REALIZATION OF PLANT LEAF DISEASE RECOGNITION USING LENET

Arpit Pawar[1], D.Bharath Kumar[2], Kuldeep Tamakuwala[3], Mrs. R. Logeshwari[4]

[1,2,3]*B. Tech Student, Dept. of Computer Science and Engineering, SRM Institute of Science and Technology, Chennai*
[4]Assistant Professor, *Dept. of Computer Science and Engineering, SRM Institute of Science and Technology, Chennai*

---***---

Abstract - *Usually, plant disease identification relies on human annotation by 'visual inspection' and takes a lot of manual work. The cost of production may significantly increase if diseases are not correctly diagnosed and cured in the initial stages. Therefore, disease detection plays a crucial role in the field of agriculture. However, it demands large amounts of manpower, processing time, and in-depth knowledge about several kinds of plant diseases. Hence, deep learning techniques may be applied for detecting diseases in plants as it analyses the data from different aspects classifying them into one of the pre-defined category or class. Using the LeNet algorithm, we aim to train a model by given image dataset and classify the disease type of leaves.*

Key Words: Deep Learning, TensorFlow, Keras, Plant disease detection, LeNet.

1.INTRODUCTION

The inventions and innovations in the field of the agriculture industry have made vital contributions in improving the quality and quantity of yield. But still, in these modern times, plant disease has long been an existential threat to food quality and crop yield.

Precise and accurate diagnosis of plant diseases has been challenging however, the latest innovations in image processing backed by deep learning have paved the way for point and shoot disease diagnosis in plants.

Deep Learning coupled with LeNet has achieved great success in classifying various diseases in plants. An assortment of techniques visualizing nerves and layers have been applied using a LeNet – 5 CNN trained on a publicly available plant disease dataset. In this manner, we have seen that in the end, neural networks can distinguish the colors and textures of specific lesions of the respective diseases, which in the real-life scenario is at par with human scouting (i.e. decision-making).

2. RELATED WORKS

The proposed approach is related to flight delay estimation and prediction, and network analysis. The related work and method of how the proposed system is placed in the literature is shown.

2.1 Title: Plant Disease Detection Using Image Processing [1]

Khirade et al, has described few feature extraction and segmentation algorithm that could be employed for determining plant diseases by utilizing the snapshots of their leaves. Whilst the procedure to determine the plant leaf diseases manually by humans is extremely extensive and time consuming as it requires one to master the scouting of plant diseases and should be able to differentiate the diseases apart.
The entire procedure of plant leaf disease has been divided into the 5 steps:

A. Acquisition of Images
B. Pre-processing
C. Segmentation
D. Extraction of features
E. Final Classification

To obtain RGB images of the leaf, transformation structure is purposed by Image acquisition. In order to augment the contrast of images and to eliminate noise the images are pre-processed.

Methods like Otsu filters, K-means clustering, etc. are purposed to extract variety of feature parts of image for segmentation. Plant Features are then extracted using segmentation and then final classification is performed using various techniques. Working this way, plant diseases can be proficiently determined.

2.2 Title: Deep learning in agriculture [2]

Kamilaris et al., have surveyed deep-learning-based research in the agricultural domain. They inspected forty relevant papers, inspecting the region and problem they focus upon, technical information of the models employed, the data source used, pre-processing and records augmentation strategies adopted, and normalized overall performance to the performance metrics hired by every paper. Then, they compared the existing strategies with this knowledge, in terms of performance.

For future work as described thru this survey, they have planned to employ the principles and best practices of deep-learning in other regions of agriculture in which this modern technology has not yet been used.

2.3 Title: Pepper cutting Unmanned Ground Vehicle and disease detection using Image-processing [3]

Rakshitha and rest's system proposed is an Unmanned Ground Vehicle. Pepper grows to greater heights and is probable to get affected by various diseases and cutting it is a serious issue. To circumvent these obstacles "pepper cutting UGV and disease detection using image-processing" was put into action which will reduces the manual human works and effort. The plucking of pepper and detection of plant disease may be done with usage of image-processing techniques.

The proposed system focuses on designing a UGV for cutting the pepper fruit, disease detection and providing solution for the disease by sprinkling the required pesticides.

The paper discusses mainly the extraction (i.e. cutting and collecting) of the pepper plant based on the color recognition and plant diseases detection using the pictures of plant leaves.

3. PROPOSED SYSTEM

The proposed work is an approach that deals with predicting the plant leaf disease using deep learning technique trained model. It can also be modified to able to give proper recommendation based on kind of disease recognized.

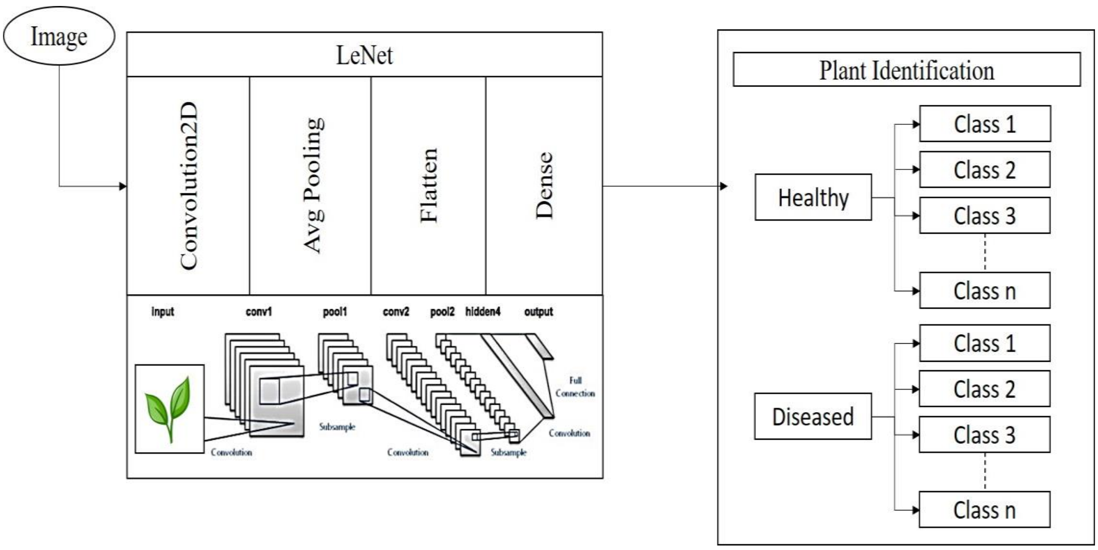

Fig -1: System Architecture

3.1 Convolutional Neural Networks

Convolutional Neural Networks (CNN) has become a industry standard for resolving tasks related with images. The key to solving tasks such as recognition of objects, face, pose and more, all of them have one or another architecture variant of CNN. There are some of the CNN architecture characteristics that brands them favorable in quite a lot of tasks related to computer vision.

3.2 LeNet - 5

The architecture of LeNet - 5 CNN consists of seven layers. The layer conformation consists three convolutional-layers, two subsampling-layers, and a couple of fully connected layers.

The first layer, the input layer — usually not considered as a network layer because there is no understanding or knowledge acquired. In order to pass the images through the subsequent-layer, a layer is developed which requires images of dimensions 32 by 32 pixels.

Also, values of pixel 0-255 are normalised to values greater than -0.1 and less than 1.175. Justification for this is to guarantee that the set has a mean of '0' & std. deviation of '1', the advantage of it is realized in the reduction of time taken during the training. In LeNet – 5 image classification, we'll standardize the values of pixel in pictures to values '0 to 1'.

The architecture of LeNet-5 also uses two other type of constructs:

- Convolutional Layer
- Sub-sampling Layer

The Convolutional Layer (C1) – i.e. actual first layer, generates six feature maps for its output and has kernel with size 5 by 5. 'Kernel' is the term of the window that contains weight values which are used for the duration of the convolution of the input values with respect to weight values.

The Sub-sampling layer (S2) trails the 'C1' layer. This layer receives 'feature maps' from C1 layer and halves their dimension, this is commonly known as down-sampling. This layer additionally produces six more feature maps, each corresponding to the one from previous layer.

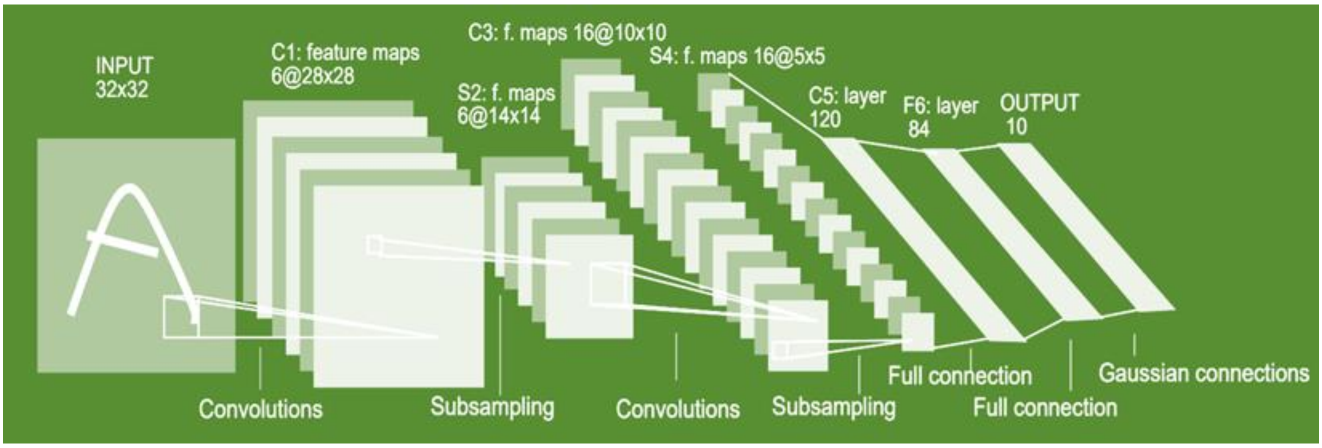

Fig -2: LeNet Architecture

3.3 LeNet TensorFlow Implementation

We commence the implementation part with importing the below libraries:

- **TensorFlow** : Platform for implementing, training, and deploying machine-learning models.

- **Keras** : Library for implementing neural net architectures to be run on CPU as well as GPU.
- **NumPy**: Library for numerical calculation of N-dimensional matrices.

After importing all the libraries, we'll now bring in the data set. The Keras library has several dataset tools that can be used with ease for this purpose.

We also need to break the data set into following parts:

- **Training data set**: The portion of the data set which is directly exposed to the neural network for training is defined as training data set.
- **Validation data set**: This portion of the data set is purposed in training and is iterated several times in order to evaluate network performance.
- **Test data set**: Following the completion of training phase, this portion of data set evaluates the network performance.

0-1 is the required pixel intensity of the images among the data set, if the pixel intensity is not within that range, we need to normalize them among the data set.

We use the Convolution2D class to create the convolution layer in the neural net by passing a pair of arguments.

❖ **Activation Fn.:**

In order to convert the signals obtained from neurons into a normalized output a mathematical operation is purposed known as Activation function. It is a neural net element that incorporates non-linearity into the system. This feature enables the neural net to provide increased representational power and help address complex operations.

The resemblance is uncanny between the descriptions of subsequent convolutional layers to that of 'C1'layer, but with discrete parameter metrics.

Within the neural net, two further layers should be considered:

- Flatten layer: transforms input into a 1D-array which can be passed to the subsequent dense layers.
- Dense layer: specifies number of units or neurons in each of the layer.

The no. of neurons in the last dense layer corresponds to no. of class or categories present in the data set.

'Tanh' has been used for activation function.

❖ **Softmax:**

To derive the probability distribution of a collection of numbers within an input vector. The result is a vector containing a 'set of values' representing the possibility of class or an event (i.e. Disease). When the values of each element in the vector are added together, the result is '1'.

3.4 Compiling and Building the model

Keras library supplies us the 'compile' method via the previously instantiated model object. The compile function allows the model to be built; we've done this in the background, behind the scenes, with few extra features like the loss function, optimizer, and metrics.

We used a loss function to train the network, which calculates the difference between the estimated value calculated by the algorithm to the actual values of the training data.

The loss values are supplemented by an optimization technique named 'Adam', which helps facilitate the number of significant changes made to the network's weights. Enabling factors such as momentum and learning rate create the ideal environment for model development to converge, with the goal of getting the loss values as negligible as possible.

We would also validate the model with the valuation set of data partition introduced earlier after each iterative process during learning phase (or epoch).

After the training process was finished, the prediction performance of the model was more than 90%. That being said, in attempt to provide a more comprehensive checking of the effectiveness on a real - world dataset, we assessed the trained model on a totally separate testing dataset.

3.5 Building Graphical User Interface

For the ease of the end user, a GUI will be developed using which the user may get the proper result and can also use it to get recommended remedy for certain diseases. This module is a tkinter based Graphical User Interface that takes the input image from the user and give it to the model which would then tell us whether the plant is healthy or diseased and if diseased what may be the possible solution for it.

4. PERFORMANCE MATRIX

Now to check for the predictability and reliability of the model we have used some performance matrix, which are as follows:

a) Confusion Matrix

It's a metric for testing machine learning classification problems with two or more groups as output. There are four different variations of expected and real values.

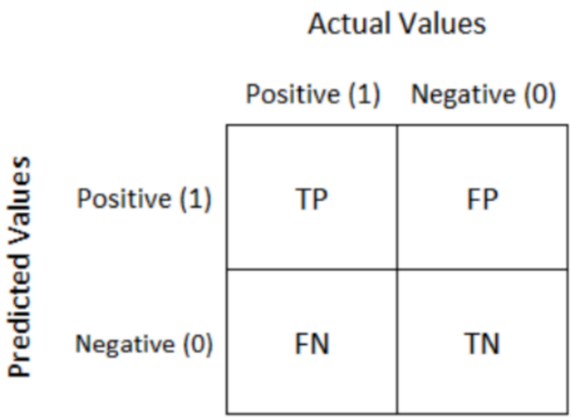

Fig -3: Confusion Matrix

True-positive (TP): model predicted true & is correct.

True-negative (TN): model predicted false & is correct.

False-positive (FP): model predicted true & is wrong.

False-positive (FP): model predicted false & is wrong.

b) Accuracy

The proportion of overall predictions that are correct. Otherwise, how much the model accurately forecasts true and non-true.

$$\text{Accuracy} = \frac{(TN+TP)}{(FN+TN+TP+FP)}$$

c) Precision

Ratio of correctly predicted +ve values to overall predicted +ve values.

$$\text{Precision} = \frac{TP}{(TP+FP)}$$

d) Recall (or Sensitivity)

proportion of correctly predicted positive observed values.

$$\text{Recall} = \frac{TP}{(TP+FN)}$$

e) F1 Score

It is the weighted average of Recall and Precision.

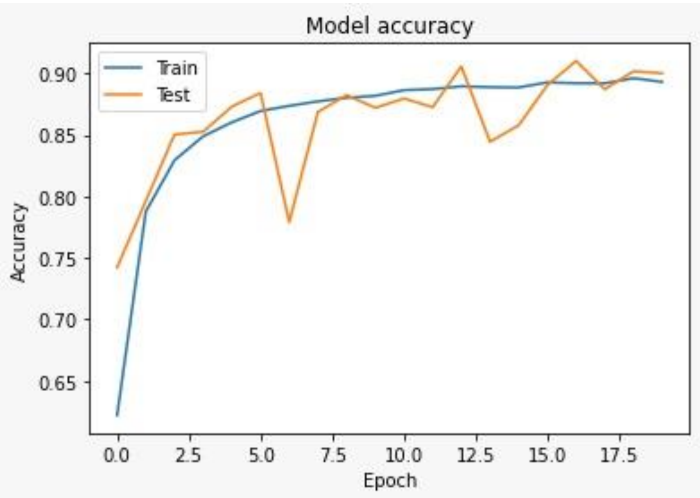

Fig -4: Model Accuracy

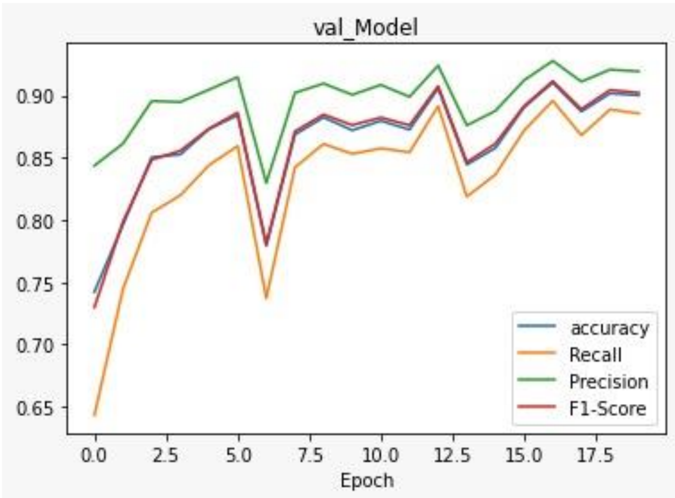

Fig -5: Model Performance

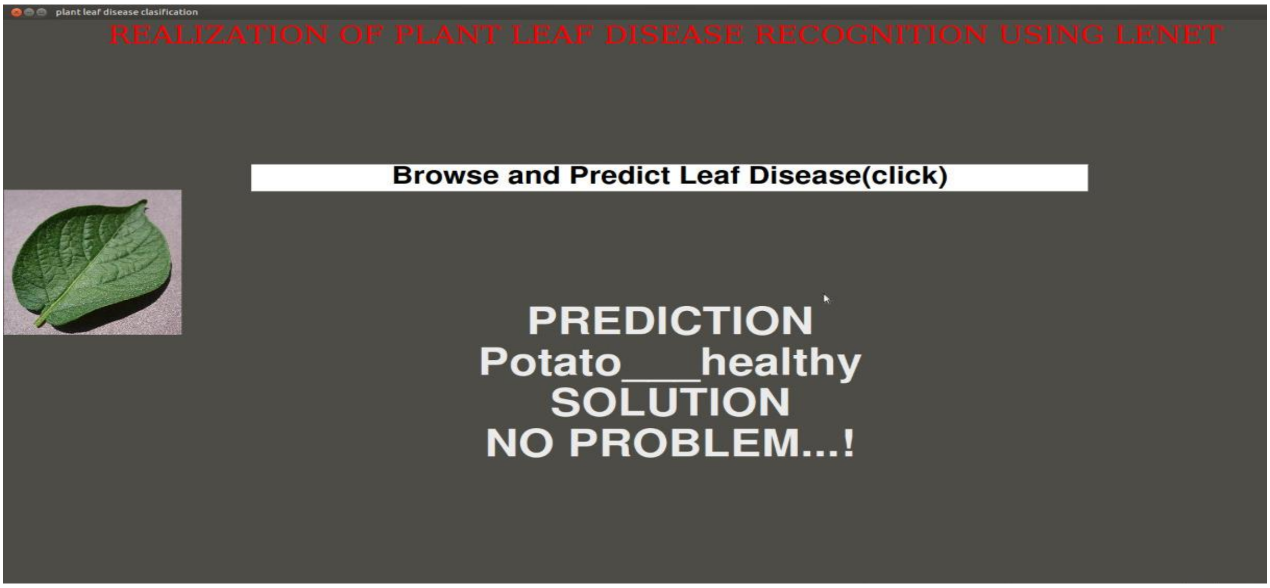

Fig -4: Program Execution (GUI Output)

5. CONCLUSION

In this paper, a study was present on plant leaf disease detection where we have been successful in implementing the LeNet-5 model onto the TensorFlow back-end to and have classified plant diseases. It infers that the proposed model optimizes the system with a great 88% accuracy and gives better result than expected. We also implemented a test of CNN and Alex-net model on the dataset. However, they give accuracy lower than the LeNet - 5 Model. This brings some of the following insights about plant leaf disease prediction.

REFERENCES

[1] Sachin D. Khirade, Plant Disease Detection using Image Processing, ICCCCA, 2015.

[2] Andreas Kamilaris and Francesc X. Prenafeta-Boldú, “Deep learning in agriculture: A survey”, Elsevier Journal, Computers and Electronics in Agriculture 147 (2018) p.70–90.

[3] Rakshitha N, Rekha H.S, Sandhya S, Sandhya V and S. Sowndeswari, Pepper Cutting UGV and Disease Detection using Image-processing, IEEE, 2nd Intl. Conf. on Recent Trends in Electronics Information & Communication Technology (RTEICT), May 19-20, 2017.

[4] Xiaoyan Guo, MingZhang, Yongqiang Dai, plant disease segmentation model based on pulse-coupled neural net with shuffle frog-leap algo, 14th International Conference on Computational Intelligence and Security (CIS), 2018.

[5] Nikos Petrellis, A Smart Phone Image Processing Application for Plant Disease Diagnosis, 6th International Conference on Modern Circuits and Systems Technologies (MOCAST), 2017.

[6] Chit Su Hlaing, Sai Maung Maung Zaw, Plant disease recognition for Smart Farming Using Model-based Statistical Features, IEEE 6th Global Conference on Consumer Electronics (GCCE), 2017.

[7] H. Poojary, A Survey on Plant Disease Detection using Support Vector Machine, International Conference on Control, Power, Communication and Computing Technologies (ICCPCCT), 2018.

[8] Vitthal S. Gutte, A Survey on Recognition of Plant Disease with Help of Algorithm, International Journal of Engineering Science and Computing (IJESC), June 2016.

[9] Vipinadas. M. J, A Survey on Plant Disease Identification, International Journal of Computer Science Trends and Technology (IJCST) – Volume 3 Issue 6, Nov-Dec 2015.

[10] Prof. Patil Ashish and Patil Tanuja, “Survey on Detection and Classification of Plant Leaf Disease in Agriculture Environment”, International Advanced Research Journal in Science, Engineering and Technology, Vol.4, Sp. Issue 4,Jan-2017.

[11] Arpita Patel and Mrs. Barkha Joshi, “A Survey on the Plant Leaf Disease Detection Techniques”, International Journal of Advanced Research in Computer and Communication Engineering, Vol. 6, Issue 1, January 2017.