INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN COMPUTER SCIENCE AND ENGINEERING TECHNOLOGIES

ISSN 2454-9924

DISEASE DETECTION OF LEAFY VEGETABLES USING DEEP ALEXNET MODEL

Narasimha Reddy Boddapati [*1] **Yedukondalu Annangi**[*2], **Dr. D Uma Devi** [*3]

[*1] Gayatri Vidya Parishad College of Engineering (Autonomous), India,
narasimhareddyboddapati@gmail.com

[*2] Gayatri Vidya Parishad College of Engineering (Autonomous), India,
yedukondaluannangi@gmail.com

[*3] Associate Professor and Associate Head of B. Tech CSE with AI & ML, Ph.D
umadevi@gvpce.ac.in

Abstract: Detection of diseases in leafy vegetables and suggesting pesticides are the most challenging tasks in the real world which can be accomplished by using deep learning models. One of the essential features to find out the yield and quality of the leafy greens and detection of different diseases can be carried out by digital image processing. Nowadays deep learning technology has made many breakthroughs in image processing. In this paper a model was trained on three leafy vegetables i.e Brassica Oleracea(Cabbage), Lactuca Sativa(Lettuce) and Amaranthus using AlexNet deep learning model. The proposed model detects the diseases like Backmoth, leafminer and mildew of cabbage, amaranthus vegetable, Blight, White Rust are the diseases classified using DDLVUDAM[1]. Bacterial, Fungal and Virus diseases can be diagnosed using our model for Lettuce. This report outlines a definition of the plant diseases classification challenge as well as using this innovative disease diagnosis and pesticide suggestion methods. AlexNet is used for diagnosis of diseases in leafy vegetables. DDLVUDAM Deployment of this application is done by using Xampp server. An accuracy of 93 percent is achieved with this model, which is better than the previous existing model[4] that was given 87% accuracy.

Keywords: Deep Learning, Classification, AlexNet model.

I. Introduction

Deep Learning is a subset of machine learning that involves the use of large amounts of data. A significant amount of information is fed into multi-layered neural networks and those learn by themselves. Neural networks are intended to work like the human brain. Deep learning algorithms [1] perform computations and increments repeatedly within each layer of the neural network, gradually 'learning' and improving the accuracy of the output over time. Deep learning inhales data from multiple data sources and analyses it in real time, similar to how the human brain absorbs and processes information entering the body through the five senses.

The significance of accurate and timely pathogen detection [2], as well as early prevention, has never been greater in this changing world. Disease outbreaks can be detected in a variety of techniques. Some diseases have no visible symptoms, or others are complex to detect in an early stage and become visible too late.

Most of the diseases, on the other hand, cause some sort of symptoms. Various algorithms are currently used to diagnose the diseases in leaf vegetables and most common are Convolutional Neural Networks.

In Deep learning, CNN is an information processing paradigm that was inspired by the actual human brain working mechanism. Network is composed of a large number of neurons connected together to solve the specific problem. These neurons act like the human brain that allows the computer to recognize the specific patterns and solve the problems associated with Artificial Intelligence, Machine Learning and Deep Learning.This information processing neural network is also known as Convolutional Neural Network, inspired by the brain that mimics the actual tasks.

OpenCV [3], is generally used to read the images from the data or input images. Reading the images is a primary and fundamental step in this diagnosis

[1] DDLVUDAM - DISEASE DETECTION OF LEAFY VEGETABLES USING DEEP ALEXNET MODEL(project title)

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN COMPUTER SCIENCE AND ENGINEERING TECHNOLOGIES

ISSN 2454-9924

process.For this purpose, we use Open computer vision. By using this we can effectively read the images from the input. The common gateway interface, or CGI, is a set of standards that defines how information is exchanged between the web servers and custom script.

Convolution Neural Network contains convolutional layers, pooling layers, fully connected layers and dropout layers.

Alexnet is a CNN model that has eight layers with learnable parameters. There are 5 convolution layers in which 3 layers combined with max following. These layers are followed by three fully connected layers. All these layers use Relu activation except the output layer which uses softmax activation.

The remaining article is arranged as follows: section II refers to related work; section III refers to a proposed methodology; section IV describes the analyses of results and discussions, and finally section V provides conclusions along with a possible recommendation for the future work.

II. Related Works

In recent works, this study examines the use of deep convolutional neural networks for the detection of diseases in plants according to the features extracted [4], by the network with the uploaded images as an input, was achieved an accuracy of 87% with batch size of 12 and 100 epochs. In the crops dataset some of the images are taken as training images and some images are testing images based on the classes of the images.

In existing works, Transfer Learning with VGG16(Visual Geometric Group) is used. This model achieves least accuracy compared to the custom model means convolutional neural network implemented by our own layers. It obtains validation-accuracy of 40% with batch size of 12 with 10 epochs.

For this process we need to create a plant image dataset manually by using the available libraries such as Keras and TensorFlow [5], in the deep learning models huge collection of the green leafy vegetable images was obtained from different web sources by recreating the available images into the multiple images. This whole process is done by using deep learning frameworks [6], availed in the sources such as TensorFlow and Keras these two also known as models

By considering the above related works, we have implemented an automatic disease diagnosis system using deep learning based Alexnet [7]model. In the coming sections we will describe our proposed methodology.

Table 1 summarizes the details of some of the related works of plant disease detection.

Author & Year	Proposed	Finding/Outcomes
Surbhi Jain, Joy dip Dhar 2017	Aim to confront an advance deep learning method, known as Convolutional Neural Network (CNN), for studying feature representations and similarity measures	retrieval of similar images, we agreed on using transfer learning to apply the Google Net deep architecture to our problem. Extracting the last-but-one fully connected layer from the retraining of Google Net CNN model
Andrew G. Howard 2017	We present extensive experiments on resource and accuracy tradeoffs and show strong performance compared to other popular models on ImageNet classification.	The results show that the proposed system yields good classification rates which they are comparable to that of feature extractions-based schemes for image classification
Y. Lu, S. Yi, N. Zeng. Liu, and Y. Zhang. 2017	In this paper, we present a deep learning-based approach to detect diseases and pests in plants using images captured in real life scenarios with heterogeneous backgrounds.	The results show that it can effectively detect a disease and pests of nine classes including healthy ones using a neural network.
P. Konstantinos Ferentinos. 2018	This paper proposes a mathematical model of plant disease detection and recognition based on deep learning, which improves accuracy, generality, and training efficiency	The deep learning algorithm proposed in the paper is of great significance in intelligent agriculture, ecological protection, and agricultural production.

Table 1: Related Works Summary

III. Methodology

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN COMPUTER SCIENCE AND ENGINEERING TECHNOLOGIES

ISSN 2454-9924

The procedure to develop our system is clearly described in this section.

A. Data Collection

First, we need to create an image dataset and collect images of different diseases of leafy vegetables for our study and these images are extracted from the different sources and combined together to form a dataset based on the image labels.

B. Data Augmentation

Data augmentation needs to be performed because of a lack of images availability in the sources. This could be performed for each and every class in the dataset. We used shear augmentation, zoom augmentation, horizontal flip augmentation and validation split augmentation. Some of the augmentation examples are mentioned below in Fig 1 and 2

Fig1: Horizontal Flip Augmentation

Fig2: Zoom Augmentation

C. Data Preprocessing

Once the image dataset is prepared, preprocessing needs to be performed on every image to remove noisy data by removing the corrupted image and the images lacking quality by means of pixels and size.

D. Train-Test Split

Split the dataset into train and test images, which will be used furtherly for training and testing of our model.

E. Model Creation

Construct a CNN by using the AlexNet model with the help of Keras and Tensorflow libraries, here we will train and test dataset for training our neural network and for validation we use test dataset, by this we obtain accuracy. Model is trained by using PyCharm IDE.

F. User Interface Creation

Build a web application using CSS, JS, HTML for web designing. In our study, Common gateway interface is used communicating web server to our backend code.

Once the application is completed, users will be able to upload the image and predict the disease like what kind of disease it is by using a trained model in the training phase. We hosted this whole application in the Xampp/Apache server. Fig 3 shows the home page of the user interface.

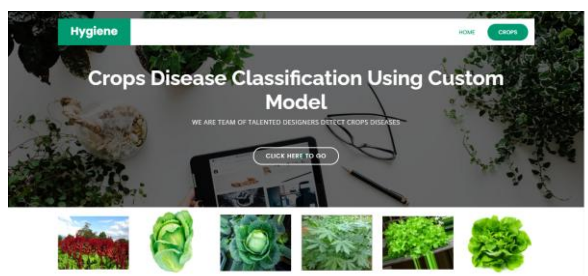

Fig3: User Interface for Plant Detection

Block diagram of proposed model :

Fig 4 shows the step by step process involved in this disease detection of leafy vegetables.

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN COMPUTER SCIENCE AND ENGINEERING TECHNOLOGIES

ISSN 2454-9924

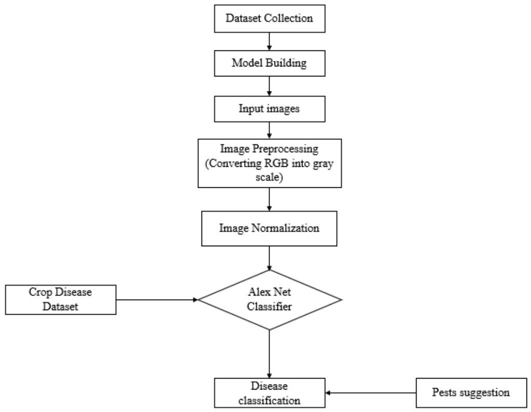

Fig 4. Block diagram of proposed method

Convolutional neural network

Convolutional neural networks (ConvNets or CNNs) are one of the most common types of neural networks used to recognize and classify pictures. CNNs are commonly utilized in domains such as object detection, face recognition, and so on. CNN image classifications take an input image, analyze it, and categorize it into several groups. (E.g., Dog, Cat, Tiger, Lion). An input image is seen by computers as an array of pixels, with the number of pixels varying depending on the image resolution. To train and evaluate deep learning CNN models, each input image will be sent through a succession of convolution layers using filters (Kernels), Pooling, fully connected layers (FC), and the SoftMax function to identify an object with probabilistic values ranging from 0 to 1. Convolutional neural networks are composed of multiple layers of artificial neurons. Artificial neurons, a rough imitation of their biological counterparts, are mathematical functions that calculate the weighted sum of multiple inputs and output an activation value.

A convolutional neural network consists of an input layer, hidden layers and an output layer. In any feed-forward neural network, any middle layers are called hidden because their inputs and outputs are masked by the activation function and final convolution. In a convolutional neural network, the hidden layers include layers that perform convolutions. Typically, this includes a layer that does multiplication or other dot product, and its activation function is commonly ReLU. This is followed by other convolution layers such as pooling layers, fully connected layers and normalization layers.

Architecture of AlexNet model:

AlexNet contains 5 convolution layers and 3 fully connected layers which includes one output layer. Fig 5 shows all the layers in the architecture of the AlexNet model.

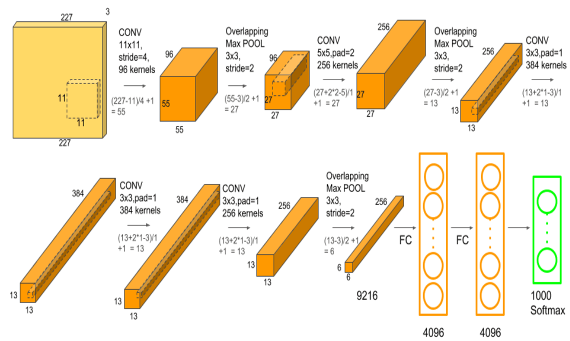

Fig 5. Detailed view of AlexNet Architecture

Thelayer-wise feature specifications of the AlexNet presented in table 2. 227x227 is the dimension or size of the input shape image.

Layers	Description
Convolution Layer 1	There are ninety six feature maps, each of the size of 55x55 and it uses same number(96) kernels, each one of 55x55 size with 4 pixel strides
Max Pooling	There are ninety six feature maps, each of the size of 27x27 and it uses same number(96) kernels, each one of 3x3 size with 2 pixel strides

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN COMPUTER SCIENCE AND ENGINEERING TECHNOLOGIES

ISSN 2454-9924

Convolutional layer 2	There are two hundred fifty-six feature maps, each of the size of 27x27 and it uses same number(256) kernels, each one of 5x5 size with 2 pixel strides
Max pooling	There are two hundred fifty-six feature maps, each of the size of 13x13 and it uses same number(256) kernels, each one of 3x3 size with 2 pixel strides
Convolutional Layer 3	There are three hundred eighty-four feature maps, each of the size of 13x13 and it uses same number(384) kernels, each one of 3x3 size with 1 pixel strides
Convolutional Layer 4	There are three hundred eighty-four feature maps, each of the size of 13x13 and it uses same number(384) kernels, each one of 3x3 size with 1 pixel strides
Convolutional Layer 5	There are two hundred fifty-six feature maps, each of the size of 13x13 and it uses same number(256) kernels, each one of 3x3 size with 1 pixel strides
Max Pooling	There are two hundred fifty-six feature maps, each of the size of 6x6 and it uses same number(256) kernels, each one of 3x3 size with 2 pixel strides
Fully connected layer 1	It contains 4096 neurons
Fully connected layer 2	It contains 4096 neurons
Output Layer	It contains thousands(1000) of neurons.

Table 2. Layers of Alexnet

Output layer neurons(units) completely rely on the class quantity. If there are one 500 classes in the dataset, the output layer contains 500 units.

Convolutional layer

This is the initial layer that extracts the different features from the input images as shown in Fig 6 . The convolution mathematical operation is done between the input image and a filter of a specific size MxM in this layer. The dot product between the filter and the portions of the input image with regard to the size of the filter is taken by sliding the filter from across the input image (MxM).

The Feature map is the outcome, and it contains information about the image such as its corners and edges. This feature map is then supplied to further layers, which learn a variety of other features from the input image. Feature maps can be shown in Fig 7.

$$\text{Output height} = \frac{(W - K + 2(P))}{S} + 1$$

Here, W is the height or width of the image. K is the height or width of the single filter or kernel. P is the number of padding. S is the number of pixels shifted over the input matrix.

ISSN 2454-9924

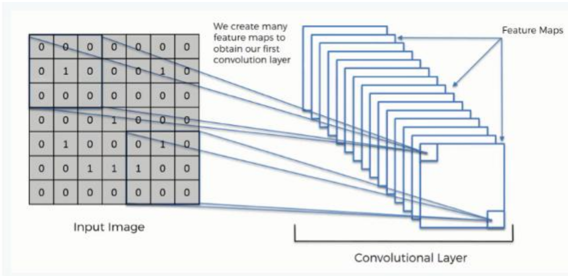

Fig 6. Feature Extraction in Convolution Layer

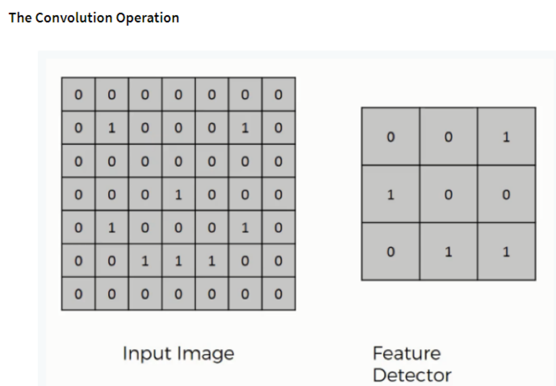

Fig 7. Feature maps

Pooling layers

The main goal of the pooling layer is to decrease the size of the feature map so that it reduces the number of computations. Apparently this reduces the connections between layers. It operates independently on each feature map. There are many types of pooling operations based on the required mechanism.

Maxpooling Operation (Feature Selection):

Selecting the most significant features is the primary goal of the max pooling layer. It distinguishes the objects

For this purpose, it first selects the window size (commonly 2 or 3). Then a stride of size 1 is selected to control the movement of the window over these feature maps, and resultantly the maximum value is selected and inserted into the pooled feature map (Eq. 2) which is the output matrix generated by the Maxpooling operation. The highest valued element is obtained in the feature map. The mathematical operation in max pooling given below.

$$P(1, 1) \Rightarrow Max(0.9, 0.7, 0.1) = 0.9$$

The average of the elements in a predefined sized Image segment is calculated using Average Pooling. Sum Pooling calculates the total sum of the components in the predefined section. The Pooling Layer links the Convolutional Layer and the FC Layer.

Fully Connected Layer

Fully connected layer consists of weights, biases and neurons. This layer provides connection between two layers of neurons. In CNN architecture few layers are prior to the output layer.

Input image obtained from previous layers is given to the FC layers after it is flattened. In this process, it generates a support vector which is sent to additional fully connected layers. The mathematical operations are performed in these FC layers.

Dropout Layer

A dropout layer is used to eradicate the overfitting problem which arises when a model achieves high performance when it is fed with training data. Dropout layers resolve this issue by dropping a few neurons while in the training process and makes the model less complex. If the model has been sent 0.3 dropout, it will drop 30 percent of the nodes in the whole neural network.

Activation Functions

The activation function plays a key role in the CNN model. Main purpose of the activation functions is to decide on neurons to fire or not. It performs non linear transformation on input signals. There are different types of activation functions namely SoftMax, ReLU, tanH, and Sigmoid functions. These are the most used functions in CNN.

In this model, ReLU activation function is used for all the layers except output layer and for that layer softmax function is used

ISSN 2454-9924

1.**ReLU : (Rectified linear activation function)**

ReLU gives output as input if it is positive, otherwise, it gives output zero.The default activation when developing convolutional neural networks and multilayer Perceptron is rectified linear activation

f(x)=max(0,x)

2.**Soft Max :**

Softmax is a mathematical function that transforms a vector of numbers into a vector of probabilities. These each value probabilities are proportional to the relative scale of corresponding value in the vector.

$$softmax(z_i) = \frac{exp(z_i)}{\sum_j exp(z_j)}$$

where **Zi** is input vector to the softmax

exp stands for exponential

Σ is sigma function

3.Model Parameters and Layers in Proposed model :

The layered architecture of the Actual alexnet model used in the proposed system is shown in Fig 8,9,10. The combination of five Convolution 2d Layers, Three max pooling layers, seven activation functions, 3 dense layers, 2 dropout layers was clearly mentioned as well.

conv2d (Conv2D)	(None, 55, 55, 96)	34944
activation (Activation)	(None, 55, 55, 96)	0
max_pooling2d (MaxPooling2D)	(None, 27, 27, 96)	0
batch_normalization (BatchNo	(None, 27, 27, 96)	384
conv2d_1 (Conv2D)	(None, 17, 17, 256)	2973952
max_pooling2d_1 (MaxPooling2	(None, 8, 8, 256)	0
batch_normalization_1 (Batch	(None, 8, 8, 256)	1024
conv2d_2 (Conv2D)	(None, 6, 6, 384)	885120
activation_1 (Activation)	(None, 6, 6, 384)	0
batch_normalization_2 (Batch	(None, 6, 6, 384)	1536
conv2d_3 (Conv2D)	(None, 4, 4, 384)	1327488

Fig 8. Alexnet Layer Architecture part 1

activation_2 (Activation)	(None, 4, 4, 384)	0
batch_normalization_3 (Batch	(None, 4, 4, 384)	1536
conv2d_4 (Conv2D)	(None, 2, 2, 256)	884992
activation_3 (Activation)	(None, 2, 2, 256)	0
max_pooling2d_2 (MaxPooling2	(None, 1, 1, 256)	0
batch_normalization_4 (Batch	(None, 1, 1, 256)	1024
flatten (Flatten)	(None, 256)	0
dense (Dense)	(None, 4096)	1052672
activation_4 (Activation)	(None, 4096)	0
dropout (Dropout)	(None, 4096)	0
batch_normalization_5 (Batch	(None, 4096)	16384
dense_1 (Dense)	(None, 4096)	16781312
activation_5 (Activation)	(None, 4096)	0
dropout_1 (Dropout)	(None, 4096)	0
batch_normalization_6 (Batch	(None, 4096)	16384

Fig 9. Alexnet Layer Architecture part 2

dense_2 (Dense)	(None, 1000)	4097000
activation_6 (Activation)	(None, 1000)	0
dropout_2 (Dropout)	(None, 1000)	0
batch_normalization_7 (Batch	(None, 1000)	4000
dense_3 (Dense)	(None, 4)	4004
activation_7 (Activation)	(None, 4)	0

```
Total params: 28,083,756
Trainable params: 28,062,620
Non-trainable params: 21,136
```

Fig 10. AlexNet model parameters

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN COMPUTER SCIENCE AND ENGINEERING TECHNOLOGIES
ISSN 2454-9924

IV. Results and Discussions

In this results and discussion session we will discuss results that are extracted from the CNN model.

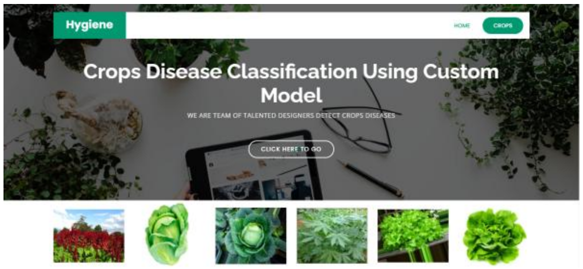

Fig 8. Home page

Figure 8 was the home page of our web page. This page contains a brief overview of different disease detection of leafy vegetables.

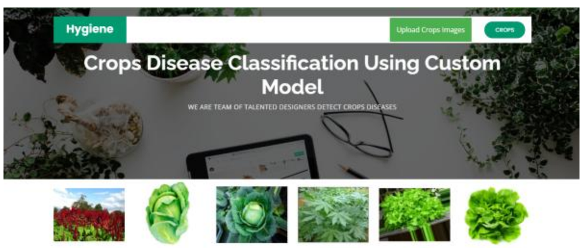

Fig 9. Upload images page

The above shown figure enables users to upload leafy vegetable images on the home page and it contains the plant list that is predicting the disease in our project.

After uploading the images, we will get corresponding disease predictions and those disease figures are shown below.

Results of Cabbage's disease Prediction :

DDLVUDAM[2] detects disease of cabbage vegetables from the uploaded images of the cabbages. Figure 10 shows the disease and pesticide suggestion of the diseases such as leafminer, mildew, backmoth and no suggestions for healthy cabbage vegetables.

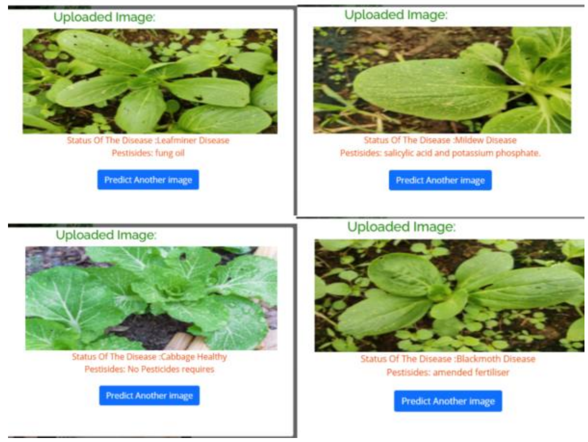

Fig 10. Cabbage diseases

Results of Amaranthus's disease Prediction :

DDLVUDAM detects disease of amaranthus vegetables from the uploaded images of the amaranthus. Figure 11 shows the disease and pesticide suggestion of the diseases such as white rust, blight and no suggestions for healthy amaranthus vegetables.

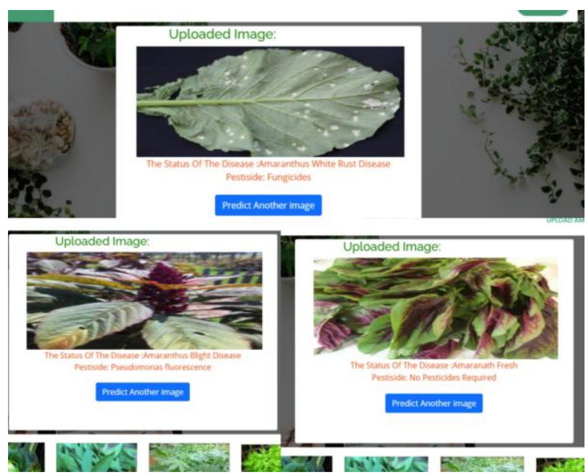

Fig 11. Amaranthus disease

Results of Lettuce's disease Prediction :

[2] DDLVUDAM - DISEASE DETECTION OF LEAFY VEGETABLES USING DEEP ALEXNET MODEL

ISSN 2454-9924

DDLVUDAM detects disease of lettuce vegetables from the uploaded images of the lettuce. Figure 12 shows the disease and pesticide suggestion of the diseases such as bacterial disease, fungal and lettuce virus and no suggestions for healthy lettuce vegetables.

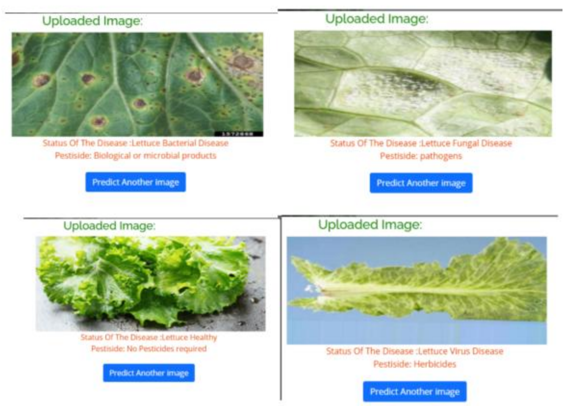

Fig 12. Lettuce Disease

.

The accuracy of Amarantus, Cabbage, Lettuce are shown in Fig 13, 14 and 15 respectively. Green line graph depicts the information about the training data set. On the other hand, the red line graph depicts the information about the validation data set. Number of epoch was mentioned on the x-axis and accuracy of **the model is mentioned on the y-axis.**

Fig 13. Line plot of Amarantus accuracy

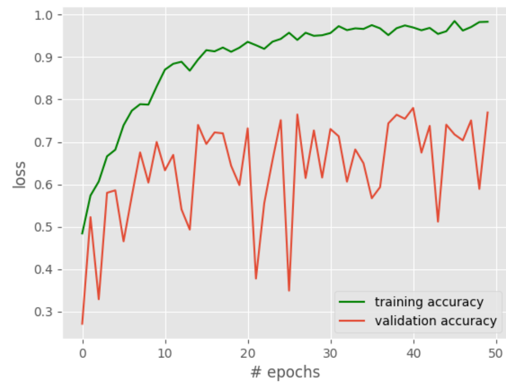

Fig 14. Line plot of Cabbage accuracy

Fig 15. Line plot of Lettuce accuray

Histogram

The Histogram from Fig 16 depicts the performance of different leafy vegetables and their training and testing accuracy.

Cabbage **Lettuce** **Amaranthus**

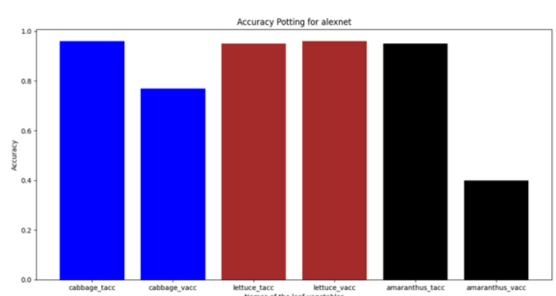

Fig 16. Histogram of results

ISSN 2454-9924

V. Conclusion

This study provides the prediction and classification of crop disease using deep learning techniques. The proposed CNN neural network obtained more than 80% of accuracy compared to the vgg16 model.

DDLVUDAM[3] is trained on leafy vegetable images dataset and by using the trained model we can easily predict the accurate results of the crop diseases.

References

[1] T. Gupta," Plant leaf disease analysis using image processing technique with modified SVM-CS classifier," Int. J. Eng. Manag. Technol, no. 5, pp. 11-17, 2017.

[2] Y. Es-saady,T, El Massi, M. El Yassa. Mammas, and A. Benazoun," Automatic recognition of plant leaves diseases based on serial combination of two SVM classifiers," International Conference on Electrical and Information Technologies (ICEIT) pp. 561-566, 2016.

[3] Greg Olm Schenk, Hao Tang, Zhigang Zhu, "Crowd Counting with Minimal Data Using Generative Adversarial Networks for Multiple Target Regression, Applications of Computer Vision (WACV), 2018 IEEE Winter Conference on Computer Vision (WACV), 2018

[4] L. Liu and G. Zhou, "Extraction of the rice leaf disease image based on BP neural network," International Conference on Computational Intelligence and Software Engineering, pp. 1-3,2009.

[5] S. Arivazhagan and S.V. Ligi, "Mango Leaf Diseases Identification Using Convolutional Neural Network," International Journal of Pure and Applied Mathematics, vol. 120, no. 6, pp. 11067-11079,2008.

[6] Kadir, A., "A Model of Plant Identification System Using GLCM, Lacunarity and Shen Features," Research Journal of Pharmaceutical, Biological, and Chemical Sciences Vol.5(2) 2014.

[7] Naik, M.R., Sivappagari, C., "Plant Leaf and Disease Detection by Using HSV Features and SVM," IJESC, Volume 6 Issue No.12, 2016.

[8] P. B. Padol and A. A. Yadav, "SVM classifier based grape leaf disease detection," Conference on Advances in Signal Processing (CASP), pp. 175-179, 2016.

[9] Surbhi Jain, Joy dip Dhar," Image based search engine using deep learning", 2017 Tenth International Conference on Contemporary Computing (IC3), August 2017

[10f] Arnab Baura,

https://arnabfly.github.io/arnab_blog/alexnet/

[3] DDLVUDAM - DISEASE DETECTION OF LEAFY VEGETABLES USING DEEP ALEXNET MODEL(project title)