made it possible to keep these plant diseases in check. The major plant disease symptoms can be detected in the various plant parts such as leaves, fruits, and stems but the majority of these plant disease symptoms are considered to be detected from the leaves. These may occur due to a change in climate and pollution. To extend a helping hand for farmers in making the identification of diseased plants easier, we have come up with an autonomous robot 'AGRIBOT' where it tends to move around the farm and captures the image of the plant. The image is captured when a command is given to the pi. Though there are many researches working related to plant disease detection, the use of 'AGRIBOT' makes us unique as it goes around the field and detects the disease. In this paper, we propose an intelligent deep learning approach to identify plant disease, which have been used or investigated for estimating or measuring disease severity. With the help of Convolutional Neural Networks (CNN) and VGG16 Architecture, it can identify 38 different plant disease classes. The accuracy of the proposed increased to 99.44. After classifying healthy and unhealthy plants. Nowadays, they used to refer the chemicals to protect from plant diseases. We turn to the most benign and natural forms of control.

Keywords: Plant Disease, CNN, VGG16, ROBOT.

1. Introduction

According to FAO world agriculture statistics in 2014, India is the second-largest producer for major food staples like wheat, rice, etc., In India 16%GDP and 10% of export earnings from agricultural sectors. Most of the disease in plants is affected by an infectious organism (bacteria, viruses, fungi, etc.,) and psychological factors. The major economic growth and societies depend on the agriculture sector. Due to disease, insects, weeds, etc., cause 25% of crop failure. Therefore we required continuous surveillance of plants to detect the plant disease at an early stage. Visual or manual detection of plant disease is a tough job[10]. Nowadays, many researchers working in various machine learning techniques and proposed a unique solution to yield plant productivity. Each technique has its own advantages and drawbacks.

There are numerous machine learning approaches used to identify and differentiate the plant disease. The most common machine learning algorithms is KNN, regression (linear/logistic), CNN, decision tree, SVM, naïve bayes ANN, etc., these methods are integrated different image preprocessing methods to enhance feature extraction. Mostly knn model used for classification and regression. It is a type of supervised learning. It doesn't have any learning[5]. It predict the value between N factors or more similar like an image search engine. The nearest neighbor takes an input as test data and compare with each one of the trained data[1]. The same classes are identified and grouped separately. The distance will be calculated using there factors.

However, human expertise is not available in all over the world, there may be some farmer have less knowledge about the diagnosis. In such a case an automatic recognition system for identifying plant disease will be really useful for the farmers in the developing countries [2] [4]. Here are some of the examples and images of diseases in tomato species.

Bacterial Diseases Fig. [1] caused by the *Xanthomonas vesicatoria*, a bacterium which causes damage to leaves and fruits by developing irregular spots. This bacteria doesn't allow the fruits to ripe, leads to reduction in the yield.



Fig. 1.Bacterial spot



Fig. 2.Bacterial wilt



Fig. 3.Tomato pith necrosis

The Fig. [2] also referred as Southern bacterial blight, caused by a bacterium named *Ralstoniasolanacearum*. This survives in soil for a long period and slowly enters the roots through the pores meant water absorption, leading to stem weakness. The Fig. [3] caused by soil borne *Pseudomonas* bacteria. This causes stems to split, crack, shrink and cuts supplying of water to stems[3].

Fungal diseases

The Fig. [4] are the most common fungal diseases infected to tomato species. Spots on leaf are appeared. Among this late blight causes serious danger to plants by spreading the infection to the plants surrounding.



Fig. 4. Early blight or late blight leaf spot

Viral diseases

Fig. [5] has a tiny insect called thrips spread this virus by feeding on weed. These form yellow spots on leaves and fruits. Older plants can resist but the younger plants start dying with wilts. Fig. [6] Whiteflies transmit this virus by eating infected weeds nearby. Up to 2-3 weeks the plant doesn't show any symptoms. Slowly the plant doesn't bear any fruit and the leaves start folding upwards.



Fig. 5. Tomato spotted wilt virus



Fig. 6. Tomato yellow leaf curl virus

Developing a self-directing machine which moves in and around the field with a camera connected on its top for capturing diseased leaves is a unique idea for monitoring the agricultural field. This notable idea made us to create our AGRIBOT Fig. [7]. An android application Bluetooth Terminal is used for controlling the movement of bot. when a target plant is insight it is stopped for a while a captures. The raspberry camera module is fixed at the top of the AGRIBOT. It captures the image of the plant and sends to jetsonnano kit where the trained images are stored. The AGRIBOT moves with the control of Arduino board where coding for movement has been installed in Arduino board. The plant diseases can be diagnosed with the help of developing accurate image classifiers and using a verified large dataset of images of the various diseased and healthy plants. These various dataset of images can be stored using NumPy as a NumPy array. The NumPy library helps in performing different image processing techniques such as the operation of the acquisition and rewriting of pixel values. A machine learning algorithm K nearest neighbour and RELU, which is used to solve both classification and regression problems



Fig. 7. AGRIBOT

2. Functioning Of Agribot

Images of 38 different classes like tomato leaves, pepper leaves, apple etc., were taken from Plant Village datasets. The dataset is compressed of nearly 54303 images with different species and diseases of 38 different classes plant. Initially all these images are trained by using CNN. As the raspberry pi camera captures the image it is sent to undergo preprocessing in this the captured is pixelated or cropped accordingly and starts applying the predictive algorithm of deep learning technique. After comparing with the trained data of tomato leaves, it displays an output of the captured image showing the status of the plant and the disease infected with and some proper remedies to avoid.

The plant diseases can be diagnosed with the help of classifiers. These various datasets of images can be stored using NumPy as a NumPy array. The NumPy library helps in performing different image processing techniques such as the operation of the acquisition and rewriting of pixel values. A machine learning algorithm K nearest neighbour and RELU, used to solve classification and regression problems.

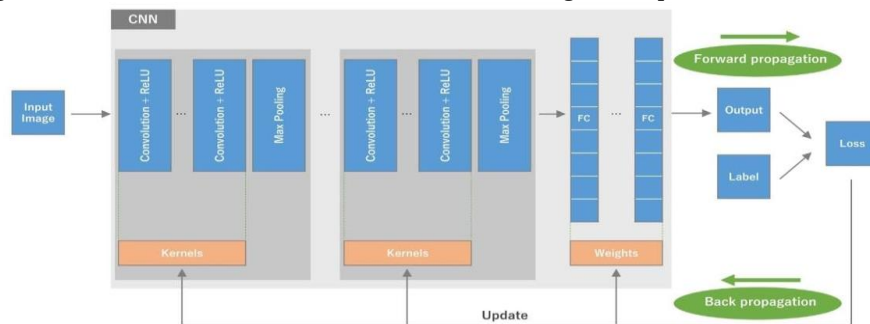


Fig. 8. CNN architecture

The CNN architecture Fig. [8] is inspired by framework of human brain. It is designed in a pattern how the neurons within the human brain are connected. A perception is a neural network unit within a CNN which is of a multilayer hierarchy. Each set of neurons analyze the features in the small of the image. Each group of neurons is specified in identifying part of the image. CNN use predictive algorithms to classify under different categories. It takes the input image in the form of pixels which depend on the image resolution. A Rectified Linear Unit acts as a nonlinear activation function. ReLU function rounds the value up to zero when it is less than 0 i.e. negative. The output comes as $f(x) = \max(0, x)$.

CNN is embedded with several layers:

Conv layer: This layer is the fundamental building block and the first layer which extract features from an input image. The layer consists of parameters like filters or kernels which extend through in depth of the input. The pixels from small regions are scanned by applying filters or kernels. It performs operations like stride, padding, blur, sharpening and detects the edges.

Pooling layer: the convolutional layer is followed by pooling layer that takes small rectangular units or blocks from the convolutional layer and pooling layers section would reduce the number of parameters when the images are too large. Pooling is done independently to the depth dimensions so that the depth remains unchanged. Spatial pooling is also called as sub sampling or down sampling which helps in reducing the dimensionality of each image. pooling is of different types: Max, Average and sum.

Fully connected layer: this layer consists of one or more fully connected layer per performing high level reasoning by collecting the neurons from the previous layer and connecting to the neurons in current layer so that it helps in generating global semantic information. The most popular CNN architectures are:

LeNet-5 CNN architecture is built up with 7 layers. The layer composition is of 3 convolutional layers, 2 fully connected layers and 2 sub sampling layers.

1st layer is the input layer; this is generally not considered as a layer of the network as nothing is learnt from this layer. This input layer is built to take 32x32, and these are the dimensions of images that are allowed to be passed into the next layer. The LeNet-5 architecture utilizes 2 significant types of layers:

1. convolutional layers
2. subsampling layers.

AlexNet is a similar architecture like LeNet but, deeper. This is designed by SuperVision group. It has more stacked convolutional layers as well as filters per. It is compressed of 5 convolutional layers along with 3 fully connected layers. This model was the winner of image net LSVRC in 2012. it has almost same lays and ReLU activations. The network is spitted up into 2 pipelines because it was trained on 2 Nvidia Geforce GTX 580 for 6 days and this accelerates the accuracy 6 times with a greater speed. Data augmentation, dropout and overlap pooling are the key features. This architecture is developed by Mahew Zeiler and Rob Fergus. This gives better accuracy than AlexNet. This uses 7*7 sized filters in its layers. The architecture is embedded with 3 overlapping

max pooling layers, 3 fully connected layers and 5 convolutional layers. The hidden layer uses ReLU activation. It has a capacity to store 60,000,000 trainable parameters[8].

This was proposed in 2014 by Google researchers. GoogLeNet was declared as a winner at ILSVRC. The framework is of 22 layers. This uses different pooling methods and activation functions for classifying and detecting the tasks. 1*1 convolution layer with 128 strides are used for decreasing the number of parameters. This makes the task identification efficiently with less than 5% almost as the performance of a human eye. VGGNet, referred as visual geometry group invented in Oxford University. This model is very helpful for classifying images at large scale. This architecture is helpful in the decrease of parameters while performing tasks. VGGNet is trained for 6 weeks on 4 GPU's to gain its performance and efficiency[9].

Commonly referred as residual neural network (ResNet). The main concept used in this architecture is "skipping connections". In this the ReLU activation function uses 2 linear transformations for calculating the input matrix. The final ReLU function gives the output. This reduces the error rate up to 3.6%.

The challenges faced in CNN architecture.

- Backpropagation plays a major role in the CNN models by practicing the method of fine-tuning based on the error occurred during the previous epoch. Proper back propagation leads in the reduction of error rate. But the generalization takes time. This algorithm is widely used in neural networks.
- Translation invariance refers to the similar object with a slight change in position or orientation of neuron or perception that is supposed to identify the task. The data augmentation tries to solve the problem partially but not completely. CNN fails to recognize the exact position effectively.
- Pooling layers make a very big mistake, leading to the loss of information and reduces spatial resolution. The pooling layers perform translational variance which has a poor performance but good in performing some tasks[13].

3. Materials and Methods

A three-layered convolutional neural network should be built and then we train and test data. To begin we need to proceed with a step by step process in a hierarchical fashion.

1. Training and testing data should be prepared.
2. Using the Tensorflow library build the CNN layers.
3. Select Optimizer.
4. Save the checkpoints after training the data.
5. Finally, test the model.

Testing evaluates the performance of the trained data.

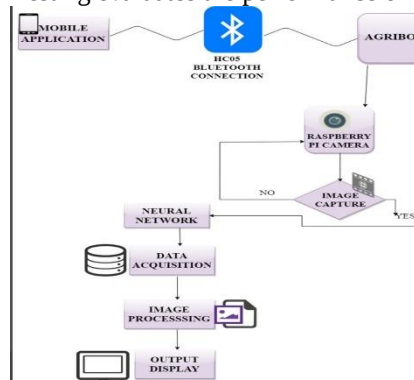


Fig. 9. Architecture diagram for AGRIBOT

In Fig. [9] Analysis takes few minutes and the output is displayed as shown along with the remedies of the disease. Detection of plant disease through some automatic machine is beneficial as it requires a large amount of work in monitoring in big farm crops. Detects accurate disease attacked to the plant. Stores database of insecticides for respective pest and disease.

Materials

The input image was collected from PlantVillage. The image consists of 38 classes of plant disease and healthy plant of 9 different crops (apple, cherry, corn, etc.) having 54303 images. In this paper, we have resized the image into 256x256 pixels. The different versions of the dataset are present in the raw directory: color, grayscale and segmented of Apple, Cherry, Grape, Peach, Pepper, Potato, Strawberry and Tomato. 8 CNN classifiers are trained to identify the diseases of each of the 9 plants of 36 classes.

Table 1. Summary of each crop and its classes

Plants	Total number of classes used
Apple	7
Cherry	2

Corn	4
Grape	4
Peach	2
Pepper	2
Potato	3
Strawberry	2
Tomato	10

Table [1] describes different classes species for a crop. The result from stage 2 is used to call the classifier that has been trained to classify the different diseases for that plant. If there are none, the leaf would be classified as Healthy. All these above plants have been trained using ResNet- 50 and VGG-16 Architecture using transfer learning from the ImageNet Weights.[14]

When we have our information assets alongside their names, we train our siamese network. From the picture pair, we feed one picture to the network A and another picture to the network B. The job of these two networks is just to extricate the element vectors. Along these lines, we utilize two convolution layers with RELU enactments for separating the highlights. When we have taken in the component, we feed the resultant element vector from both of the networks to the vitality work which quantifies the closeness, we utilize Euclidean separation as our vitality work. Along these lines, we train our network by taking care of the picture pair to gain proficiency with the semantic likeness between them[16].

Table 2. Summary of methodologies and its accuracy rate

SLN O	Type of features	DATA SET	CLAS S LABELS	DL MODEL USED	ACCURAC Y	Re f
1	Automat ic	Flavia dataset	32	CNN + RF classifier	97.3	[8]
2	Hand- Crafted features	Own dataset	15 – 13 plant disease, 2 backgroun d image	CNN	96.30%	[6]
3	Automat ic	PlantVilla ge	38	Alex Net, Google Net, CNN	0.9935	[4]
4	Hand- Crafted features	PlantVilla ge dataset. (3700)	3	CNN	96	[12]
5	Automat ic	Landsat-8 and Sentinel- 1A RS	11	CNN	94.60%	[17]
6	Automat ic	Foulum Research Center	7	VGG16 CNN	79	[7]
7	Hand- Crafted features	SENTINE L 2A	19	LSTM	76.2	[14]
8	Automat ic	Agroscope research center.	23	CNN + HistNN	0.90	[16]
9	Automat ic	LifeCLEF 2015 plant dataset	1000	AlexNet CNN	48.60	[15]

10	Hand crafted	Own dataset	2	Auto Redefined CNN	98.4	[13]
11	Automat ic	MK Leaf Dataset	44.	AlexnNet CNN	99.60	[21]
12	Automat ic	INTA Argentina.	3	CNN	96.90	[9]
13	Hand- Crafted features	TARBIL Agro- informatics Research Center of ITU	9	AlexNet CNN	87	[22]
14	Hand- Crafted features	Arabidops is thaliana.	4	CNN+ LSTM	93	[17]

In Table [2] take summary of 14 various research article, whereas 9 article discuss about automatic feature and the remaining 5 article discuss about hand crafted features.

Convolutional Networks

The term “CONVOLUTIONAL NEURAL NETWORKS” specifies that the framework implements a mathematical function called **convolution**. It is a specific kind of linear operation. That makes use of convolution instead a normal matrix multiplication not less than once in their layers. The activation function referred to Rectified Linear Unit (RELU layer) and eventually followed by additional convolutional layers like fully connected layers, pooling layers and normalization layers, these can be referred as hidden layers as the inputs and outputs are masked by the final activation and convolution function. CNN signifies one of the earliest deep neural networks hierarchy that emerges hidden layers between the input and output layers which helps the system to extract features found in a given input image. . The hidden layers compose a series of convolutional layers that combine with a dot product or multiplication. CNN, a deep learning algorithm used in classification and recognition of different features in an image. This has ability to perform different tasks like segmentation and classification of a image, detecting and analyzing an object[17].

Convolutional networks abuse spatially nearby relationship by authorizing a neighborhood network design between neurons of contiguous layers: every neuron is associated with just a little district of the information volume. The degree of this network is a hyper parameter called the open field of the neuron. The associations are nearby in space (along width and stature), yet consistently reach out along the whole profundity of the info volume[18]. Such design guarantees that the educated channels produce the most grounded reaction to a spatially neighborhood input design. The three major components used in CNN are Convolution, Pooling and output.

1. Depth: it controls the no. of neurons in the layer, the same as been connected to region of the input. All the neurons will activate in different features from the region of input.
2. Stride: it controls the spatial dimensions and how depth (width and height) are allocated.
3. Pooling is accomplished fixed shape window, it is slid over all parts in the input according to the stride [20].

VGG-16

VGG-16 [Fig. [10]] consists of deep 16 CNN layers. A. Zisserman and K. Simonyan from Oxford University initiated this VGG-16 concept in the paper “Very Deep Convolutional Networks for Large-Scale Image Recognition”. This was a solitarily notable model presented to ILSVR in 2014 where it won 1st and 2nd place in this challenge[19]. VGG-16 was one of the greatest architecture model up to now. The most distinctive concept regarding VGG-16 is that rather than containing more quantity by specifications that resolved for getting convolutional zones containing three multiplied three set of learnable weights in addition to step 1. VGG-16 model has 92.7% of first five accuracy test with data present in ImageNet containing images of about fourteen milliard that belongs to thousand class. Model size is 528 MB.

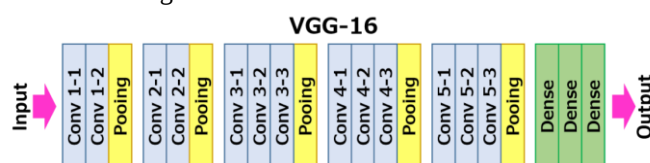


Fig. 10. VGG Architecture

This model is built using

- Max pooling layers (2*2 size)
- Convolutional layers (3*3 size)

- 16 layers
- Fully connected layers.

ImageNet is an image dataset which are systemized as stated by WorldNet hierarchy containing over 15 million great quality pictures which belongs till twenty two thousand classifications. The image collection tool of amazon company was utilized to fetch the images from the web and human classifiers. ILSVRC utilizes a subcategory of when image data accompanied by almost thousand pictures in every 1000 classification. Overall, it contains about one point two million pictures undergoing training, 50,000 recognized images and 1,50,000 images that are tested. ImageNet is embodied with variable-resolution images. These images have a pre-determined resolution by 256x256. When an image of rectangular size is provided, the provided image is scaled again and the size is cropped at the central 256x256 patch in emerging image. The image with dimensions (224,224,3) is given as the input to the network. The input image is proceeded by passing across a drift of convolutional layers, by utilizing the learnable weight(filter) containing least receiving area by three multiplied three (least dimension to represent concept containing right/left, centre, up&down). In this provided architecture image, the primary two layers consist of sixty-four techniques containing three multiplied three size learnable weight along with identical layer addition. Subsequently the maximum pooling of step 2,2, the 2 zones containing convolutional zones by two hundred and fifty six dimension filter and also the weight dimension of 3,3. It succeeds another maximum pool step 2,2 similar to preceding layer. Then 2 convolutional zones containing dimension of 3,3 and two hundred and fifty-six learnable weight is present. Next to this layer is a layer consisting of two block with three convolutional zones also single maximum pooling zone.

Every single layer in this consists of five hundred an twelve weight with 3,3 dimension with identical layer addition. That picture passes through the collection of 2 convolutional zones. Containing few zones, 1*1 pixel used for manipulating the current number of input channels is used. It consists of layer addition of pixel 1 (identical) completed aftermath every convolutional zone to avoid commutative characteristic per picture

3 FC (fully connected) layers go along with a stack of convolution layers where the first 2 zones contain four hundred and ninety-six channels for each and the 3rd zone executes thousand methods of ImageNet Large Scale Visual Recognition Challenge categorization also containing thousand techniques. Last zone contains the layout of the fully connected layers is identical in all networks.

Every single concealed zone is provided beyond ReLU. It is identified that not a single network consists of Local Response Normalization which paves way for increase in storage consumption and computation time but no improvement in performance. The below Table [3] lists various VGG architecture. The below table contains two kind of this model. Those 2 versions consists of hundred and thirty four million and hundred and thirty eight million specifications accordingly. The convolutional layers are separated into five groups and every group is accompanied by a max-pooling layer[21].

Table 3. Different VGG configuration

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 LRN	conv3-64	conv3-64	conv3-64	conv3-64
maxpool					
conv3-128	conv3-128	conv3-128	conv3-128	conv3-128	conv3-128
maxpool					
conv3-256	conv3-256	conv3-256	conv3-256	conv3-256	conv3-256
conv3-256	conv3-256	conv3-256	conv3-256	conv3-256	conv3-256
maxpool					
conv3-512	conv3-512	conv3-512	conv3-512	conv3-512	conv3-512
conv3-512	conv3-512	conv3-512	conv3-512	conv3-512	conv3-512
maxpool					
conv3-512	conv3-512	conv3-512	conv3-512	conv3-512	conv3-512
conv3-512	conv3-512	conv3-512	conv3-512	conv3-512	conv3-512
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Identification is the process of finding a particular object or a specification in a image, explained by a bounding box. It mainly predicts the categories. A bounding box place is recognized by 4-D vector. Bounding box shared between various candidates and the bounding box is class determined is the main two sort of localization architecture. This work is tested including all two approaches on its design. In this, modification of losing change against categorization losing to reversion losing outcome which condemns this diversion by anticipated losing of rightness is necessary to be done by us.

It remarkably outstands various precedent generation models present in ILSVRC-2012 and ILSVRC-2013 competitions. VGG-16 proved to be very competitive compared to GoogleNet as for the classification task

winner. It came in second place in the categorization function containing flaw with seven-point three two percent which was barely a step beyond from its competitor containing flaw of 6.66%. In the localization task VGG-16 was winner containing 25.32% localization error. The challenges face in VGG16. It takes a lot of time to be trained. The size of trained VGG-16 ImageNet weighs 528MB. Therefore, it occupies a lot of memory along with frequency which becomes ineffectual[21].

SVM

In the last two decades, several types of research are carried out to detect disease affected in the plant using the Image processing technique. By using this technique, We extract the feature and feed the inputs to the machine learning algorithms to produce an accurate classification. In [Semary et al] this approach they have been used color and texture features using SVM Classifier. They have been tested SVM with 4 kernels [Linear, Quadratic, RBF and MLP] using 2 normalization methods [Min-max and Z Score] were compared with 70% training and 30% testing samples produced accuracy of 92%. Prasad et al. proposed an automated leaf diagnosis using GWF. In this approach, through mobile devices capture the image and preprocess the leaf and then computational task will be performed with help of: GWT– GLCM feature extraction and k-Nearest Neighbor classification. It produces an accuracy rate of 93%. H. Sabrol,et.al Tomato plant disease classification in digital images using classification tree deployed Ostu's Method to segment the leaf images. Then finally classifying the output using decision tree with an accuracy rate of 97.3%. P.B. Padol et.al proposed, the diseased region identified using K-means clustering, then color and texture are extracted from that we examined the disease with accuracy rate of 88.89%. 12 leaf features which are extracted and orthogonalized into 5 principal variables are given as input vector to the SVM. Classifier tested with flavia dataset and a real dataset and compared with k-NN approach, the proposed approach produces very high accuracy of [Flavia – 78 (KNN) 94.5 (SVM)], [Real dataset – 81(KNN) 96.8(SVM)] and takes very less execution time.

1. AGRIBOT User Interface

Step 1 : User Interface to load the image

Step2 :Analyze the disease.

Step 3: Build the Model for Classifying Healthy or Unhealthy using Keras and Tensorflow

Step 4: Remedial action for plant disease

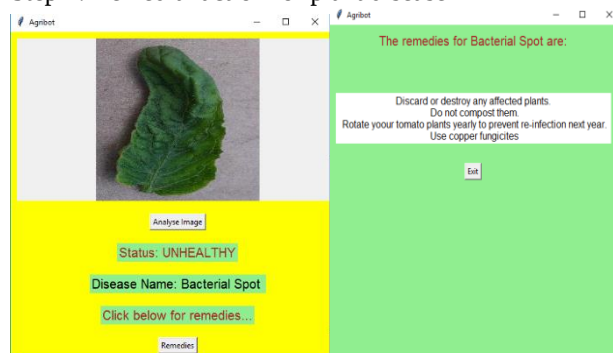


Fig. 11. User interface for AGRIBOT and its remedial action for disease

In the Fig. [11] user interface has been created to capture the image automatically. Latter, it has been diagnosed using CNN. Finally designed estimations and models to see species and infirmities in the yield leaves by using cnn. We used an jupyter notebook to test the image result. Then it has been automated using Python source code. Downloaded public dataset of 54000 images of 17 basic disease, 4 bacterial , 2 viral disease and one disease affected by mist from plantvillage Dataset. In this we loaded a map from category label to category name. Which is helpful to map the integer encoded categories in to the actual names of the species and its disease. The train model using validation dataset for validate each epochs Table [4][22]. Initially, get an accuracy of 96%, we can improve more than 99% using fine tuning. In Fig. [12] clearly shows the after 25thepochs we get an accuracy closer to 99.33%.

Table 4. Train model using validation dataset for validate each epochs

Epoch	Trainloss	Validloss	Acc	Error	Time
0	0.275081	0.137157	0.956834	0.043166	03:43
1	0.146267	0.064265	0.978372	0.021628	03:28
2	0.097072	0.037028	0.987368	0.012632	03:28
3	0.060917	0.027360	0.990549	0.009451	03:27
4	0.042702	0.025219	0.991094	0.008906	03:28

```

Epoch 16/25
97/97 [=====] - 796s 8s/step - loss: 0.0289 - accuracy: 0.9897 - val_loss: 0.0630 - val_accuracy: 0.9845
Epoch 17/25
97/97 [=====] - 807s 8s/step - loss: 0.0273 - accuracy: 0.9903 - val_loss: 0.0800 - val_accuracy: 0.9822
Epoch 18/25
97/97 [=====] - 829s 9s/step - loss: 0.0256 - accuracy: 0.9909 - val_loss: 0.1230 - val_accuracy: 0.9761
Epoch 19/25
97/97 [=====] - 822s 8s/step - loss: 0.0247 - accuracy: 0.9909 - val_loss: 0.1299 - val_accuracy: 0.9765
Epoch 20/25
97/97 [=====] - 814s 8s/step - loss: 0.0243 - accuracy: 0.9913 - val_loss: 0.2022 - val_accuracy: 0.9696
Epoch 21/25
97/97 [=====] - 820s 8s/step - loss: 0.0210 - accuracy: 0.9922 - val_loss: 0.0584 - val_accuracy: 0.9862
Epoch 22/25
97/97 [=====] - 824s 8s/step - loss: 0.0198 - accuracy: 0.9926 - val_loss: 0.0557 - val_accuracy: 0.9838
Epoch 23/25
97/97 [=====] - 825s 9s/step - loss: 0.0203 - accuracy: 0.9925 - val_loss: 0.0264 - val_accuracy: 0.9916
Epoch 24/25
97/97 [=====] - 818s 8s/step - loss: 0.0197 - accuracy: 0.9925 - val_loss: 0.0481 - val_accuracy: 0.9887
Epoch 25/25
97/97 [=====] - 813s 8s/step - loss: 0.0187 - accuracy: 0.9933 - val_loss: 0.0497 - val_accuracy: 0.9875

```

Fig. 12. Training model

In Fig. [13 (a) and (b)] Comparing the accuracy and loss by plotting the graph for training and validation. Then evaluating the accuracy using evaluate method.

```

acc = history.history['accuracy']
val_acc = history.history['val_accuracy']
loss = history.history['loss']
val_loss = history.history['val_loss']
epochs = range(1, len(acc) + 1)

# Train and validation accuracy
plt.plot(epochs, acc, 'b', label='Training accuracy')
plt.plot(epochs, val_acc, 'r', label='Validation accuracy')
plt.title('Training and Validation accuracy')
plt.legend()

plt.figure()

# Train and validation Loss
plt.plot(epochs, loss, 'b', label='Training loss')
plt.plot(epochs, val_loss, 'r', label='Validation loss')
plt.title('Training and Validation loss')
plt.legend()
plt.show()

```

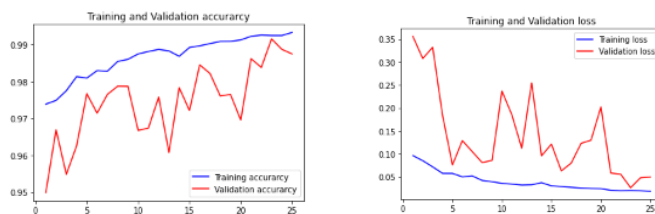


Fig. 13. (a) and (b) Accuracy and loss

Then to test the model, have written the following predict disease function to predict the class or disease of a plant image. We just need to provide the complete path to the image and it displays the image along with its prediction class or plant disease. The prediction can be improved if you change some hyper parameters.

```

SOURCE: class: Soybean_healthy, file: Soybean_healthy/dd42c35d-c510-42dc-89f4-f3f
PREDICTED: class: Soybean_healthy, confidence: 0.979864

```

```

<Figure size 432x288 with 0 Axes>
SOURCE: class: Peach_Bacterial_spot, file: Peach_Bacterial_spot/e19f5bb9-bcab-43f
PREDICTED: class: Peach_Bacterial_spot, confidence: 0.556456

```

```

<Figure size 432x288 with 0 Axes>

```

Fig. 14. Predicted species

4 Conclusion

AGRIBOT helps in getting complete info of a plant which it is examining. Compared with human eyes, it can identify plant disease with greater accuracy. It helps in lessening the manual work thereby identifying and protecting plants with the help of this robot. Since solar panel is used, renewable energy is utilized. Therefore, it is not necessary to use any electricity power. There are various methods to automate the plant disease detection and portrayal process, yet simultaneously, this investigation field is inadequate. Furthermore, there is no proper solution in the market to automate them, beside those overseeing plant species affirmation subject to the leaf's pictures. In this paper, another system of using significant learning procedure was examined to thusly arrange and recognize plant disease from leaf pictures. The ML model had the alternative to distinguish leaf proximity and perceive sound leaves and 13 unmistakable diseases, which can be apparently investigated. The complete

framework was depicted, independently, from social event the photos used for planning and endorsement to picture preprocessing and development, finally the system of setting up the significant CNN and tweaking. Different tests were performed to check the introduction of the as of late made model. Another plant affliction picture database was made, containing more than 3,000 extraordinary pictures taken from the available Internet sources and connected with more than 30,000 using fitting changes. The exploratory results achieved precision some place in the scope of 91% and 98%, for detached class tests. The last all things considered accuracy of the readied model was 98.6%. Aligning has not demonstrated significant changes in the general accuracy, anyway the amplification technique had an increasingly vital effect on achieve great result.

References

- A. Rastogi, R. Arora, S. Sharma, "Leaf disease detection and grading using computer vision technology & fuzzy logic", 2nd international conference on signal processing and integrated networks (SPIN), pp. 500-505, 2015.
2. G. Tripathi, J. Save, "An image processing and neural network based approach for detection and classification of plant leaf diseases", Journal Impact Factor, Vol. 6, No. 4, pp. 14-20, 2015.
3. S. Arivazhagan, R.N. Shebiah, S. Ananthi, S.V. Varthini, "Detection of unhealthy region of plant leaves and classification of plant leaf diseases using texture features", Agricultural Engineering International: CIGR Journal, Vol. 15, No. 1, pp. 211-217, 2013.
4. S.P. Mohanty, D.P. Hughes, M. Salathé, "Using deep learning for image-based plant disease detection", Frontiers in plant science, Vol. 7, pp. 1-10, 2016.
5. K. Muthukannan, P. Latha, R.P. Selvi, P. Nisha, "Classification of diseased plant leaves using neural network algorithms", ARPN Journal of Engineering and Applied Sciences, Vol. 10, No. 4, pp. 1913-1919, 2015.
6. S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, D. Stefanovic, "Deep neural networks based recognition of plant diseases by leaf image classification", Computational intelligence and neuroscience, pp. 1-11, 2016.
7. A.K. Mortensen, M. Dyrmann, H. Karstoft, R.N. Jørgensen, R. Gislum, "Semantic segmentation of mixed crops using deep convolutional neural network", Proc. of the International Conf. of Agricultural Engineering (CIGR), 2016.
8. D. Hall, C. McCool, F. Dayoub, N. Sunderhauf, B. Upcroft, "Evaluation of features for leaf classification in challenging conditions", IEEE Winter Conference on Applications of Computer Vision, pp. 797-804, 2015.
- B. McCool, T. Perez, B. Upcroft, "Mixtures of Lightweight Deep Convolutional Neural Networks: Applied to Agricultural Robotics", IEEE Robotics and Automation Letters, Vol. 2, No. 3, pp. 1344-1351, 2017.
9. D.A. Horneck, D.M. Sullivan, J.S. Owen, J.M. Hart, "Soil Test Interpretation Guide", Oregon State University, EC 1478-E, <http://extension.oregonstate.edu/catalog>, pp. 1-12, 2011.
10. R. Miller, "Reliability of soil and plant analyses for making nutrient recommendations", Western Nutrient Management Conference, 2013.
11. J. Amara, B. Bouaziz, A. Algergawy, "A deep learning-based approach for banana leaf diseases classification", Datenbanksysteme für Business, Technologie und Web (BTW 2017)-Workshopband, pp. 79-88, 2017.
12. S. Sameerunnisa, J. Jabez, V. Maria Anu, "The power of deep learning models: Applications", International Journal of Recent Technology and Engineering, Vol. 8, No. 2S11, pp. 3700-3705, 2019.
13. M. Rußwurm, M. Körner, "Multi-temporal land cover classification with long short-term memory neural networks", The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. 42, pp. 551-558, 2017.
14. A.K. Reyes, J.C. Caicedo, J.E. Camargo, "Fine-tuning Deep Convolutional Networks for Plant Recognition", CLEF (Working Notes), pp. 467-475, 2015.
15. J. Rebetz, H.F. Satizábal, M. Mota, D. Noll, L. Büchi, M. Wendling, B. Cannelle, A. Perez-Urbe, S. Burgos, "Augmenting a convolutional neural network with local histograms", European Symposium on artificial neural networks, Computational Intelligence and Machine Learning, pp. 515-520, 2016.
16. N. Kussul, M. Lavreniuk, S. Skakun, A. Shelestov, "Deep Learning Classification of Land Cover and Crop Types Using Remote Sensing Data", IEEE Geoscience and Remote Sensing Letters, Vol. 14, No. 5, pp. 778-782, 2017.
17. P. Sharma, Y.P.S. Berwal, W. Ghai, "Performance analysis of deep learning CNN models for disease detection in plants using image segmentation", Information Processing in Agriculture, Vol. 8, pp. 1-9, 2019.
18. J. Boulent, S. Foucher, J. Théau, P.L. St-Charles, "Convolutional neural networks for the automatic identification of plant diseases", Frontiers in plant science, 2019.

19. M. Dyrmann, A.K. Mortensen, H.S. Midtiby, R.N. Jørgensen, "Pixel-wise classification of weeds and crops in images by using a fully convolutional neural network", *Proceedings of the International Conference on Agricultural Engineering*, pp. 26-29, 2016.
20. S.H. Lee, C.S. Chan, P. Wilkin, P. Remagnino, "Deep-plant: Plant identification with convolutional neural networks", *IEEE international conference on image processing (ICIP)*, pp. 452-456, 2015.
R. Aroul Canessane, R. Dhanalakshmi, V. Maria Anu. "Implementation of Tensor Flow for Real-time Object Detection", *International Journal of Recent Technology and Engineering (IJRTE)*, Vol. 8, No. 2S11, 2019.