EasyChair Preprint

№ 9571

Plant Leaf Disease Detection and Classification Based on Machine Learning Model

Aashish Jha, Madhavi Purohit, Vivek Maurya and
Amiya Kumar Tripathy

January 15, 2023

Plant Leaf Disease Detection And Classification Based On Machine Learning Model

Aashish Jha, Madhavi Purohit, Vivek Maurya, Amiya Kumar Tripathy.
Department of computer engineering, Don Bosco Institute of Technology, Mumbai, India
ashish.03aj@gmail.com, maadhavipurohit@gmail.com, vivekmauryavk2003@gmail.com, amiya@dbit.in

***Abstract*— Nowadays, many fields have benefitted from the advent of new technologies, especially the technologies like data science, ML and AI, and Deep learning. Agriculture is a part of this. According to previous studies, 40–42% of agricultural production is lost (Cost: 12.42 billion euros; Source: United Nations Food and Agriculture Organization (FAO)), and the single factor contributing to this is the growing rate of loss from plant leaf diseases. This significant problem may be solved by using this approach for identifying plant leaf disease from the input photos. This procedure includes processes like image pre-processing, picture segmentation, and feature extraction. A classification method based on convolutional neural networks is then used. Plant leaf diseases were predicted with 98.3% accuracy by the proposed implementation.**

Keywords*— *Machine Learning(ML) Classification, Leaf Disease, Image Processing, Pre-processing, segmentation, Feature Extraction, pesticides for crops, prototype, diseases, Disease Classification and Detection.

I. INTRODUCTION

Agriculture is said to be the best source of income in India. It has a significant share of the country's income. Hence, in the agriculture field, the identification of diseases in plant leaves has played a major role. Because of diseases, the quality and quantity of the leaf get affected

While cultivating the crops, farmers face many challenges like leaf infection phenomena, which causes the loss of major crops, resulting in economic loss. Detection of plant diseases is an important research topic as it may prove beneficial in monitoring large fields of crops by identifying the symptoms of diseases as soon as they appear on plant leaves. Our aim is to build a model using data set containing various images of diseased plant leaves then input the image as a test dataset, identify an infected area, and build a model based on the CNN algorithm. This will help farmers to classify the disease and to take required actions.

In farming, it is very important to identify and classify the infected plant or leaf at the right time. The identification is done through physical techniques, we require the experts' help sometimes, also there is scope for errors. A human eye classifies based on color, size, etc. if these quality methods are recorded into an automatic system by using appropriate program design language then the effort will be error-free and faster.

II. RELATED WORKS

Image classification and object detection are one of the most basic processes of machine learning and neural network fields and hence they are the fundamental building blocks of any AI-based classification problem. Neural networks are a class of machine learning algorithms that are used to model complex patterns in datasets using multiple hidden layers and non-linear activation functions.

Historically, experts have used observation to diagnose plant diseases, however, there is a chance for inaccuracy due to subjective perceptivity. The machine learning algorithms are better suitable for identifying images of plants with a homogeneous backdrop. Convolutional networks and deep learning have lately made substantial progress in computer vision [8],[5].

The technique used in the paper by Malvika Ranjan et al[3]. provided a system that aids in the early detection of cotton leaf diseases. Their model has an accuracy rate of 80%. They trained neural networks using a variety of image processing techniques, such as RGB to greyscale conversion and RGB to HSV. In this study, a novel two-stage approach to leaf identification was used. The feature extraction procedure combines the two texturing approaches known as a local binary pattern (LBP) and a bag of features (BOF). The final feature set is sent to a decision-making model constructed on a multiclass support vector machine (SVM) classifier.

Another paper titled Image classification based on the boost convolutional neural network[5]. written by Shin Jye Lee et al. this paper describes how mathematics is involved in the learning of a neural network and how we can enhance the performance of the multilevel neural network by manipulating some mathematical functions i.e. to increase the accuracy of NN for multi-class classification problems. The function is activated in the second phase, and a non-linear eigenvalue is extracted by using the resulting activation value in a non-linear compression transformation. relu, sigmoid, and tanh are some of the most often utilized activation functions. As the accuracy of NN is determined by the edge weight of the edge connecting one perceptron from one to another stage, and the edge weight is set by activation functions so different edge weights can be obtained by using different activation functions. We can extend this idea over the Gradient descent of NN (learning rate) this is discussed in detail in section [IV] of this paper along with some

of the problems that can occur while training the CNN model such as the problem of the model overfitting, etc.

Another concept [9] focuses on the technique of segmenting the defective region and using color and texture as attributes For the categorization, in this case, a neural network classifier was applied. The primary benefit is that it extracts the image's chromaticity layers and classification is determined to be 97.30% accurate. The biggest drawback is that it is only utilized for a few types of crops. Seven invariant moments are used as the form parameter when GLCM (Gray-Level Co-occurrence Matrix) is used to express color characteristics in RGB to HIS. They employed an SVM classifier with MCS, which is utilized for offline disease detection in plants. A different piece of work by Ibrahim Karabayir, Oguz Akbilgic, and Nihat Tas [11] is based on the gradient-based technique (Evolved Gradient Direction Optimizer-EVGO) that is used to optimize the parameters of deep neural network (DNN) architectures. They have also examined EVGO using a few additional methods, and on the validation, test, and training sets, EVGO is proven to be superior in terms of accuracy and loss value. They also looked at EVGO's properties in comparison to other methods, demonstrating the technique's distinctiveness. It's also important to note that EVGO produces findings that are at the cutting edge in all of the studies included in this article. The article's suggested algorithm has remarkable potential for generalization and updating the neural network parameters.

III. PROPOSED ARCHITECTURE

The complete working procedure and the architecture of CNN based leaf disease detection (LDD) model are described in this section. The block diagram and all the basic steps involved from feature extraction to classification and generating output are shown in fig.1

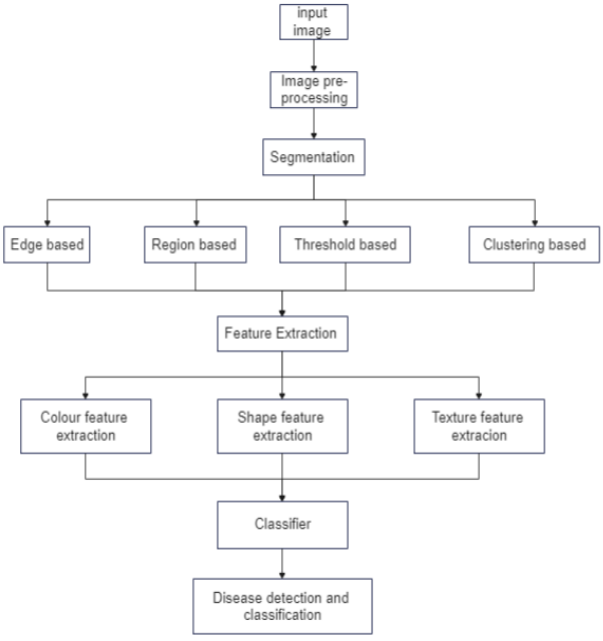

Fig.1 Steps of plants leaf disease detection and Classification

The proposed method consists of a variety of phases, including picture capture, image pre-processing, image segmentation, feature extraction, classification, and performance-based based evaluation of the model. The first phase in the identification and categorization of leaf illness is "Input picture", and the second step is image pre-processing. It will just take the necessary material out of the image and throw away the rest, or it will change the image to fit the dataset. It necessitates certain operations, such as RGB separation, image blurring, image augmentation, etc. A digital picture is segmented into many groups, or "image segments," in order to simplify the structure of the image and make further processing or analysis of an image easier. Depending on discontinuities and similarities, there are two methods for segmenting a picture. Images in discontinuities are split based on unexpected variations in the intensity of data, such as edge detection. In contrast, the similarities category divides the photographs based on certain predetermined standards. Thus, the label edge detection approach is used in picture segmentation. Additionally, it determines the gradient of picture intensities at every pixel in the image.

In feature extraction, all the extracted features are illustrated as an entity, which is further classified into three groups according to texture, color, and shape. Heterogeneous datasets The algorithm proposed in this article requires a combination of features. However, the texture is identified as the best feature for the detection of plant diseases. From the binary segmentation images, two shape features are extracted area and perimeter. generally, color is defined as histograms and moments. The features of colors are extracted from color segmentation images. The color features include meaning, variance, and skewness of grey values of RGB components respectively. Properties like entropy, variance, homogeneity, and contrast, can be put together for texture. All the mentioned steps execute one by one in order to determine whether the uploaded image is of an infected leaf or not. After that, it will apply a classifier model to categorize the diseases in accordance with the relevant data set [1],[9].

IV. TRAINING THE MODEL AND ITS ACCURACY

To train our model we have used a data set available on Kaggle, made by Arun Tejaswi(Reference dataset). This data set nearly contains nine thousand images of potato (healthy, early blight, late blight), Tomato (healthy, early blight, late blight), and paper bell (healthy and bacterial spot). We have divided the entire data set into eight classes for their identification. It is important to mention that the data is labeled data. In the training experiment, images were randomly divided into training data sets (80%), validation data set(10%), and test data set (10%). Both a training and test set were used to train and obtain the best parameters for each segmentation method. The training set was combined with a test set to compare the best configuration of the method using segmentation matrices [4].

We have trained our model in a different number of epochs, after completion of each epoch the validation and test accuracy were noted and for the next epoch, the entire data set is shuffled in the ratio of (80%:10%:10%) along with the image augmentation (to expand theavailable data set). The

accuracy plots for different epochs are shown below, the highest accuracy that we achieved is 98.44 % for fifty epochs.

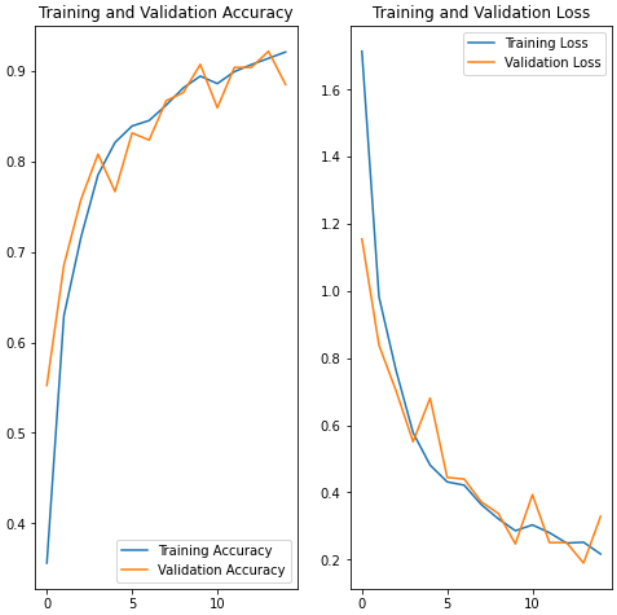

Fig. 2 Training the model in 15 epochs

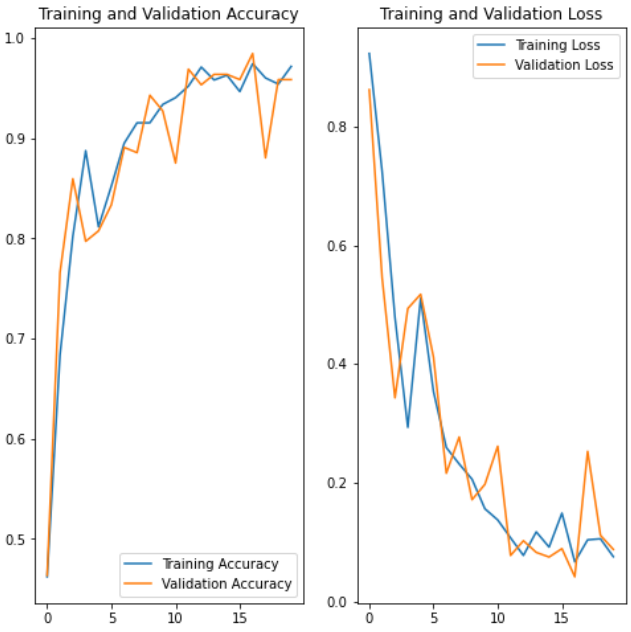

Fig. 3 Training the model in 20 epochs

Fig. 4 Training the model in 30 epochs

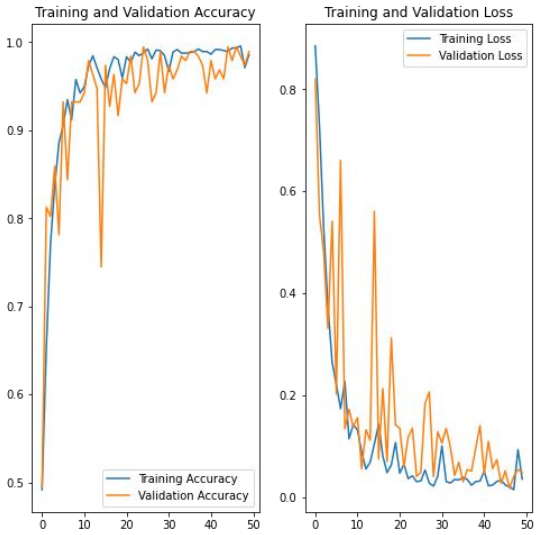

Fig. 5 Training the model in 50 epochs

Quantifying loss :

The loss curves generated while training the neural network are shownin section IV, in general, we can say that the accuracy of the model increases with an increase in the number of epochs (fifteen) i.e. model performance will increase as it will be more and more trained. But in curves, it can be seen clearly that sometimes accuracy is decreasing although the model is more trained.

The reasons behind this fluctuation are as follows :

1. One of the common reasons behind this is the usage of a stochastic gradient descent algorithm for the determination of weights and biases for the training of neural networks.
2. Due to variations in the dataset, we are changing our training, validation, and testing data sets randomly after each epoch.

3. The point where these fluctuations are observed is also not the same, which means every time one performs the testing, they may not get the exact curve as the previous one, as is seen in fig.2 and in fig.6, the curves are not exactly the same.
4. Possibly because the activation function is chosen while creating a neural network.

The loss of our network measures the cost incurred from incorrect predictions. If one wants to build a neural network to determine or to predict something, for any particular given problem and it is giving some answer which may or may not be close to the actual value. So, the reason why did the network get this answer correctly / incorrectly, all depends on how we have trained our neural network and training data set accuracy [6].

Fig. 6 Training the model in 15 epochs

A. NEURAL NETWORK IN PRACTICE: OPTIMIZATION

The backpropagation algorithm is a simple algorithm that just computes the gradients and steps in the opposite direction of yourgradient [11].

However, training these networks in practice is very different. So, optimizing, neural networks inpractice can be extremely difficult because of the fact that loss function can be difficult to optimize.

$$W = W_0 - \eta \, . \, \delta/\delta W \, [J(W)] \qquad (1).$$

Where W is a new weight, W_0 is the previous weight and

η is the learning rate.

And make note that $\delta/\delta W \, [J(W)]$ can be very computationallyintensive to compute [11].

Learning rate in practice determines a lot about how many steps wetake and how much trust we take in our gradients.

Setting the learning rate :

(1) Small learning rate converse slowly and get stuck in falselocal minima.

(2) Large learning rate overshot, becomes unstable, and diverges.

To deal with this kind of problem while training, try lots of different learning rates and choose the one which just works right or do something smarter, design an adaptive learning rate that adapts to the change of parameters.

B. ADAPTIVE LEARNING RATES :

Gradient Descent algorithm : [11]

- Stochastic Gradient Decent Algorithm (SGD)

1. Initialize weight randomly
2. Loop until convergence
3. Compute gradient , $\delta/\delta W[J(W)]$
4. Update weights (Using equation(1))
5. Return the weight

Mini-batch gradient descent is a variant of gradient descent. It divides the training dataset into small batches that are used to calculate model error and update model parameters. Implementations may decide to reduce the gradient's variance by summing the mini-batch over the larger data set. Mini-batch gradient descent aims to strike a balance between batch and stochastic gradient descent. Upside and Downside are two of the approaches used for mini-batch training, as well as single-batch and mixed-batch methods.

Overfitting is the term used to characterize a statistical model when it begins to explain the random error in the data rather than the relationships between variables. Overfitting has the potential to provide inaccurate R-squared values, regression coefficients, and p-values [9].

C. REGULARIZATION :

Regularization is the technique that constrains our optimization problem to discourage complex models. To address the problem of overfitting, we can employ a technique called regularization. Regularization is simply a method for you can introduce in your training to discourage complex models. As we have seen before it's actually critical and crucial for our model to be able to generalize past our training data. So we can fit our model to our loss to almost zero in most cases but that's not what we really care about we always want to train on the training set but have that model be deployed and generalized to a test set to which we don't have access. So the most popular regularization technique for deep learning is a very simple idea of dropout.

Regularization 1: Dropout

(i). During training randomly set some activation to zero By doing so we force it to not rely on those neurons too much so this forces the model to kind of pathways through the network. Or in some other iteration, we may choose some other activation to set to zero.

(ii)This is going to encourage some different pathways and encourage the network to identify different forms of processing its information to accomplish its decision-making capabilities.

Regularization 2: Early stopping

(i) Now the idea here is that as we know the identification of overfitting when our model starts to have very bad performance on our test set but we can kind of create an example test set using our training set so we can split up our training set into two parts one that we'll use for training and one that will not show to training algorithm but we can use to start to identify when we try to overfit a little bit.

(ii) Stop training your model at the point where you get a global minimum validation error.

Fig.7 Early stopping

V. FINAL OBSERVATIONS

The images in fig.7 show the final output generated by the model when a random image sample is given to it as the input image, each output contains three parts namely the actual image, the predicted image, and the accuracy.

Fig.7 Result

VI. CONCLUSION

This paper describes a detailed survey on the prediction of plant leaf disease detection and classification. The literature survey has shown a reliable comparison among several machine learning classifiers for the purpose of plant leaf disease detection and classification. Our results have led us to the conclusion that we can obtain greater accuracy than SVM in multi-class classification problems by using data augmentation techniques, which are used to artificially expand our dataset for improved performance of the machine learning model. The detection and classification of plant leaf disease is said to be a difficult task. The paper also covers several gradient-based strategies for deep neural network design optimization. A significant possibility for updating exists in the algorithm Evolved Gradient Direction Optimizer (EVGO)[11].

REFERENCES

1. Swain, Sripada, Sasmita Kumari Nayak, and Swati Sucharita Barik. "A review on plant leaf diseases detection and classification based on machine learning models." Mukt Shabd 9.6 (2020): 5195-5205.

2. R. Ali, R. Hardie and A. Essa, "A Leaf Recognition Approach to Plant Classification Using Machine Learning," NAECON 2018 - IEEE National Aerospace and Electronics Conference, 2018, pp. 431-434, DOI: 10.1109/NAECON.2018.8556785.

3. Ranjan, Malvika, et al. "Detection and classification of leaf disease using artificial neural network." International Journal of Technical Research and Applications 3.3 (2015): pp. 331-333.

4. Islam, Md Ashiqul, et al. "An automated convolutional neural network based approach for paddy leaf disease detection." International Journal of Advanced Computer Science and Applications 12.1 (2021).

5. S. -J. Lee, T. Chen, L. Yu and C. -H. Lai, "Image Classification Based on the Boost Convolutional Neural Network," in IEEE Access, vol. 6, pp. 12755-12768, 2018, DOI: 10.1109/ACCESS.2018.2796722.

6. Valtchev, S.Z., Wu, J. Domain randomization for neural network classification. J Big Data 8, 94 (2021),springer open, DOI: 10.1186/s40537-021-00455-5

7. Plant Leaf Disease Detection and Automated Medicine Using IoT. Nireeshma R, Rashmi R, Sangeetha M, Spoorthi M E, Shilpa B L.Published in International Research Journal of Engineering and Technology (IRJET),Volume: 07, April 2020.

8. A. S. Tulshan and N. Raul, "Plant Leaf Disease Detection using Machine Learning," 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 2019, PP.1-6, DOI: 10.1109/ACCESS.2019.2914929.

9. P. Jiang, Y. Chen, B. Liu, D. He, and C. Liang, "Real-Time Detection of Apple Leaf Diseases Using Deep Learning Approach Based on Improved Convolutional Neural Networks," in IEEE Access, vol. 7, pp. 59069-59080, 2019,DOIi: 10.1109/ACCESS.2019.2914929.

10. Ramesh Maniyath, Shima & V, Vinod & M, Niveditha & R, Pooja & N, Prasad & N, Shashank & Ram, Hebbar. (2018). Plant Disease Detection Using Machine Learning. 41-45. DOI: 10.1109/ICDI3C.2018.00017

11. I. Karabayir, O. Akbilgic, and N. Tas, "A Novel Learning Algorithm to Optimize Deep Neural Networks: Evolved Gradient Direction Optimizer (EVGO)," in IEEE Transactions on Neural Networks and Learning Systems, vol. 32, no. 2, pp. 685-694, Feb. 2021, DOI: 10.1109/TNNLS.2020.2979121.