ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE)
Vol. 7, Issue 5, May 2021

Plant leaves disease detection using CNN

1st S.Irin Sherly
Dept of IT
Panimalar Institute of Technology
Chennai, India
irinsherly15@gmail.com

2nd S.Ramya
Dept of IT
Panimalar Institute of Technology
Chennai, India
ramya9600434700@gmail.com

3rd S.Santhiya
Dept of IT
Panimalar Institute of Technology
Chennai, India
sanjothi99@gmail.com

4th V.Vasantha
Dept of IT
Panimalar Institute of Technology
Chennai, India
vasantha190600@gmail.com

***Abstract*— Smart farming system is an innovative technology that helps improve the quality and quantity of agricultural production in the country. Plant leaf disease is one of the major threats to food security because it dramatically reduces the crop yield. Accurate and precise diagnosis of diseases has been a significant challenge and the recent advances in computer vision made possible by deep learning has paved the way for camera-assisted disease diagnosis for plant leaf. It described the innovative solution that provides efficient disease detection and deep learning with convolutional neural networks (CNNs) has achieved great success in the classification of various plant leaf diseases. A variety of neuron-wise and layer-wise visualization methods were applied using a CNN, trained with a publicly available plant disease given image dataset. So, it is observed that neural networks can capture the colors and textures of lesions specific to respective diseases upon diagnosis, which resembles human decision-making.**

Keywords— *Disease Detection, deep learning, TensorFlow,* Convolutional neural networks

I. INTRODUCTION

Traditionally, identification of plant diseases has relied on human annotation by visual inspection and the agricultural production cost can be significantly increased if plant diseases are not detected and cured in their early stages using CNN model [2]. In order to train the model we are using image data set to classify. Plant diseases cause a major production and economic losses in the agricultural industry. Inorder to reduce loss we are trying to measure the affected area by disease, to determine the affected leaf. We all know that agriculture is a important occupation and backbone of our country. Inorder to improve this agriculture in a smart and better ways smart farming system is required.

U. P. Singh et al used CNN and Deep Neural Networks for extracting features for predicting diseases in mango leaves [10]. Dhakate et al Fast Region-based Convolution Neural network(Faster R-CNN) and Region based Fully Convolution Network(R-FCN). The system used Single Shot MultiBox Detector to detect diseases in plant leaves. This system is used for analysis and classification purpose only and not implemented [16]. M. H. Saleem [4] proposed a system that used image processing and artificial neural network. This model obtained accurate result in less computing time but the model was not implemented in real time. S. P. Mohanty [7] used deep learning and CNN for plant disease detection but was not deployed.

Coming to Smart farming, We all know that while doing forming these are various threats associated wit it such as weather changes, pests, insects, micro organism , water scarcity, soil variations etc. We are trying to reduce one of these threads [14]. So we are trying to detect disease at its early stage that affect plants. We all know that Food Security is directly proportional to Crop yield. When crop yield decreases food security decreases and vice versa. So, detecting these diseases at its earlier stage will be the smart farming. This earlier stage is the place where innovative part gets involved. We all know that Artificial Intelligence, Machine Learning, Deep Learning, Data Science are giving accurate results within a short span of time. We are going to classify and detect these diseases using Deep Learning and its algorithm. We are going to use an efficient Algorithm for classification CNN (Convolution Neural Networks) where we train the dataset to it. LeNet, Alex Net, Manual creation. A variety of neuron wise and layer wise visualization methods are applied to it so that we get accurate results. This mainly classifies based on the color, shape etc. Plant diseases cause a major production and economic losses in the agricultural industry it includes detection of

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

IJARMATE
Your ubi-MATE Research Paper !!!

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE)
Vol. 7, Issue 5, May 2021

diseased leaf to measure the affected area by disease, to determine the affected leaf.

II. DATA DESCRIPTION

The initial step for any image processing-based system is acquiring proper dataset which is valid. Most of the time the standard dataset is preferred but in certain circumstances we do not get proper dataset. In such conditions we can collect the images and form our own dataset [1]. Figure 1 and 2 shows the training and testing data for plant disease detection system.

The dataset can be accessed from kaggle or any other websites which is plant disease classification challenge. The data available is not labeled. Therefore, the first task is to clean and label the database.

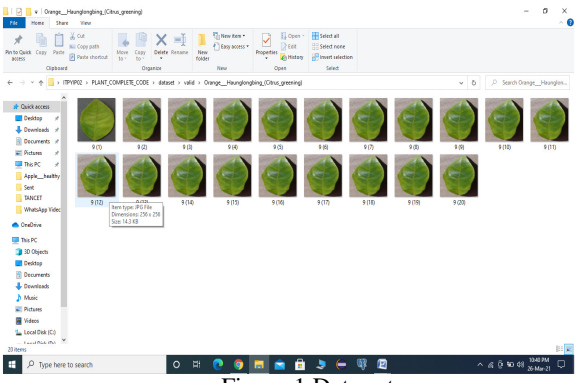

Figure:1 Data set

The database is huge so basically the images with better resolution and angle are selected. After selection of images, we should have deep knowledge about the different leaves and the disease they have. Huge research is done from plant village organization repository. Different types of plant images are studied and corresponding diseases are analyzed. After detailed study, labeling in done by segregating the images and with different diseases.

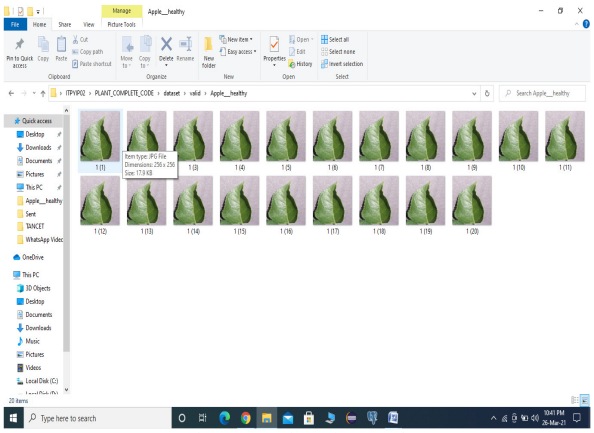

Figure:2 Plant leaves images

Anaconda Navigator

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution that allows users to launch applications and manage conda packages, environments and channels without using command-line commands. Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository, install them in an environment, run the packages and update them. It is available for Windows, macOS and Linux.

III. PROPOSED METHOD

The proposed method is described in Figure 3.

Methodology: Preprocessing and Training the model (CNN): The dataset is preprocessed such as image reshaping, resizing and conversion to an array form. Similar processing is also done on the test image. A dataset consisting of about 19 different plant species is obtained, out of which any image can be used as a test image for the system. The training dataset is used to train the model using CNN algorithm and it can identify the test image and the disease it has. CNN has different layers that are Dense, Dropout, Activation, Flatten, Convolution2D, and MaxPooling2D. After the model is trained successfully, the disease detection system can identify the disease if the plant species is contained in the dataset. After successful training and preprocessing, comparison of the test image and trained model takes place to predict the disease.

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com
International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE)
Vol. 7, Issue 5, May 2021

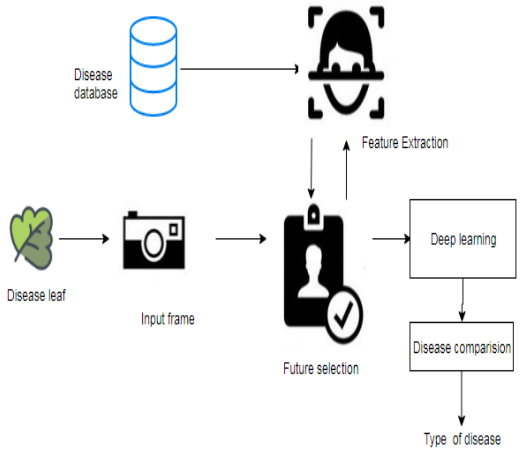

Figure 3: System Architecture of proposed system

CNN Model steps:
Conv2D: It is the layer to convolve the image into multiple images.
MaxPooling2D: It is used to max pool the value from the given size matrix and same is used for the next 2 layers.
Flatten: It is used to flatten the dimensions of the image obtained after convolving it.
Dense: It is used to make this a fully connected model and is the hidden layer.
Dropout: It is used to avoid over fitting on the dataset and dense is the output layer contains only one neuron which decide to which category image belongs.
Image Data Generator: It is that rescales the image, applies shear in some range, zooms the image and does horizontal flipping with the image. This Image Data Generator includes all possible orientation of the image.
Training Process: train_datagen.flow_from_directory is the function that is used to prepare data from the train dataset directory Target_size specifies the target size of the image. test_datagen.flow_from_directory is used to prepare test data for the model and all is similar as above. fit_generator is used to fit the data into the model made above, other factors used are steps_per_epochstells us about the number of times the model will execute for the training data.
Epochs: It tells us the number of times model will be trained in forward and backward pass.
Validation process: validation_data is used to feed the validation/test data into the model. validation_steps denote the number of validation/test samples.

Architecture of CNN: A typical architecture of a convolutional neural network contains an input layer, some convolutional layers, some fully-connected layers, and an output layer. CNN is designed with some modification on LeNet Architecture. It has 6 layers without considering input and output.

Input Layer:
The input layer has pre-determined, fixed dimensions, so the image must be pre-processed before it can be fed into the layer. Normalized gray scale images of size 48 X 48 pixels from given dataset are used for training, validation and testing. For testing propose laptop webcam images are also used, in which face is detected and cropped using OpenCV Haar Cascade Classifier and normalized.

Convolution and Pooling (ConvPool) Layers:
Convolution and pooling are done based on batch processing. Each batch has N images and CNN filter weights are updated on those batches. Each convolution layer takes image batch input of four dimension N x Color-Channel x width x height. Feature map or filter for convolution is also four dimensional (Number of feature maps in, number of feature maps out, filter width, filter height). In each convolution layer, four dimensional convolutions are calculated between image batch and feature maps. After convolution only parameter that changes is image width and height.

- ✓ New image width = old image width – filter width + 1
- ✓ New image height = old image height – filter height + 1

After each convolution layer down sampling / subsampling is done for dimensionality reduction. This process is called Pooling. Max pooling and Average Pooling are two famous pooling methods. In this project max pooling is done after convolution. Pool size of (2x2) is 12 taken, which splits the image into grid of blocks each of size 2x2 and takes maximum of 4 pixels. After pooling only height and width are affected. Two convolution layer and pooling layer are used in the architecture. At first convolution layer size of input image batch is Nx1x48x48. Here, size of image batch is N, number of color channel is 1 and both image height and width are 48 pixel. Convolution with feature map of 1x20x5x5 results image batch is of size Nx20x44x44. After convolution pooling is done with pool size of 2x2, which results image batch of size Nx20x22x22. This is followed by second convolution layer with feature map of 20x20x5x5, which results image batch of size Nx20x18x18. This is followed by pooling layer with pool size 2x2, which results image batch of size Nx20x9x9.

Fully Connected Layer:
This layer is inspired by the way neurons transmit signals through the brain. It takes a large number of input features and transforms features through layers connected with trainable weights. Two hidden layers of size 500 and 300 unit are used in fully-connected layer. The weights of

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE)
Vol. 7, Issue 5, May 2021

these layers are trained by forward propagation of training data then backward propagation of its errors. Back propagation starts from evaluating the difference between prediction and true value, and back calculates the weight adjustment needed to every layer before. We can control the training speed and the complexity of the architecture by tuning the hyper-parameters, such as learning rate and network density. Hyper-parameters for this layer include learning rate, momentum, regularization parameter, and decay. The output from the second pooling layer is of size Nx20x9x9 and input of first hidden layer of fully-connected layer is of size Nx500. So, output of pooling layer is flattened to Nx1620 size and fed to first hidden layer. Output from first hidden layer is fed to second hidden layer. Second hidden layer is of size Nx300 and its output is fed to output layer of size equal to number of facial expression classes.

Output Layer:
Output from the second hidden layer is connected to output layer having seven distinct classes and output is obtained using the probabilities for each of the seven classes. The class with the highest probability is the predicted class.

IV. EXPERIMENTAL SETUP

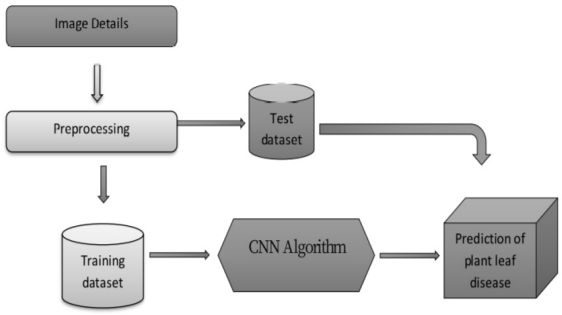

Figure 4 : Flow diagram for Prediction system

The figure 4 shows the flow diagram of our plant disease prediction system.

Import the given image from dataset: We have to import our data set using keras. In preprocessing image, using datageneratorfuction also we create size, rescale, range, zoom range, horizontal flip. Then we import our image dataset from folder through the data generator function. Here we set train, test, and validation also we set target size, batch size and class-mode from this function we have to train.

To train the module by given image dataset:
To train our dataset using classifier and fit generator function also we make training steps per epoch's then total number of epochs, validation data and validation steps using this data we can train our dataset

Gray Scale Base: Gray information within a leaf can also be treating as important features. leaf features such as shape, vein and damaged part of leaf appear generally darker than its surrounding leaf regions. Various recent feature extraction algorithms search for local gray minima within segmented leaf regions. The input images are first enhanced by contrast-stretching and gray-scale morphological routines to improve the quality of local dark patches and thereby make detection easier. The extraction of dark patches is achieved by low-level gray-scale threshold. Based method and consist three levels. Leaf gray scale behavior in pyramid images and it utilizes hierarchical Leaf location consist three levels. Higher two level based on images at different resolution and the lower level, edge detection method.

Edge Base: This work was based on analyzing line drawings of the leaves from photographs, aiming to locate leaf features. To initially the images are enhanced by applying median filter for noise removal and histogram equalization for contrast adjustment.

Working process of Layers in CNN model:
A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics. The architecture of a ConvNet is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field. Their network consists of four layers with 1,024 input units, 256 units in the first hidden layer, eight units in the second hidden layer, and two output units.

Input Layer: Input layer in CNN contain image data. Image data is represented by three dimensional matrixes. It needs to reshape it into a single column. Suppose you have image of dimension 28 x 28 =784, it need to convert it into 784 x 1 before feeding into input.

Convo Layer: Convo layer is sometimes called feature extractor layer because features of the image are get extracted within this layer. First of all, a part of image is connected to Convo layer to perform convolution operation

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

IJARMATE
Your ultimate Research Paper !!!

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE)
Vol. 7, Issue 5, May 2021

as we saw earlier and calculating the dot product between receptive field(it is a local region of the input image that has the same size as that of filter) and the filter. Result of the operation is single integer of the output volume. Then the filter over the next receptive field of the same input image by a Stride and do the same operation again. Itwill repeat the same process again and again until itgoes through the whole image. The output will be the input for the next layer.

Pooling Layer: Pooling layer is used to reduce the spatial volume of input image after convolution. It is used between two convolution layers. If itapplies FC after Convo layer without applying pooling or max pooling, then it will be computationally expensive. So, the max pooling is only way to reduce the spatial volume of input image. It has applied max pooling in single depth slice with Stride of 2. It can observe the 4 x 4 dimension input is reducing to 2 x 2 dimensions.

Fully Connected Layer (FC): Fully connected layer involves weights, biases, and neurons. It connects neurons in one layer to neurons in another layer. It is used to classify images between different categories by training.

Softmax / Logistic Layer: Softmax or Logistic layer is the last layer of CNN. It resides at the end of FC layer. Logistic is used for binary classification and softmax is for multi-classification.

Output Layer: Output layer contains the label which is in the form of one-hot encoded.

Plant disease identification: We give input image using keras preprocessing package. That input mage converted into array value using pillow and image to array function package. We have already classified disease of leaf in our dataset. It classifies what are the plant disease leaves. Then we have to predict our leaf diseases using predict function.

V. RESULTS

The deep learning and machine learning algorithms are developed using Mat lab. The workstation consists of Intel Pentium IV 2.80 GHz and a 2.00 GB to train the system. Figure 5a, 5b and 5c shows the implementation results of our proposed model. This system produces the result with the accuracy of 90.63%.

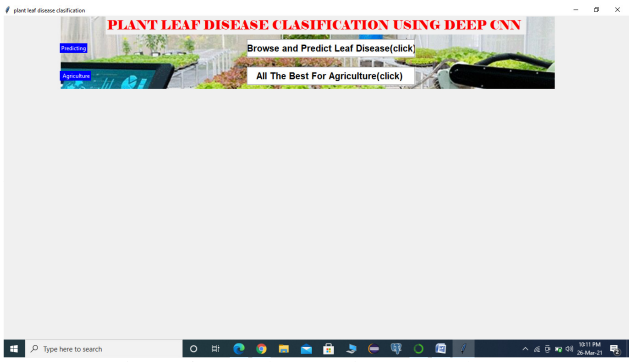

Figure 5a : Implementation screenshots

Figure 5b:Test data for the model

Figure 5c: Plant leaves detected with accuracy

V. CONCLUSION

It focused how image from given dataset (trained dataset) in field and past data set used predict the pattern of plant diseases using CNN model. This brings some of the following insights about plant leaf disease prediction. As maximum types of plant leaves will be covered under this system, farmer may get to know about the leaf which may

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

IJARMATE
Your 4th IMATE Research Paper !!!

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE)
Vol. 7, Issue 5, May 2021

never have been cultivated and lists out all possible plant leaves, it helps the farmer in decision making of which crop to cultivate. This system produces the result with the accuracy of 90.63%. Also, this system takes into consideration the past production of data which will help the farmer get insight into the demand and the cost of various plants in market.

References

[1] D. P. Hughes and M. Salathe, ''An open access repository of images on plant health to enable the development of mobile disease diagnos- tics,'' 2015, *arXiv:1511.08060*. [Online]. Available: http://arxiv.org/abs/ 1511.08060.

[2] J. G. A. Barbedo, ''Factors influencing the use of deep learning for plant disease recognition,'' *Biosyst. Eng.*, vol. 172, pp. 84–91, Aug. 2018, doi:10.1016/j.biosystemseng.2018.05.013.

[3] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, ''Convolutional neural networks: An overview and application in radiology,'' *Insights Imag.*, vol. 9, no. 4, pp. 611–629, Aug. 2018, doi: 10.1007/s13244-018- 0639-9.

[4] M. H. Saleem, J. Potgieter, and K. M. Arif, ''Plant disease detection and classification by deep learning,'' *Plants*, vol. 8, no. 11, p. 468, Oct. 2019, doi: 10.3390/plants8110468.

[5] K. P. Ferentinos, ''Deep learning models for plant disease detection and diagnosis,'' *Comput. Electron. Agricult.*, vol. 145, pp. 311–318, Feb. 2018, doi: 10.1016/j.compag.2018.01.009.

[6] A. K. Rangarajan, R. Purushothaman, and A. Ramesh, ''Tomato crop disease classification using pre-trained deep learning algorithm,'' *Procedia Comput. Sci.*, vol. 133, pp. 1040–1047, Jan. 2018, doi: 10.1016/j.procs. 2018.07.070.

[7] S. P. Mohanty, D. P. Hughes, and M. Salathé, ''Using deep learning for image-based plant disease detection,'' *Frontiers Plant Sci.*, vol. 7, p. 1419, Sep. 2016, doi: 10.3389/fpls.2016.01419.

[8] E. C. Too, L. Yujian, S. Njuki, and L. Yingchun, ''A comparative study of fine-tuning deep learning models for plant disease identification,'' *Com- put. Electron. Agricult.*, vol. 161, pp. 272–279, Jun. 2019, doi: 10.1016/j. compag.2018.03.032.

[9] J. Shijie, J. Peiyi, H. Siping, and S. Haibo, ''Automatic detection of tomato diseases and pests based on leaf images,'' in *Proc. Chin. Autom. Congr. (CAC)*, Oct. 2017, pp. 2510–2537, doi: 10.1109/CAC.2017.8243388.

[10] U. P. Singh, S. S. Chouhan, S. Jain, and S. Jain, ''Multilayer convolution neural network for the classification of mango leaves infected by anthrac- nose disease,'' *IEEE Access*, vol. 7, pp. 43721–43729, 2019, doi: 10. 1109/ACCESS.2019.2907383.

[11] Rahat Yasir, Md. Ashiqur Rahman, "*Dermatolomical disease detection usinm imame processinm and artificial neural netgork*".ICECE, International Conference on pp. 687-690, IEEE, 2014.

[12] Samajpati, thavini J. and Sheshang D. Degadwala, "*Hybrid approach for apple fruit diseases detection and classification",* Advance Computing Conference (IACC), International on ,pp. 468-471, IEEE, 2015.

[13] Francis, Jobin, and t. K. Anoop, "*Identification of leaf diseases in pepper plants usinm soft computinm techniques*", In Emerging Devices and Smart Systems (ICEDSS), Conference on, pp. 168-173, IEEE, 2016.

[14] Shah, J.P., Prajapati, H.t. and Dabhi, V.K., "*A survey on detection and classification of rice plant diseases*", In Current Trends in Advanced Computing (ICCTAC), International Conference on (pp. 1-8), IEEE, 2016.

[15] Sangamithraa, P.t. and Govindaraju, S., "*Lunm tumor detection and classification usinm EK-Mean clusterinm*", In WiSPNET, International Conferenceon (pp. 2201- 2206). IEEE, 2016.

[16] Dhakate, Mrunmayee, and A. Ingole. "*Diamnosis of pomegranate plant diseases using neural netgork*", In NCVPRIPG, Fifth National Conference on, pp. 1-4. IEEE, 2015.

[17] Tansal, P., Yadav, H. and Sunkaria, R.K., "*Impulse noise removal usinm MDBUTMF gith histomram estimation*", In Advance Computing Conference (IACC), International on ,pp. 468-471, IEEE, 2015.

[18] Ramakrishnan M., Sahaya Anselin Nisha, "*Groundnut leaf disease detection and classification by usinm back probamation almorithm*", In Communications and Signal Processing (ICCSP), International Conference on, pp. 0964-0968, IEEE, 2015.

[19] Hinton, G.E.; Osindero, S.; Teh, Y.-W. A fast learning algorithm for deep belief nets. *Neural Comput.* **2006**, *18*, 1527–1554. [CrossRef]

[20] Hinton, G.E.; Salakhutdinov, R.R. Reducing the dimensionality of data with neural networks. *Science* **2006**,*313*, 504–507. [CrossRef]