Abstract

One of the significant areas of Indian Economy is Agriculture. Work to practically half of the nation’s workforce is given by Indian horticulture segment. As a part of Agriculture, Cotton plays a major role in economic resource of Telangana. Huge number of farmers grows cotton in their fields as the lands fit to that crop. Beside the advantage the major problem affecting the crop are the diseases that are unknown to the farmers at early stages and losing the entire crop when he gets aware on that. As a solution, we can identify the disease in the early stage and rectify before it affects the entire crop. This can be done by looking into images collected from the crop and given it as a test sample to the convolution neural network, where we test the sample with the existing training data and identify the major areas that are affected with the disease. As an improvement we can also identify the disease that is also affected and apply the required pesticides. As a result, 91% of the diseases were correctly identified.

Keywords: Neural Networks, Layers, Filter, Pooling, Padding, softmax
I. Introduction

Agriculture is major resource in India, and to Telangana cotton is one of the major crop resource and an economic strength to the farmers, but due to the effect of crop diseases, the farmers are not able to get the profits. In spite of the difficulties given in the issue explanation plant disease recognition is as yet a functioning zone of research. Various methodologies have been proposed throughout the years. In standard systems approach for acknowledgment and partition of plant diseases can be practiced using Support Vector Machine computations. Profound Neural Networks[I] is applied on a wide range of uses to produce start to finish learning. Neural Network gives a mapping between an info and yield. The contribution to this paper is given as a picture of an infection plant and yield is a harvest disease [II].

I.i. Convolution Neural Network (CNN) for Image Classification

CNN [I] [II] process raw image data to produce useful features for learning image label. The layers of the network are discussed below:

I.i.a. Convolution Layers

For an input channel of 6*6 a filter of 3*3 is applied in order to summarize the data. Convolution layer use filter to scan over local regions in the input space which are just the collections of pixel near one another, in capturing information about features in the region. Each filter is a small square image of 3*3 or 5*5.

I.i.b. Filter

When a filter is applied on input, feature map is generated. Filter is tiled across input space to provide across region between input filter. They are used to generate a small representation of input space.

For 6*6, 3 input channel, when a 3*3 filter is applied a 4*4 feature map is generated with a 50% decrease in the overall size of the dataset [V].

I.i.c. Stride

It is number of steps we take to move the filter to a location. On a 6*6 input channel, if we apply the filter with stride=1, then 4*4 feature map is generated, but when stride=2, then it will be out of box.

The size of the input space can be modified by just zeros to padding dimensions[IV]. Adding zero’s affect the values in the feature map by the bias in the padding dimensions. The choice of filter Map and padding dimension depend on the desired size of the input dimensions.

\[
\left(\frac{n+2p-f}{s} + 1\right) \times \left(\frac{n+2p-f}{s} + 1\right)
\]

(1)

Where n is the input dimension, p is padding dimension and f is filter dimension.

I.i.d. Pooling Layers

Pooling layer reduces the size of the incoming data by producing the local summary of a small region of an input space. Unlike convolution layers they have no
parameters. The three common pooling layers used are Maximum, Minimum and Average Pooling layers. Among these the most common layer used is Max pooling layer.

I.i.e. Fully Connected and Output Layer

Compressed and extracted data from convolution and pooling layers from pixel data; this is the input to fully connected layer.

The fully connected layer is used to generate probabilities for which label to assign to each image. The convolution and pooling layers are for data pre-processing and fully connected layers learns decision boundaries.

At the end of the network is softmax activation function[VI][VII] which is to assign the possible probabilities to each possible label.

This strategy was actualized for cotton and relying upon the sort and phase of infection, the order exactness was somewhere in the range of 75% and 90%. Another methodology dependent on leaf pictures and utilizing CNNs as a system for a programmed recognition and grouping of plant infections was utilized with K-means as a bunching strategy. CNN comprised of 10 hidden layers. The output comprises of 4 different classes which helps us to identify the disease that affected for cotton crop[III].

II. Network Model

Here a network of 3*3 layers with max poling is used to construct a network model with 4 output layers with a ReLu activation function.

The following Table 1 shows the structure of the network model in identifying the disease of a crop plant.

| Layer (type)               | Output Shape            | Param # |
|----------------------------|-------------------------|---------|
| conv2d_1 (Conv2D)          | (None, 32, 128, 128)    | 320     |
| activation_1(Activation)   | (None, 32, 128, 128)    | 0       |
| conv2d_2 (Conv2D)          | (None, 32, 126, 126)    | 9248    |
| activation_2 (Activation)  | (None, 32, 126, 126)    | 0       |
| max_pooling2d_1 (MaxPooling2) | (None, 32, 63, 63) | 0       |
| dropout_1 (Dropout)        | (None, 32, 63, 63)      | 0       |
| conv2d_3 (Conv2D)          | (None, 64, 61, 61)      | 18496   |
| activation_3 (Activation)  | (None, 64, 61, 61)      | 0       |
II.i. **Architecture for Detecting Plant Disease**

The following is the diagram used to identify the disease of a plant:

**Phase 1: Data Set Collection**

The data set comprises of 4 various diseases [VII] [VIII], among which each disease consist a set of 100 leaf images, a total of 400 images, among which 80% of images (320) are used for training set and 20% of images (80) are used for test case.

**Phase 2: Pre-processing of the dataset**: The data is pre-processed in order to remove the irrelevant data.

**Phase 3: Feature Extraction from the Plant**: The dataset is partitioned into 80/20 proportion of preparing and approval set protecting the index structure.

**Phase 4: Epochs**: As the number of epochs increased the accuracy also increased.

**Phase 5: Testing**: A new directory containing 33 test images is created later for prediction purpose.
III. Steps for Detecting Plant Disease at early stage:

III.i. Data Collection and Dataset Preparation

Data Collected from internet and research Laboratories. The data set comprises of 4 various diseases, among which each disease consist a set of 100 leaf images, a total of 400 images, among which 80% of images (320) are used for training set and 20% of images (80) are used for test case.

III.ii. Methods

Architectures used for Classification

The architecture used for classification is Alexnet which are designed over Large Scale Visual Recognition Challenge” (ILSVRC) for the image dataset. Alexnet consists of 5 Convolution layers, 3 fully connected layers and one with a softMax layer. The first two layers are followed by a normalization and a pooling layer. All the seven layers of AlexNet have a ReLu activation unit and the first 2 fully connected layers have a dropout layer with 0.5.
III.ii.a. Image Pre-Processing and Labeling

Images may be of different size, quality or resolution, so firstly they need to be preprocessed, for example images with less than 500 px will be resized to 128 X 128 in order to reduce the time for training or resize the image to a fixed size, then flatten the image into a list of raw pixel intensities.

III.ii.b. Training

Training convolutional neural network (CNN) for image classification model is done. Convolution2D, MaxPooling2D architecture will be utilized and changed in accordance with help our various classifications (classes). Rectified Linear Units (ReLU) will therefore be utilized as substitute for immersing nonlinearities. This activation function adaptively will gain proficiency with the parameters of rectifiers and improve precision at irrelevant extra computational expense. Training is done on 240 samples, validate on 60 samples[XII][XIII]. Train on 240 samples, validate on 60 samples, First Epoch 240/240, loss: 1.0512 - acc: 0.5792 - val_loss: 0.9060 - val_acc: 0.6667, Second Epoch 240/240, loss: 0.6529 - acc: 0.6917 - val_loss: 0.6996 - val_acc: 0.6833. As the number of epochs increase the chance, the accuracy for the classification increases[XIII].

Test Loss: 0.493409152826
Test accuracy: 0.916666654746
The accuracy is increased to 91% in classifying the correct disease
When an input is given as a test case for already trained data, 74% it was properly classified and in the second epoch the accuracy was increased to 91%.

III.ii.c. Testing

The test set for prediction of disease will evaluate the performance of the classifier.

(a) Fine-Tuning: Adjusting grows the precision of forecast by rolling out little improvements to improve or upgrade the outcome. The most fitting model for plant infection acknowledgment will be practiced through the system of test change of the parameters.

(b) Evaluation Measures: Measures, for instance, precision and mean survey score, mean exactness will be handled to evaluate the introduction of the classifier.

(c) Number of Epochs: As the number of epochs increases the accuracy of the classification increases.

The accuracy of the classification was increased on 2 measures[X1]:
1. Distribution of data based on choice of train and test data set.[XIII]
2. As the number of epochs increased the accuracy also increases.
Table 2: Accuracy for different percentage of training and test data.

| Choice of training-testing set distribution | Number of Epochs | Percentage of Accuracy |
|--------------------------------------------|------------------|------------------------|
| Training data -80%, Test Data – 20%        | 2                | 86                     |
|                                            | 5                | 88                     |
|                                            | 10               | 88.4                   |
|                                            | 15               | 90.5                   |
|                                            | 20               | 91                     |
| Training data -60%, Test Data – 40%        | 2                | 68                     |
|                                            | 5                | 69.72                  |
|                                            | 10               | 70.17                  |
|                                            | 15               | 73.26                  |
|                                            | 20               | 76.79                  |

Fig. 2: Accuracy of the classifier in identifying the accurate disease.
Figure 2 shows the increase in the accuracy of classification as the number of epochs is increased, and Figure 3 shows the training stages of a leaf test image with training data 80% and test data 20%.

Figure 4: Accuracy for the training 80% and test 20% data.
IV. Results and Conclusion

The work was extended to work on the Plant village dataset collected from kaggle website- this dataset consists of about 87K RGB images of healthy and diseased crop leaves which is categorized into 38 different classes.

Figure 6 and 7 shows the accuracy and loss percentage on the plant village dataset.

![Train-Test Accuracy](image)

**Fig. 5:** Accuracy for the training 60% and test 40% data.

![Training and Validation loss](image)

**Fig. 6:** Plant Village Dataset showing Training and Validation Loss
It is observed that the accuracy is increased from 68% to 91% on training the dataset through convolution neural networks, as the number of epochs are increased the accuracy is gradually increased.

References

I. Aakanksha Rastogi, Ritika Arora, Shanu Sharma, “Leaf Disease Detection and Grading using Computer Vision Technology & Fuzzy Logic,” presented at the 2nd International Conference on Signal Processing and Integrated Networks (SPIN), IEEE, 2015, pp. 500–505.

II. A. Harshavardhan, S. Babu and T. Venugopal, "An Improved Brain Tumor Segmentation Method from MRI Brain Images," 2017 2nd International Conference On Emerging Computation and Information Technologies (ICECIT), Tumakuru, 2017, pp. 1-7.

III. Barbedo, J.G., 2018. Factors influencing the use of deep learning for plant disease recognition. Biosystems engineering, 172, pp.84-91.

IV. Harshavardhan, A. Mohammad, M.D. S. Ramesh, D. RaviChythanya, K," Design methods for detecting sensor node failure and node scheduling scheme for WSN", International Journal of Engineering and Advanced Technology 9 (1),pp.5430, 2019

V. Khirade, Sachin D. and A. B. Patil. "Plant disease detection using image processing." In 2015 International conference on computing communication control and automation, pp. 768-771. IEEE, 2015.
VI. Liu, Bin, Yun Zhang, Dongjian He, and Yuxiang Li. "Identification of apple leaf diseases based on deep convolutional neural networks." Symmetry 10, no. 1 (2018): 11

VII. Md. Nazrul Islam, M.A. Kashem, Mahmuda Akter and Md. Jamilur Rahman, “An Approach to Evaluate Classifiers for Automatic Disease Detection and Classification of Plant Leaf,” presented at the International Conference on Electrical, Computer and Telecommunication Engineering, RUET, Rajshahi-6204, Bangladesh, 2012, pp. 626–629.

VIII. Prakash, Rajanala Vijaya, and Srinath Taduri. "Safe Navigation for Elderly and Visually Impaired People Using Adhesive Tactile Walking Surface Indicators in Home Environment." In Information and Communication Technology for Sustainable Development, pp. 771-778. Springer, Singapore, 2020.

IX. Roopa, Goje, and M. Sampath Reddy. "A study on pattern matching intrusion detection system for providing network security to improve the overall performance of security system." Indian Journal of Public Health Research & Development 9, no. 11 (2018): 683-687.

X. Sallauddin, M. Ramesh, D. Harshavardhan, A. Pasha, S.N. Shabana, “A comprehensive study on traditional AI and ANN architecture”, International Journal of Advanced Science and Technology 28 (17) .pp.479, 2019

XI. Shaik, Mohammed Ali, P. Praveen, and R. Vijaya Prakash. "Novel Classification Scheme for Multi Agents.", Asian Journal of Computer Science and Technology 8, no. S3 (2019): 54-58.

XII. Traore, B.B. Kamsu-Foguem, B. and Tangara, F., 2018. Deep convolution neural network for image recognition. Ecological informatics, 48, pp.257-268.

XIII. Praveen P., Rama B(2020). “An Optimized Clustering Method To Create Clusters Efficiently” Journal Of Mechanics Of Continua And Mathematical Sciences, ISSN (Online) : 2454 -7190 Vol.-15, No.-1, January (2020) pp 339-348 ISSN (Print) 0973-8975, https://doi.org/10.26782/jmcms.2020.01.00027