Corn Leaf Disease Detection Using Deep Learning

Zaid Khan, Saud Ahmed Ansari, Russel Fernandes, Jaya Jeswani
Xavier Institute of Engineering Mumbai, India

***Abstract*— The leaf disease detection system is designed to identify and provide a solution for any disease that may be present in a leaf or plant. This is especially crucial in developing countries where agriculture plays a significant role, and it is essential to recognize unhealthy plant leaves and classify the disease to prevent significant loss of plants. By providing faster and more accurate results, farmers can reduce their losses. The process of discovering the type of disease involves four stages: image pre-processing, feature extraction, and classification. Image pre-processing is used to enhance the quality of the image, and for classification, a Convolution Neural Network (CNN) is utilized, which includes various layers that aid in prediction. Finally, a cure is suggested to the user during the terminal stage**

***Index Terms*—Convolutional neural network (CNN), Feature Extraction, Classification.**

I. INTRODUCTION

The appearance of plants is a crucial qualitative factor in agriculture, as it greatly impacts the buying and selling behavior of goods. As such, grading systems and quality control are essential in developing healthy plants. Diseases in plants can lead to significant economic losses, making illness management a challenging task. Symptoms such as colorful spots or streaks are typically visible on the leaves or stems of the plants, which are caused by various kinds of fungi, bacteria, and viruses. However, some diseases may not show visible symptoms initially, resulting in societal and economic losses. To prevent certain diseases, image processing can recognize the color aspects of leaves and aid in disease identification, reducing human effort and providing accurate results. The purpose of this paper is to review different techniques for plant disease detection and analyze them based on various parameters. The paper is divided into four sections: the first provides a brief introduction to the significance of plant disease detection, the second reviews existing work in this area, the third outlines the basic methodology for developing a disease detection system, and the final section concludes the paper.

II. LITERATURE REVIEW

[1] Mohanty et al focused their study on two CNN architectures, namely GoogleNet and AlexNet. They utilized the PlantVillage dataset, which included around 54,306 images of various plants from 38 different classes of diseases. The images were resized to 256 x 256 pixels before performing predictions and model optimization. The entire process was executed for a total of 30 epochs, and the dataset was divided into training and testing in a 1:4 ratio.

[2] Kulkarni et al. developed a methodology that accurately detects diseases in plants through the use of artificial neural networks and various image processing techniques. Their method achieved a recognition rate of up to 91%, using an ANN classifier for disease classification and the Gabor filter for feature extraction. The ANN classifier distinguishes various diseases in plants and leverages a combination of color, features, and texture to identify them.

[3] Vijayaraghavan et al. stated that support vector machines are powerful AI methods that can be employed to solve classification problems. Support Vector Regression (SVR) is a type of SVM used to address regression problems. Researchers often prefer SVR due to its ability to provide generalization capability to the solution model.

[4] Gayatri Kuricheti et al. developed an effective algorithm for detecting and preventing the spread of crop diseases. The authors employed K-Means image segmentation and GLCM for texture analysis of leaf images. They used an information gain algorithm to rank attributes and then used an SVM classifier to classify the features extracted from the images. However, K-Means clustering has a limitation in that it forms a fixed number of clusters, with three clusters being ideal, representing the healthy part of the leaf, infected part, and the image background.

[5] Peng Jiang et al. presented a dataset of real-world images of infected apple leaves and proposed a model for detecting various diseases, including Grey spots, Brown spots, Alternaria leaf spots, Mosaic, and Rust. They employed data augmentation and annotation techniques to prepare the dataset and trained their model using a dataset of 26,377 images of diseased apple leaves. The authors proposed deep CNNs using Rainbow concatenation and the Google Net Inception structure for the detection of apple leaf diseases

[6] Khamparia et al. combined autoencoders with convolutional neural networks (CNN) for the detection of crop diseases. The authors used a dataset of 900 photos of three crops with five distinct forms of diseases, including early and late blight in potatoes, leaf molds and yellow leaf

curl in tomatoes, and rust disease in maize. The accuracy of the model varied for different convolution filters of sizes 2 X 2 and 3 X 3 with different numbers of epochs, and the Adam optimizer was used to reduce loss and enhance accuracy during training.

[7] In 2019, Kamal et al. introduced two models, Modified MobileNet and Reduced MobileNet, based on the depthwise separable convolution architecture, and compared their performance to MobileNet, AlexNet, and VGG. The authors used 82,161 photos from the publicly available PlantVillage dataset with 55 unique classes of healthy and diseased plants for the training and testing of the models. The authors also employed different optimizers, including SGD, Adam, and Nadam, and found that Nadam outperformed the other two optimizers in terms of performance and convergence rate

.

[8] In 2019, G. Geetharamani and A. Pandian proposed a plant leaf disease identification strategy using a deep convolutional neural network. The authors trained and tested their deep CNN architecture using open-source data obtained from the PlantVillage dataset, consisting of 54,448 photos of 13 different plant leaves. The authors used a combination of augmented and nonaugmented image datasets and applied techniques such as gamma correction, color augmentation, noise injection, rotation, PCA, and picture flipping to construct the enhanced images, which increased the size of the augmented dataset to 61,486 images.

III. PROPOSED METHODOLOGY

The block diagram depicted in Figure 1 illustrates three phases of the system. The first phase is responsible for collecting and transmitting data, such as images from digital cameras or mobile phones, with desired resolution and size. The dataset images are then selected and prepared for the learning process to achieve high accuracy of the model architecture. In the third phase, the model is trained using the training dataset, and its accuracy is validated using different images. The model is then deployed on a web page for practical use. By uploading an image from a digital camera, the CNN algorithm is applied to determine whether the plant is healthy or infected and to classify the disease. The system provides appropriate measures to help farmers deal with the disease based on its type.

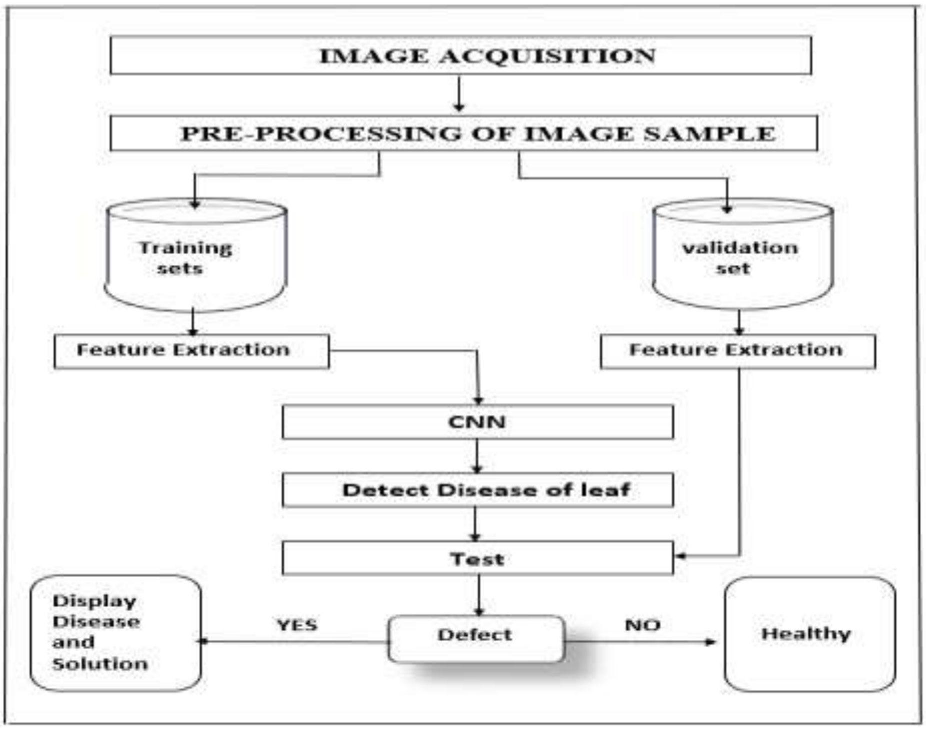

Fig.1 Block Diagram

A. CNN MODEL

To detect and classify diseases, the implemented model is a CNN. The model comprises four layers: Input, Convolutional, Pooling, Fully connected, and Output layers. To implement the model, Python programming language and the TensorFlow library are used. TensorFlow is a platform that provides an interface for defining machine learning algorithms and executing them. The detection system's implemented model is shown in Table I and Fig. 2.

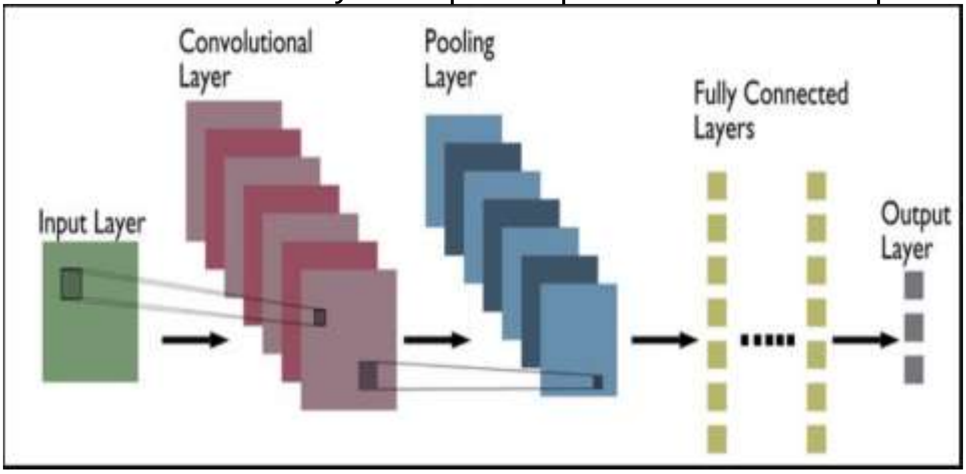

Fig 2 CNN model architecture

The CNN model utilized in this study includes four convolutional layers that are responsible for extracting feature maps from input images by applying filters. The size of the filters used in the model varies across layers, with a 7x7 filter used in the first layer to capture general features of the input image, and progressively smaller 5x5 and 3x3 filters used in subsequent layers. The Softmax activation function is applied to the output matrix of the convolutional layers. Max pooling with a filter size of 5x5 is applied to each pooling layer to condense the output matrix from the previous convolutional layer. The output layer of the model is activated using the Softmax function, containing three neurons. The TensorFlow and Keras libraries, implemented using the Python programming language, were utilized to design and implement the model architecture, which was optimized for high graphical and computing power. This approach can be useful in developing automated systems for detecting plant diseases, allowing farmers to identify and treat diseased crops early and improve crop yields.

Table I - CNN Model Details

Layer Type	Weight Filter size	Output Shape
Input Layer	-	3x256x256
Convolutional Layer 1 Pooling Layer 1	7x7 3x3	32x250x250 32x83x83
Convolutional Layer 2 Pooling Layer 2	5x5 2x2	32x79x79 32x39x39
Convolutional Layer 3 Pooling Layer 3	3x3 2x2	64x37x37 64x18x18
Convolutional Layer 4 Pooling Layer 4	3x3 2x2	128x16x16 128x8x8
Fully Connected Layer 1	-	128
Fully Connected Layer 1	-	128
Output Layer	-	3

B. Dataset

In this study, the dataset utilized comprises of images of corn leaves sourced from the Kaggle dataset. The classification task involves only three classes, namely Northern Early Blight-infected, Common Rust-infected, and healthy corn leaves. Fig. below exhibits instances of healthy leaves and infected leaves obtained from the dataset. To help illustrate the differences between healthy and infected corn leaves, the study likely includes visual examples of these leaves in their respective classes.

Northern Leaf Blight

Common Rust

Healthy Leaf

To develop the proposed system, a dataset of 3000 colored leaf images with a resolution of 256x256 pixels was used. The dataset consisted of 1100 images of leaves infected with Common Rust, 900 images of leaves infected with Northern Leaf Blight, and 1000 images of healthy leaves. To prepare the dataset for training and testing, images with the same class were grouped together in a file named after the class name. Next, the Keras preprocessing library was used to resize all images to 256x256 pixels. The images were then converted into arrays to match the input layer of the CNN model. After that, each image was labeled with its corresponding class name. Finally, the dataset was split into training, validation, and testing sets.

C**.** Testing and Training Method

The training process is the main step in the CNN model. It is responsible for adjusting the network with the input dataset to generate the corresponding labels. The dataset is divided into 70% for training, 25% for validation, and 5% for testing the model. The model is trained using 70% of the dataset, which is approximately 2100 images, for 20 epochs. Each epoch includes a forward pass of all training data and a reverse pass. The validation accuracy is computed after the testing data is fed into the model. Finally, the overall training accuracy and validation accuracy are calculated. The model is coded in Python language and various libraries are used to implement it, such as Keras for building the model, CV2 or Opencv2 for image processing, NumPy for handling arrays, and Matplot for data visualization. Google Collab is utilized to train and build the model due to its powerful GPU and computing power, which include 2 CPU cores, 14 Gigabytes of RAM for GPU, 12 Gigabytes of RAM, and 78 Gigabytes of disk space.

D. Proposed Model Validation and Testing

The proposed model's desired performance accuracy was achieved by using around 2100 different leaf images for the training process. The training set included 50 healthy leaf images, 55 images infected by Common Rust, and 45 images infected by Northern Leaf Blight used for model testing. The model was evaluated for every epoch to calculate the validation accuracy, and the validation and training accuracies for 20 epochs are shown. The total accuracy for the proposed model is 98.2%, which is calculated after the completion of all the epochs. The algorithm used to categorize corn leaf images into healthy, Northern Leaf Blight, and Common Rust achieved higher accuracy with the VGG19 architecture compared to VGG16. Additionally, the training time for the VGG19 architecture was less than that of VGG16. Table II presents a comparison between the two architectures, VGG16 and VGG19.

Architecture	Dataset Size	Training-Loss	Validation-loss	Training-accuracy	Validation-accuracy
VGG16	300	0.45	0.35	84	79.2
	1500	0.33	0.17	95.2	93.5
	3000	0.2	0.15	96	94.1
VGG19	300	0.4	0.31	90.4	82.8
	1500	0.18	0.03	97.8	96.2
	3000	0.12	0.011	98.7	96.5

Table II

Based on the outcomes of the proposed model, it can be inferred that identifying corn diseases necessitates considering color, shape, and texture as essential factors. The CNN model shows its potential in identifying such features. Enhancing the model's performance could be achieved by diversifying the training data to incorporate images captured in uncontrolled settings. Thus, it is crucial to provide such resources in upcoming research studies, as these findings emphasize their significance.

E. MODEL DEPLOYMENT

The weights of the CNN network were saved for deployment on various devices and to minimize the computing power required for utilizing the trained model. Once the model was saved, it was downloaded onto a computer with two CPU cores and four Gigabytes of RAM. Subsequently, the model was deployed as an API on the webpage, and an uploaded image was transformed into a POST request. The result was displayed on the webpage via the GET request. The deployed model is presented in Fig. 4.

IV. CONCLUSION

This paper presents a solution to tackle crop diseases, which can cause significant losses in agricultural production up to 25%. The proposed system is based on Convolutional Neural Network technology and a bespoke model architecture, aiming to detect infections in corn crops such as Northern Leaf Blight and Common rust early, thus minimizing the plant production loss. The model is trained and evaluated using a dataset of 3000 colored corn leaf images, divided into three parts for validation purposes: 70% for training, 25% for validation, and 5% for testing. The study finds that increasing the dataset from 1500 to 3000 images doesn't impact the accuracy significantly, but reducing it to 300 images decreases it by 10%. The suggested model architecture outperforms other approaches with a 99.52% accuracy, compared to the VGG16 methodology.

REFERENCES

[1] Mohanty, S.P., Hughes, D.P., Salathé, M.: Using deep learning for image-based plant disease detection. Front. Plant Sci. **7**(September), 1–7 (2016)

[2] Kulkarni Anand H, Ashwin Patil RK. Applying image processing techniques to detect plant diseases. Int J Mod Eng Res 2012;2(5):3661–4.

[3] Vijayaraghavan Venkatesh, Garg Akhil, Wong Chee How, Tail Kang, Bhalerao Yogesh. Predicting the mechanical characteristics of hydrogen functionalized graphene sheets using an artificial neural network approach. J Nanostruct Chem 2013;3:83.

[4] Kuricheti, Gayatri, and P. Supriya. "Computer Vision-Based Turmeric Leaf Disease Detection and Classification: A Step to Smart Agriculture." In 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), pp. 545-549. IEEE, 2019.

[5] Jiang, Peng, Yuehan Chen, Bin Liu, Dongjian He, and Chunquan Liang. "Real-Time Detection of Apple Leaf Diseases Using Deep Learning Approach Based on Improved Convolutional Neural Networks." IEEE Access 7 (2019): 59069-59080.

[6] Khamparia, A., Saini, G., Gupta, D., Khanna, A., Tiwari, S., de Albuquerque, V.H.C., 2020. Seasonal crops disease prediction and classification using a deep convolutional encoder network. Circ. Syst. Sign. Proc. 39, 818–836. https://doi.org/10.1007/s00034-019-01041-0.

[7] Kamal, K.C., Zhendong Yin, Mingyang Wu, and Zhilu Wu, (2019), "Depthwise Separable Convolution Architectures for Plant Disease Classification" Computers and electronics in agriculture https://doi.org/10.1016/j.compag.2019.104948

[8] G. Geetharamani, A. Pandian, (2019), "Identification of plant leaf diseases using a nine-layer deep convolutional neural network," in Computers & Electrical Engineering. Vol. 76, Pages 323-338. https://doi.org/10.1016/j.compeleceng.2019.08.010