CROP QUALITY PREDICTION USING CONVOLUTIONAL NEURAL NETWORKS (CNNS)

Abstract

Deep learning acts as an important element for predicting crop quality, including assisting with crop selection and crop management decisions. Determining the quality of the plant as well as the type of plant based on the leaf of a plant is the goal. Dataset used is Plant Village dataset, which include images of the plants leaves like potato, tomato etc. In order to identify quality and the category of plant we can use Deep Learning, more specifically it can be done by building a convolution neural network. Crop diseases have grown significantly in recent years due to drastic climate changes and crop immunity deficiencies. This leads to widespread crop damage, less cultivation, and, ultimately, economically loss for farmers. Crop disease recognition and dealing have become a challenge as a result of rapid disease growth and insufficient farmer knowledge. The texture and visual similarities of the leaves aid in disease identification. As a result, the combination of deep learning and computer vision provides a solution for this problem. The proposed deep learning model is trained on plants images of healthy and diseased from a publicly available dataset. The model predicts the whether the images of leaves a diseased or healthy.

Keywords: Deep Learning; Convolutional Neural Networks; Plant Disease Detection.

Authors

Dr. I. Hemalatha
Professor
Department of IT
S.R.K.R. Engineering College(A)
Bhimavaram
Andhar Pradesh, India
drihl@srkrec.ac.in

Bh. Hema Sai Harshini
B.Tech (ECE)
Department of ECE
S.R.K.R. Engineering College(A)
Bhimavaram
Andhar Pradesh, India
harshinibhupathiraju2002@gmail.com

I. INTRODUCTION

Agriculture plays an important role in India's economy. This sector make available of approximately 52% of the total no of jobs in India and contributes around 18.1% to the GDP. The early prognosis of crop disease can give support to take decisions in usage of fertilizers, pesticides and to reduce the complications, which can be beneficial in agriculture [5]. This helps in early detection. The use of appropriate technology support in this regard can be extremely beneficial to field agriculture and farmers[1]. Crop quality prediction is a critical task for national and regional decision-makers in order to make quick decisions. [2]. A reliable crop quality prediction model can assist farmers in deciding what to grow, when to grow and the measures that are needed to be taken to ensure the health of the plants [3]. Proper estimation of the crops quality allows the farmers to know about the disease and then he/she will be able to take certain further prevention measures. Using Deep learning and Convolutional Neural Network preparing a tool or a model is taken into consideration to bring the work of physical or manual detection of crop quality to easy prediction system [6].

I. METHOD

1. Crop quality prediction model

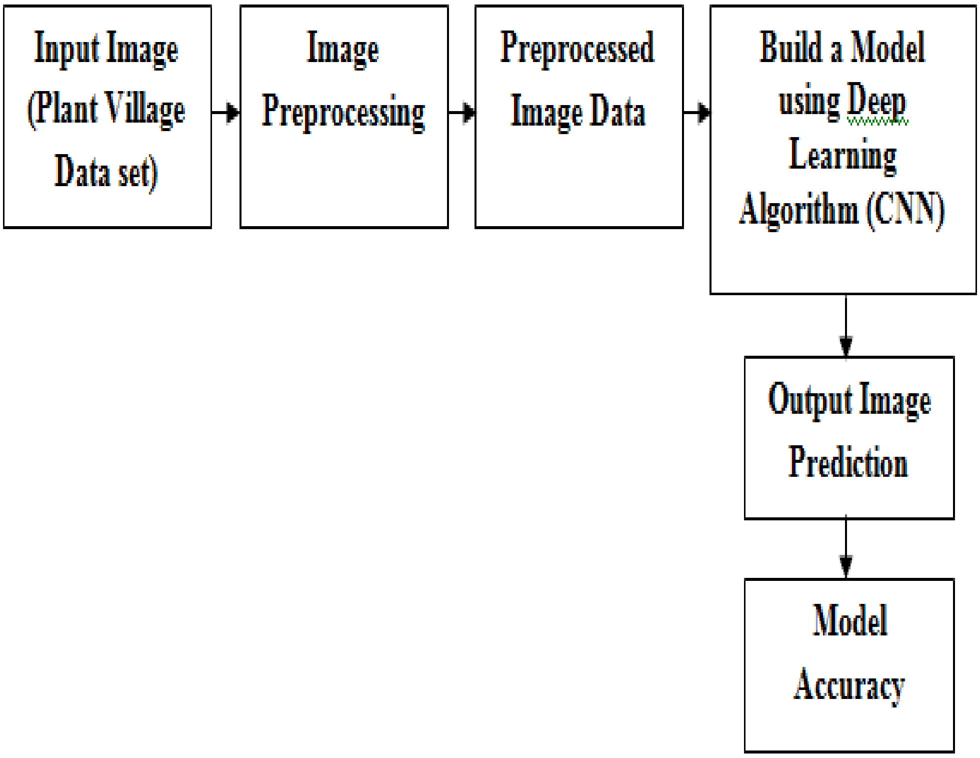

Figure1: System Architecture of Crop Quality Prediction Model

To predict whether the leaf is diseased or healthy the steps to be fallowed are shown in the Fig.1. To come up with a system that indicates the use of deep learning in crop quality detection systems which would help the farmers to know the quality of their crops which in turn helps them to take appropriate measures to protect the crops and increase their crop yield [4].

A Deep learning model is a Convolutional Neural Network (ConvNet/CNN) that have an input image, assign significance i.e learnable weights and biases to different objects in the image, and differentiate one to the other. When compared with other classification algorithms, the amount of pre-processing required by a ConvNet is significantly less. While primitive methods require hand-engineered filters, ConvNets can learn these filters/characteristics with enough training[9].

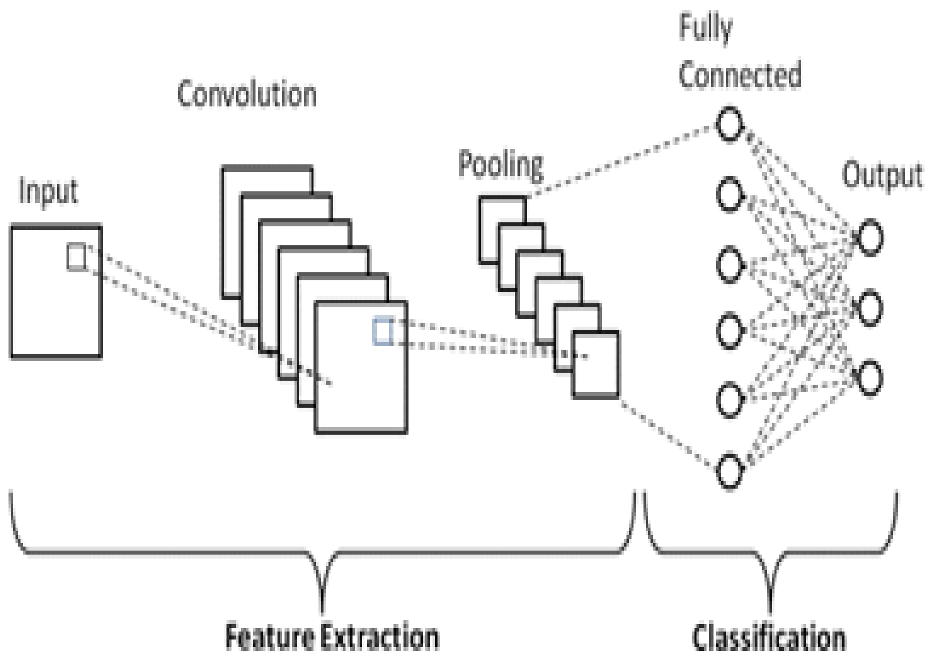

Figure 2: CNN Architecture

Figure 2 depicts the architecture of a Convent, which is similar to the connectivity pattern of neurons in the human brain and was inspired by the organisation of the Visual Cortex.

2. **Dataset description:** The Dataset consists of diseased plant leaf images with corresponding labels is used from the Plant Village Dataset available with different types of crops Tomato, Potato, Bell Pepper leaf images a total number of images 20,620 in 15 number of directories.https://www.kaggle.com/datasets/emmarex/plantdisease

Pepper__bell___Bacterial_spot

Potato___Early_blight

Potato___healthy

Tomato_Early_blight

Tomato_Leaf_Mold

Tomato_healthy

Figure 3: Leaf images of plants with labels

In this proposed system the dataset is separated into training, testing and validation datasets. As shown in the Fig.3 the data set consists of diseased plant leaf images with corresponding labels. The data set is taken as Training dataset is 60%, Testing dataset is 20% and Validation dataset is 20% of the data.

3. **Pre-process the data:** Data pre-processing is used to transform the raw data in a useful and efficient format. Because image data cannot be directly fed into the model, we must perform some operations and process the data before it can be used by our neural network. The data is normalised in this step. Range of Pixel values is between 0 and 256. Each value corresponds to a different colour. The computation of high numeric values can be complex while using the image as it and passing through a DNN model.

We can reduce this by converting the values to a 0 to 1 scale. The numbers will be smaller as a result, and the computation will be easier and faster. Apart from 0, the pixel values range from 0 to 256. So dividing all the values by 255 converts them to a range from 0 to 1. The following Fig. 4 shows the output of pre-processed images found from the 15 classes image dataset.

```
training = ImageDataGenerator(rescale = 1/255)
validation = ImageDataGenerator(rescale = 1/255)

```

```
train_dataset = training.flow_from_directory("processed_data/train/",
                                              target_size = (200,200),
                                              batch_size = 32 ,
                                              class_mode = 'categorical'
                                             )

validation_dataset = validation.flow_from_directory("processed_data/val/",
                                              target_size = (200,200),
                                              batch_size = 32 ,
                                              class_mode = 'categorical'
                                             )


```

```
Found 12378 images belonging to 15 classes.
Found 4121 images belonging to 15 classes.
```

Figure 4: Pre-processing the data

4. Pseudo code for training and testing images

Pseudo code for training images:

- Labelled images of leaves
- for Every image in training image in dataset do
  - I ← image
- Pre-processing
  - I resized ← Image Resizing (I)
  - I feature extraction ← feature extraction(I resized)
- Pass this instance to the pre-trained model Get the output
- Calculate the loss
- Update weights based on loss for n number of epochs end for save the model trained

Pseudo code for testing images:

- Labelled images of Leaves
- Input image I Pre-processing
  - I resized ← Image Resizing(I)
  - I feature extraction ← feature extraction(I resized)
- The testing image will be classified based on the trained model
- The output will be displayed

5. **Build a model using deep learning algorithm (CNN):** Convolution and pooling layers are typically found in CNN models. CNN performs effectively for challenges involving picture classification because it performs better for data that are represented as grid structures. Convolutional layers are the layers in a deep CNN where filters are applied to the original image or other feature maps. Pooling layers are used to reduce the dimensions of the feature maps. Thus, it reduces the number of parameters to learn and the amount of computation performed in the network. The pooling layer summarizes the features present in a region of the feature map generated by a convolution layer.

   The building model will then compile using the RMSprop optimizer, as shown in Fig. 5. Optimizers are algorithms or techniques that modify your neural network's weights and learning rates in order to reduce losses. You can tune a plethora of hyper parameters to improve the performance of your neural network. However, not all of them have a significant impact on network performance. The optimizer you choose could mean the difference between your algorithm converging or exploding. The vertical oscillations are limited by the RMSprop optimizer. As a result, our learning rate can be increased, and our model can take larger steps in the horizontal direction, converging faster.

```
model = tf.keras.models.Sequential([tf.keras.layers.Conv2D(16, (3,3), activation='relu', input_shape=(200,200,3)),
                                    tf.keras.layers.MaxPool2D(3,3),
                                    #
                                    tf.keras.layers.Conv2D(32, (3,3), activation='relu'),
                                    tf.keras.layers.MaxPool2D(3,3),
                                    #
                                    tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
                                    tf.keras.layers.MaxPool2D(3,3),
                                    #
                                    tf.keras.layers.Conv2D(96, (3,3), activation='relu'),
                                    tf.keras.layers.MaxPool2D(3,3),
                                    ##
                                    tf.keras.layers.Flatten(),
                                    ##
                                    tf.keras.layers.Dense(512, activation='relu'),
                                    ##
                                    tf.keras.layers.Dense(15, activation='softmax')
                                   ]
                                   )
```

```
model.compile(loss = 'categorical_crossentropy',
              optimizer = RMSprop(lr=0.001),
              metrics = ['accuracy']
             )
```

Figure 5: Building the model and applying the optimizer

- **Train the model:** We use a CNN for model design. We also specified the filter size and shape for the Conv and Pool layers. The model makes use of ReLU and Softmax activation. The model.fit() function of the Keras will start training of the model. It gets the training data, validation data, epochs, and batch size.

6. **Evaluate the model:** A 20,620-image input data set is utilised to assess how well our model performs. As a result, the testing data is new data for our model since it was not used to train the data. Finally the main aim of our model is to detect the quality of the crop whether it is diseased or healthy using CNN model which is as depicted in the Figure 6.

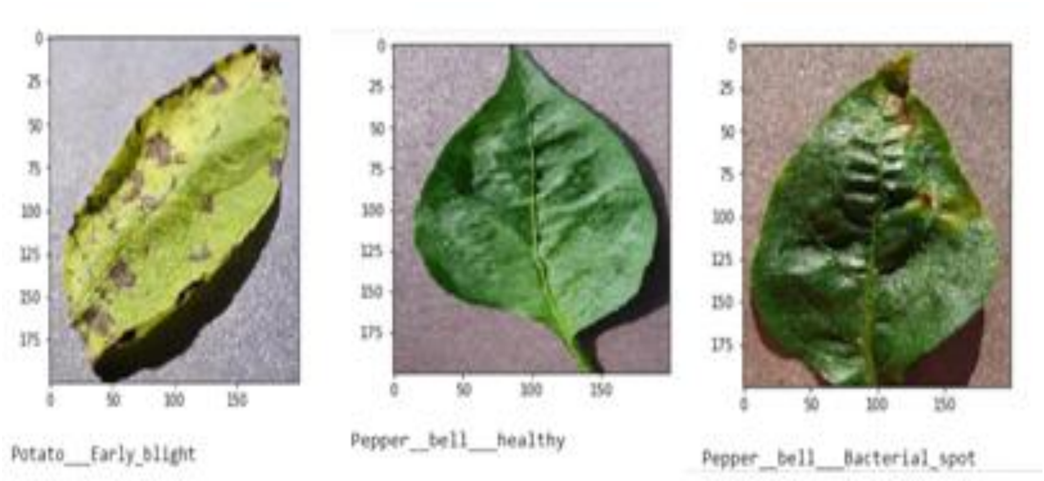

Figure 6: Evaluation of the model

7. **Experiments and results:** Accuracy is a metric used to illustrate how well the model will perform across all classes in general. [10].

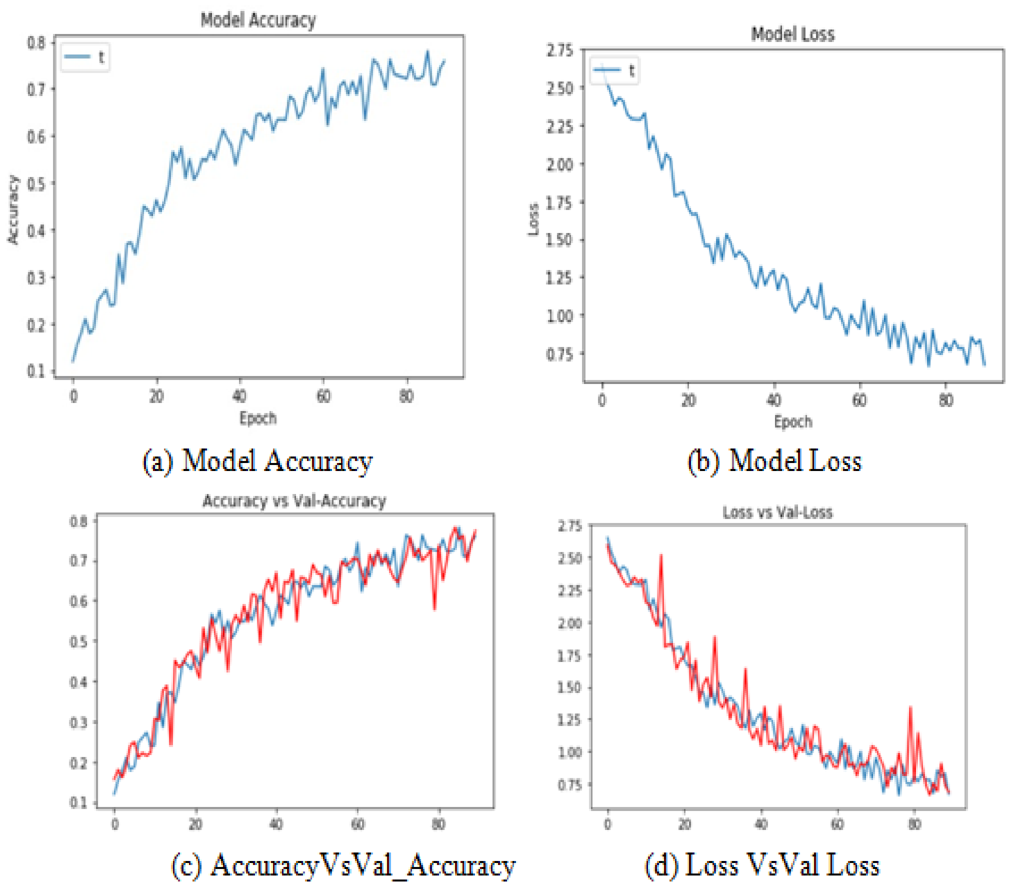

Figure 7: Shows the (a) Model Accuracy (b) Model Loss (c) Accuracy vsVal_Accuracy (d) Loss VsVal_Loss

When all classes are equally important, it is beneficial. It is determined by dividing the number of correct predictions by the total number of predictions.

IV. CONCLUSION

Crop farming was previously made on the source of the farmer's hands-on skill. On the other hand, change in climate and new types of diseases have impact to change crop yields. Therefore, farmers may not be capable to know the quality of their crops based on the process of manual prediction. The quality of the crops of land has, more often than not, resulted in failure. Exact crop quality prediction increases crop production because when the quality of the crop is known appropriate measures can be taken. where Deep Learning plays a major role in prediction of crop quality. Deep Learning algorithms work well when the data is huge. Hence, using CNN for crop quality prediction using PlantVillage dataset gives good results. Convolutional neural network trained for identifying and recognising plant leaf disease could correctly categorize and predict the diseases for almost all images with few anomalies, achieving 80% model accuracy as shown in Figure 7.

REFERENCES

[1] Van Klompenburg, Thomas, AyalewKassahun, and CagatayCatal. "Crop yield prediction using machine learning: A systematic literature review." Computers and Electronics in Agriculture177(2020):105709.
[2] Nischitha, K., Mahendra N. DhanushVishwakarma, and Manjuraju MR Ashwini. "Crop prediction using machine learning approaches." Int. J. Eng. Res. Technol.(IJERT) (2020): 23-26.
[3] Medar, Ramesh, Vijay S. Rajpurohit, and ShwetaShweta. "Crop yield prediction using machine learning techniques."2019 IEEE 5th International Conference for Convergence in Technology (I2CT).IEEE, 2019.
[4] Khaki, Saeed, and Lizhi Wang. "Crop yield prediction using deep neural networks." Frontiers in plant science 10 (2019): 621.
[5] Sharma, Bhawana, Jay Kant Pratap Singh Yadav, and SunitaYadav. "Predict crop production in India using machine learning technique: a survey." 2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO). IEEE, 2020.
[6] GarimaTripathi, Jagruti Save, “AN IMAGE PROCESSING AND NEURAL NETWORK BASED APPROACH FOR DETECTION AND CLASSIFICATION OF PLANT LEAF DISEASES,” Int. J. Comput. Eng. Technol. IJCET, vol. 6, no. 4, pp. 14–20, Apr. 2015.
[7] S. Arivazhagan, R. NewlinShebiah, S. Ananthi, S. Vishnu Varthini, “Detection of unhealthy region of plant leaves and classification of plant leaf diseases using texture features,” AgricEngInt CIGR J., vol. 15, no. 1, pp. 211–217, Mar. 2013.
[8] [8]. Prof. Sanjay B. Dhaygude, Mr.NitinP.Kumbhar, “Agricultural plant Leaf Disease Detection Using Image Processing”IJAREEIE, vol. 2(1),pp. 599-602, January 2013.
[9] K. Muthukannan, P. Latha, R. PonSelvi and P. Nisha, “CLASSIFICATION OF DISEASED PLANT LEAVES USING NEURAL NETWORK ALGORITHMS,” ARPN J. Eng. Appl. Sci., vol. 10, no. 4, pp. 1913–1918, Mar. 2015.
[10] Md. Nazrul Islam, M.A. Kashem, MahmudaAkter and Md. JamilurRahman, “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.

[11] HrishikeshKanjalkar P. and Prof. Lokhande S. 2013. Detection and Classification of Plant Leaf Diseases using ANN. International Journal of Scientific & Engineering Research. ISSN: 2229 5518.
[12] SuhailiBeerankutty. and Noor ezan Abdullah. 2013. 'Classification of Watermelon Leaf Diseases Using Neural Network Analysis', IEEE Business Engineering and Industrial Applications Colloquium (BEIAC). International Journal of Engineering Science and Computing, March 2017 5328 http://ijesc.org/.