Plant Disease Detection using Deep Learning

Gourav Agrawal
Student
School of Computer Science Engineering and Information Systems
Vellore Institute of Technology
Vellore, Tamil Nadu, India
gourav.agrawal2023@vitstudent.ac.in

Rohit Singh
Student
School of Computer Science Engineering and Information Systems
Vellore Institute of Technology
Vellore, Tamil Nadu, India
rohit.singh2023@vitstudent.ac.in

Shubham Bharadwaj
Student
School of Computer Science Engineering and Information Systems
Vellore Institute of Technology
Vellore, Tamil Nadu, India
shubham.bharadwaj2023@vitstudent.ac.in

Dr. Ephzibah E.P
Associate Professor
School of Computer Science Engineering and Information Systems
Vellore Institute of Technology
Vellore, Tamil Nadu, India
ep.ephzibah@vit.ac.in

***Abstract* – Agriculture involves the cultivation of crops, the rearing of livestock, and the responsible management of natural resources to generate food, fiber, and other crucial items for both human consumption and industrial purposes. However, the agricultural industry faces the challenge of plant diseases that damage the crop yields and their quality by approximately 40 percent across the world. To address this challenge, we present an effective approach using Convolutional Neural Networks (CNNs) for plant disease detection. CNN proves to be a powerful deep learning model suitable for tasks like image classification, object detection, and image segmentation. Our research begins by utilizing a dataset containing images depicting plants affected by diverse diseases. This dataset lays the groundwork for validating our disease detection model based on CNN. In conclusion, this research paper provides a holistic view of various plant disease detection by making use of a few Deep Learning (DL) techniques and frameworks as required. It shows the potential of this technology to revolutionize agriculture by identifying diseases accurately, which helps farmers to protect their crops and increase agricultural productivity.**

***Keywords* - CNN, DL, VGG-19**

I. Introduction

The Indian economy greatly depends on the agricultural sector. Around 50% of the country's workforce is engaged in farming, and the financial well-being of farmers depends on the quality of their crops, which, in turn, relies on the health of the plants. Detecting plant diseases early is crucial for ensuring a successful harvest. Advanced tools like deep learning are necessary for spotting these diseases in the initial stages. This research aims to categorize plant diseases by examining images of leaves through Deep Learning. To address this challenge, we have tried to present a viable approach by incorporating Convolutional Neural Networks (CNNs) for the identification of plant diseases. Remote sensing technology can also be used, which is the future scope for this work, to detect plant diseases, which enables real-time crop management and monitoring across wide areas.

II. Literature Survey

In [1], according to what the author saw in past studies, depending solely on machine learning techniques doesn't efficiently extract features automatically. On the other hand, employing deep learning is computationally demanding and requires extensive data for effective training and performance. In response to these challenges, the author presents a novel framework that combines the merits of both machine learning and deep learning. This framework encompasses 40 unique hybrid deep learning models, integrating eight different adaptations of pre-trained deep learning architectures (such as Efficient Net for feature extraction) along with diverse machine learning techniques like k-Nearest Neighbors, Logistic Regression, etc., for classification. The effectiveness of this approach was evaluated using a real-time image dataset depicting tomato early blight disease, achieving impressive accuracy ranging from 87.55% to 100%. Additionally, they validated the proposed framework using openly available datasets.

The author in [2] introduces a model that initiates importing a dataset incorporating images of both diseased and healthy plants. During image acquisition, various characteristics are transformed into a digital format, and after preprocessing to enhance clarity, the images are simplified for analysis. In the context of image classification, the author utilizes CNN, a Deep Learning algorithm, to differentiate between healthy and diseased plants, pinpointing specific illnesses. A dataset comprising 4352 images, evenly distributed with 300 images in each category, is prepared for examination. Prior to being split into an 80:20 ratio for training and testing the CNN model, the data undergoes shuffling, labeling, and normalization processes to ensure randomness. The proposed model is compared with other methods, revealing a higher accuracy of approximately 97.5%.

In [3], the authors labeled the CNN model they employed as MCNN (Multilayer Convolutional Layer Network)

specifically designed for classifying various plant images. MCNN is essentially a CNN model inspired by the architecture of AlexNet. Ferentinos trained multiple CNNs, including AlexNet and GoogleNet, using a database featuring 58 different plant permutations and various diseases. The reported accuracy for this model is 97.13%.

In [4], the authors assess the efficiency of machine learning techniques (such as Support Vector Machine (SVM) and Random Forest (RF)) in comparison to deep learning methods (including Inception-v3 and VGG-16) for detecting diseases in citrus plants. To assess the performance of these models, they initiated the process with a dataset focused on citrus leaf diseases. Employing multi-class classification, they utilized both machine learning and deep learning classifiers for prediction. The results indicated that, in this context, deep learning approaches surpassed the performance of machine learning methods.

III. Methods and Material

A. Dataset

The dataset utilized in this research is the Plant Village Dataset from kaggle, which consists of images featuring both healthy and diseased plants. These images are diverse, including color, grayscale, and segmented variations.

B. Model

Convolutional neural networks (CNNs) are a specific category within deep neural networks frequently employed for the processing and interpretation of visual data, encompassing images and videos. CNNs have exhibited notable efficiency in various tasks, such as object identification, facial recognition, and image categorization. The architecture of CNN typically comprises multiple layers, each serving a distinct purpose. CNN is like a stack of filters for images. It starts with **recognizing basic shapes**, then combines them to **understand complex patterns**, adds some **decision-making** with an activation function, **simplifies things** with pooling, connects the dots with **fully connected layers**, and gives you the result. Fig. 1 provides a clearer insight into the CNN architecture.

In this research, we applied VGG19, which is a part of the VGG (Visual Geometry Group) model series. VGG19 is an extension of VGG16, featuring a total of 19 layers, including 16 convolutional layers and 3 layers as fully connected layers.

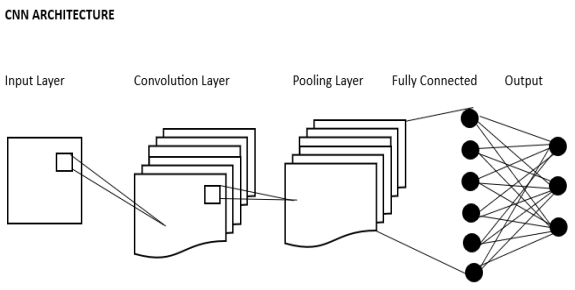


Fig. 1

IV. Methodology

A. Preprocessing

It commonly involves the removal of incorrect images or the conversion of images into a standardized format, resulting in minimizing bias during the training and testing phases of the model. Examples of images from the dataset are displayed in Fig. 2.

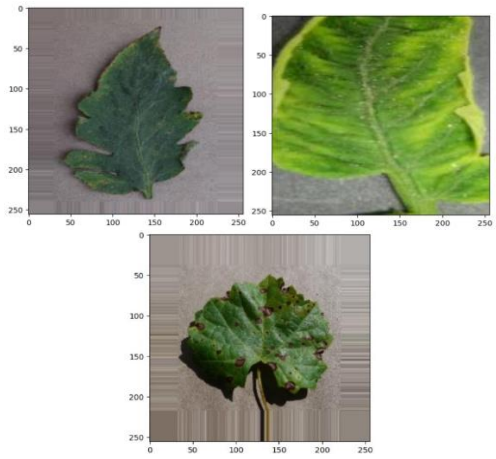


Fig. 2: Visual Representations

B. Train Test Split

We segmented the dataset into two parts, namely, train and valid, to facilitate the training and validation processes for our model. The technique of partitioning may vary based on the problem's characteristics. In the context of image classification, a common approach is to randomly divide the images using Python's built-in functions. Figure 3 visually depicts the division of images into training and validation datasets.

```
train = train_datagen.flow_from_directory(directory = '/content/plantvillage dataset/splitted/train',
                                          target_size = (256,256),
                                          batch_size = 32)

valid = val_datagen.flow_from_directory(directory = '/content/plantvillage dataset/splitted/val',
                                        target_size = (256,256),
                                        batch_size = 32)
```

```
Found 37997 images belonging to 38 classes.
Found 10849 images belonging to 38 classes.
```

Fig. 3

C. Importing Model

For any deep learning task, it is possible to use existing models from libraries instead of starting from scratch with the entire source code. We have used VGG19, for example, a model that can be imported from the Keras library of TensorFlow. Upon importing this model, it functions as a template for both the model's parameters and the provided data. The fully trained model is an identical replica of the trained one, saving both time and computational resources. In the implementation, the objective is not to export the

trained model, as the focus is on showcasing the maximum achievable accuracy.

D. Compiling Model

We need to compile our model to define the loss function to calculate the loss, choosing an optimizer for optimizing the model to get the best solution to a problem, for instance here we have used Adam as our optimizer and the metrics talks about what we want to calculate and, in our case, it is accuracy. We have implemented early stopping, a regularization technique employed to prevent overfitting during the training of a learner with an iterative method. Early stopping halts the iteration when there is no notable increase in the metric over a set number of iterations.

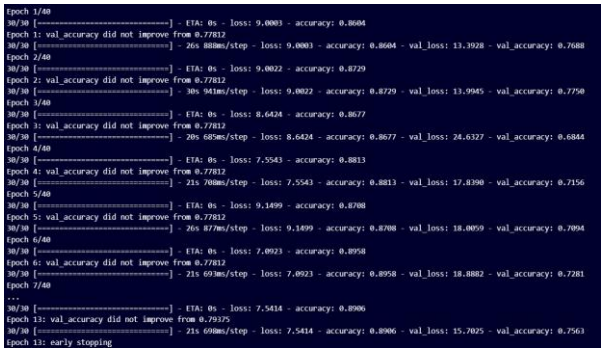
```
model.compile(optimizer='adam',loss=keras.losses.categorical_crossentropy,metrices=['accuracy'])
```

E. Training Model

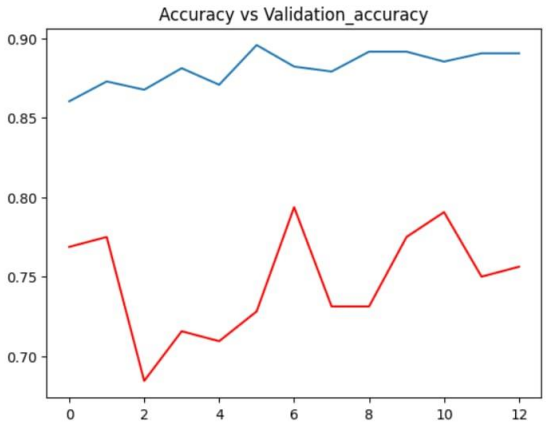
In deep learning, the training of a model involves the organization of neural networks with features to develop an algorithm capable of recognizing and learning from specific features extracted from diverse datasets, primarily composed of images. The training of the model is an iterative process commonly known as "fitting." Throughout this process, accuracy and loss are observed at each stage of training, utilizing both training and validation data.

V. RESULT DISCUSSION

A. Epoch



B. Plotting Accuracy



C. Prediction

With the completion of the training and validation tasks, we can now input an image and expect our model to predict the class (plant disease) of the provided image. The image in Fig. 4 provides insight into the predictions made by the model.

Fig. 4

VI. CONCLUSION

In this study, we successfully utilized VGG19, a component of the VGG (Visual Geometry Group) models accessible in the Keras library. We limited the epochs to 40 and set the steps per epoch to 30, resulting in an achieved accuracy of 79%. The potential extension of this work involves the development of a mobile-based application for farmers, enabling them to identify and address plant diseases at an early stage.

```
# Evaluating the Model
acc = model.evaluate_generator(valid)[1]
print(f"The accuracy of our VGG model is : {acc*100}%")
```

```
<ipython-input-75-616e10d4e644>:2: UserWarning: `Model.evaluate_generator`
  acc = model.evaluate_generator(valid)[1]
The accuracy of our VGG model is : 78.76302003860474%
```

VII. REFERENCES

1. Chug A, Bhatia A, Singh AP, Singh D. A novel framework for image-based plant disease detection using hybrid deep learning approach. Soft Computing. 2023 Sep;27(18):13613-38.

2. Hemavathi and S. Akhila, "Deep Learning Based Approach for Plant Leaf Disease Detection for Smart Farming," 2023 International Conference on Advances in Electronics, Communication, Computing, and Intelligent Information Systems (ICAECIS), Bangalore, India, 2023, pp. 496-500, Doi: a 10.1109/ICAECIS58353.2023.10170703.

3. D. Agarwal, M. Chawla, and N. Tiwari, "Plant Leaf Disease Classification using Deep Learning: A Survey," 2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 2021, pp. 643-650, doi: 10.1109/ICIRCA51532.2021.9544640.

4. Sujatha R, Chatterjee JM, Jhanjhi NZ, Brohi SN. Performance of deep learning vs machine learning in plant leaf disease detection. Microprocessors and Microsystems. 2021 Feb 1; 80:103615.