Grenze International Journal of Engineering and Technology, Jan Issue

Boosting the Accuracy of Deep Learning Models for Detecting and Classifying Multiple Diseases in Plant Leaves using a Voting Classifier

Jesal Desai[1] and Dr. Amit Ganatra[2]
[1-2]Computer Science & Engineering, Devang Patel Institute of Advance Technology and Research, Charotar University of Science and Engineering (CHARUSAT), Changa, India
Email: jesal.vasavada@gmail.com, amitganatra.ce@charusat.ac.in

***Abstract*—Indian economy largely depends upon agriculture. Emergence of Deep Learning has given solutions to many issues related to farming. One of the agricultural issues is plant disease detection and classification. Much research has been done and is still going on in this field as it directly relates to productivity of crops and plants. Loss in productivity of crops impacts farmer's income and thus country's economy. It has now become possible to avoid huge losses as many models have been implemented lately and various performance metrics have been used to evaluate them and compare their outcomes. In this study, we look into and analyze Deep Learning methods, that have been developed over the period of last few years. This article concentrates on a two-step approach. The first step is to find and classify nine types of tomato leaf diseases using six Deep Learning architectures: VGG16, VGG19, InceptionNet, ResNet50 V2, DenseNet 121 and MobileNet and analyze their performances based on precision, recall, F1-Score, testing and validation accuracy. The second phase is an ensemble technique known as Voting Classifier (VC), which trains an ensemble of many models and predicts an output (class) based on the highest probability of the chosen class being the outcome. Output of Deep Learning Models are compared with the proposed approach. The experimental analysis signifies that the proposed approach is effective and acheives better accur.**

***Index Terms*— Artificial Intelligence, Deep Convolutional Neural Networks, Plant Disease, Challenges, Future Vision, Open Datasets.**

I. INTRODUCTION

In the farm outputs, India holds second place worldwide. The study of 2018 shows that around 17–18% of country's GDP depends on Agriculture. (Financial Express, 2018) [1]. As stated by Agriculture Secretary Sanjay Aggarwal "AI will be the game-changer in the agriculture". Artificial Intelligence is the future of farming. Researches are going on and researchers are trying their best to prepare accurate models that can work on any undesirable conditions and protect the yields. Farmers need solutions for realistic challenges they face. Farming is still at a nascent stage. A lot of research is required to be done to solve the challenges of Indian Agriculture. Detecting and classifying plant diseases accurately and efficiently is one such difficult task. One major reason that the quality of fruits, vegetables and crops is badly affected is plant diseases which negatively impacts the economy of the nation. It is estimated that population of whole world will reach 9.1 Billion people at the halfway Mark of the 21st century, thus in order to maintain a steady supply, food production would be required to be

Grenze ID: 01.GIJET.9.1.507_1

ramped up by 70% as per a statement by the Food and Agricultural Organization (FAO) (The Times of India, 2019) [2]. Two critical factors affecting plants are diseases and disorders. The diseases are caused by various bacteria, fungi etc. which comprise of the living components of the ecosystems. While, the disorders are caused by temperature, rainfall, soil quality etc. which comprise of the abiotic components of the ecosystems. Research indicates that change in climate conditions hugely impacts the pattern of development of pathogens in plants and even new diseases can occur which have never occurred before for which no local expertise is available. Misjudgment of diseases leads to inappropriate application of pesticides in plants which affects the health of consumers. Therefore, accurate and efficient detection of plant disease in timely manner is crucial. Disease detection and classification has become the major hotspot in agricultural research.Hence, we need Artificial Intelligence (AI) Powered solution [3]. AI is the branch of Computer Science that focuses on developing intelligent systems.

Due to the advent of advance digital cameras, disease detection in plants with the aid of ML and DL models has seized the attention of the research community. ML approaches are suitable for identification of plant diseases from laboratory images with uniform background. A lot of research on plant disease detection has been done using ML techniques [4], [5], [6]. But ML approaches have limitations as these approaches are not able to perform automatic feature extraction and thus rely on some other technologies for the same and therefore sometimes unable to achieve higher classification accuracies due to improper feature extraction. Convolutional Neural Networks (CNN) overcome this disadvantage of Machine Learning as input image features can be extracted easily through computations like simple convolution operations on image pixel intensities rather than relying on other methods.

In recent few years use of Ensemble Learning has benefitted a lot in improving the performance of deep learning models. Its main benefit is that it improves the average prediction performance of all ensemble members. Increased performance using ensembles is frequently due to a reduction in the variance component of prediction mistakes made by contributing models.

A. Motivation

A lot of work has already been done using transfer learning as it has made the task of object detection very easy using pre- trained networks. But sometimes to relying on single model may not give the best results. To improve the productivity of crops, a model is required which gives the best results for detection of diseases. Hence Ensemble Learning serves the purpose There are two key reasons to utilize an ensemble rather than a single model, both of which are related: (1) Prediction and performance: An ensemble can forecast and perform better than any single contributing model. (2) Robustness: The spread or dispersion of predictions and model performance is reduced by using an ensemble. As a result, in this work, a voting classifier is used to six state-of-the-art models in order to reduce error and obtain maximum accuracy for tomato leaf disease classification.

B. Problem Domain

Now a days, it has been felt by the researchers that to depend on the single machine learning or deep learning model and conclude the results based on the performance of single model is not a wise option. Combining the advantages of two or more models can give better predictions as one model can overcome the disadvantage of another model. Ensemble Learning is a broad area which includes methods like Bagging, Stacking and Boosting. Which method to be used is totally dependent on the requirement, dataset and models to be combined and the expected outcome.

C. Problem Definition

In this study, the goal is to achieve best possible accuracy and minimize the loss for multi class classification of tomato leaf diseases by implementing the voting classifier method is generally used for classification problems.

D. Statement

The aim of this study is to boost the performance of deep learning models for multi class detection and classification of Tomato plant disease using voting classifier. Voting classifier predicts the output class based on the highest majority of voting of each output class.

E. Voting Classifier

We used a Voting Classifier in this work, which is a machine learning model that learns from an ensemble of models and predicts an output (class) based on the likelihood of the chosen class being the outcome. The Voting Classifier forecasts the output class based on the most votes using the results of each classifier. To put it another way, this technique makes predictions for each data point using a variety of models. Each model's predictions are regarded as a 'vote.' The final prediction is based on the predictions obtained from the majority of the models. Figure 1 depicts a high-level overview of the voting classifier's operation.

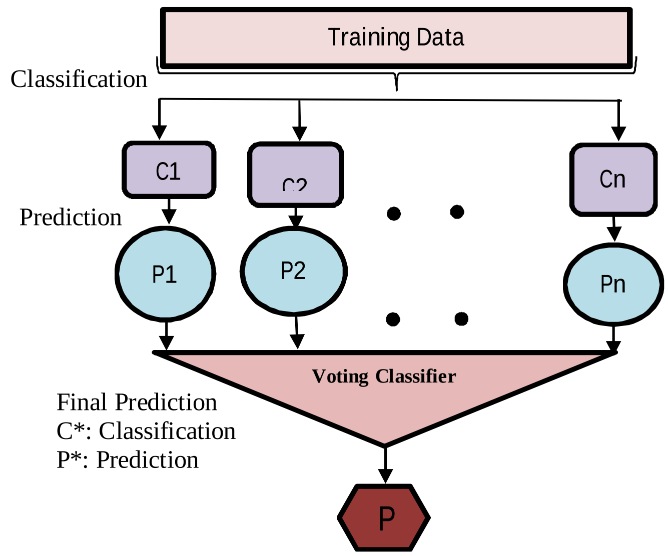

Figure 1: working of Voting Classifier

The remaining sections of paper are structured in the following way: Sections 2 briefly mentions the research carried out by researchers using existing and new/modified/improved DL architectures for detecting diseases in plants. Section 3 compares six state of the art deep learning architectures and voting classifier for boosting the performance of the detection of nine tomato leaf diseases. Finally, the last section discusses the outcomes and observations of the research following which is the conclusion.

This section in the paper examines the related research accomplished for identification and classification of plant disorders and illnesses by utilizing distinctive Deep CNN architectures and Ensemble Learning Methods. Table 1 summarizes recent related work of researchers and gives idea about different approaches they have used.

TABLE I: SUMMARY OF FEW RECENT RELATED WORK

Ref(s)	Approach Used	Dataset	No of Images	Plant	Performance Parameters
Honglei et al., 2021 [8]	Machine Learning Models + Stacking Ensemble	Baidu image search engine	10,000	Apple	97.34 % (Validation Accuracy)
Haixia et al., 2020 [9]	Deep Learning Models + Stack Ensemble	Collected manually from South China Agricultural University	6029	Peanut	97.59% (Validation Accuracy)
Ruoling Deng1, et al., 2021 [10]	Deep Learning Models + Average Ensemble Model	Captured from the farm field	33026	Rice	91 % (Accuracy)
Musa Peker 2021 [11]	Voting Ensemble	Tomato Plant disease	14785	Tomato Leaves	98.15 %
Muammer Turkoglu et al., 2021[12]	Voting Ensemble	Turkey-PlantDataset	4,447	Turkey-PlantDataset	97.56 %

II. MATERIALS AND METHODS

A. Dataset

For this study, the publicly available ‘New Plant Diseases Dataset also known as the ‘Tomato Dataset’ was used. The Tomato dataset was made available on Kaggle.com by Nouaman Lamrahi in 2019. The dataset contains images of tomato leaves from 10 classes out of which 9 classes have images with diseased leaves and 1 class has images with healthy leaves. For this study the images were augmented and the dataset was divided into training and validation sets which contained 18345 and 4585 images respectively. Following Table 2 shows the original number of images belonging to each class. It also showcases that the classes are balanced. The images were rescaled so that each image had dimensions on 224 X 224. To augment the dataset, the images were randomly zoomed by a margin of up to 20% of the original and horizontally flipped in some cases. The validation data was used to evaluate the 6 models used in this study which were trained on the training data.

TABLE II. NO. OF TRAINING AND VALIDATION IMAGES FOR EACH DISEASE TYPE

Class	No. of Training Images	No. of Validation Images
Tomato_Bacterial_Spot (DS1)	1702	425
Tomato_Early_Blight (DS2)	1920	480
Tomato_Late_Blight (DS3)	1851	463
Tomato_Leaf_Mold (DS4)	1882	470
Tomato_Septoria_Leaf_Spot (DS5)	1745	436
Tomato_Spider_Mites	1741	435
Tomato_Target_Spot (DS7)	1827	457
Tomato_Yellow_Leaf_Curl_Virus (DS8)	1961	490
Tomato_Mosaic_Virus(DS9)	1790	448
Healthy	1926	481

B. Experimental Setup and Training

Google Collaboratory was used to carry out the experiments in Python programming language using the TensorFlow and Keras libraries. NVIDIA GPU with CUDA Version 11.2 was used for the experiments. Table 3 shows the specifications used in the study.

TABLE III. LIST OF HARDWARE AND SOFTWARE SPECIFICATIONS USED IN THE STUDY

Hardware and Software	Specifications
Memory (RAM)	13 GB
GPU	NVIDIA GPU with CUDA Version 11.2
GPU Memory	11.4 GB
Operating System	Windows 10 Pro version 20H2
Deep Learning Library	TensorFlow and Keras
Programming Language	Python

The 6 models viz VGG16, VGG19, InceptionNet, ResNet, DenseNet and MobileNet are trained and then the results of models are then fed into Voting Classifier. Figure below shows work flow diagram used in this study. Voting Classifier and predicts the output class based on the highest majority of voting.

Algorithm for Voting Classifier

```
def predict_binary(predictions_proba):
  predictions = predictions_proba.copy()
  for i in range(len(predictions_proba)):
      if predictions[i][0] > 0.5:
          predictions[i][0] = 1
      else:
          predictions[i][0] = 0
  predictions = np.argmax(predictions, axis=1)
  return predictions
def predict_classes(model_input):
  result = pd.DataFrame()
  for i in range(len(model_input)):
    model = load_model[i])
    test_set_classes = test_set.classes
    predictions_proba = model.predict(test_set)
```

```
    predictions = predict_binary(predictions_proba)

    result[list(model_input.keys())[i]] = predictions
  return(result)
df = predict_classes(model_input)
def voting_classifier(df):
  df['Votes'] = np.nan
  for i in range(df.shape[0]):
    df['Votes'][i] = int(df.iloc[i].mode()[0])
  df['Votes'] = df['Votes'].astype('int64')
voting_classifier(df)
```

C. Evaluation Parameters

Following Parameters are used to evaluate the performance of models. TP denotes True Positive , FP is for False positive, TN is for True negative and FN is for False Negative.

Sensitivity/ Recall (R): This measure indicates what proportion of plants that are actually infectious are classified by the algorithm as infectious / diseased. The formula to calculate Recall is given as equation 1:

$$TP/(TP+FN) \ldots\ldots\ldots (1)$$

Specificity: This measure indicates what proportion of plants/leaves that did NOT have disease, were predicted by the model as healthy. Equation 2 shows the formula to calculate Specificity as:

$$TN/(TN+FP) \ldots\ldots\ldots (2)$$

Accuracy: Accuracy is the number of correct predictions by the model over all kind of predictions. Accuracy is calculated using equation 3

$$(TP+TN)/(TP+TN+FN+FP) \ldots\ldots (3)$$

Precision (P): This measure indicates what proportion of plants/ leaves that we diagnosed as having disease, actually had disease. Prediction is calculated using equation 4

$$TP/(TP+FP) \ldots\ldots\ldots\ldots (4)$$

F1 Score: The formula to calculate F1 Score is as equation 5. It is harmonic mean of precision and recall.

$$2*(P*R)/(P+R) \ldots (5)$$

III. RESULTS AND DISCUSSION

A. Comparison of Deep Learning models

From the Table 4 and Figure 3, it can be seen that the maximum training time of 50 minutes and 3 seconds and minimum accuracy of 83.3. % is taken by Inceptionv3 model while MobileNet took minimum training 30 minutes and 8 seconds time and maximum accuracy of 94.29 % in comparison to all other models which shows that MobileNet is best fit for this Tomato Dataset. The number of layers and training and non-training parameters were less in VGG16 but from the results it can be analyzed that accuracy and efficiency does not depend only on those factors. Depthwise Separable nature of MobileNet has played significant role and outperformed all other models.

Results of MobileNet, DenseNet121 nearly similar in terms of accuracy but DenseNet due to its dense nature and maximum number of layers took more computation time as compared to light model of MobileNet. Surprisingly VGG-16 performed slightly better than VGG-19, perhaps due to the lesser number of layers reducing the problem of vanishing gradients. Figure also gives us idea about model overfitting of VGG-19 as the gap between the training and validation accuracies is quite large. This is also the case for VGG-16 and ResNet50 models but the margin is still acceptable.

TABLE IV: COMPARISON OF ACCURACY AND CPU TIME BETWEEN MODELS

Machine Learning Model	Time Taken (CPU time)	Training Accuracy	Validation Accuracy	(Correct, Misclassified)
VGG16	39min 41s	93.94%	89.79%	(3870,715)
VGG19	41min 56s	92.03%	86.56%	(3969,616)
MobileNet	36min 8s	96.56%	94.29%	(4323,262)
DenseNet121	45min 3s	95.15%	93.39%	(4281,304)
ResNet50	41min 28s	94.75%	90.36%	(4143,442)
InceptionNetV3	50min 13s	89.01%	83.38%	(3823,762)

B. Results after applying Voting Classifier

The results of above six models are fed to Voting Classifier as represented in Fig. Voting Classifier which works as a meta learner and predicts the output class based on the highest majority of voting. From the figure below, it can be clearly observed that how voting classifier helps in boosting the accuracy of plant disease detection. After applying Voting Classifier, the accuracy is 96.4 % which is 2 % more as compared to MobileNet.

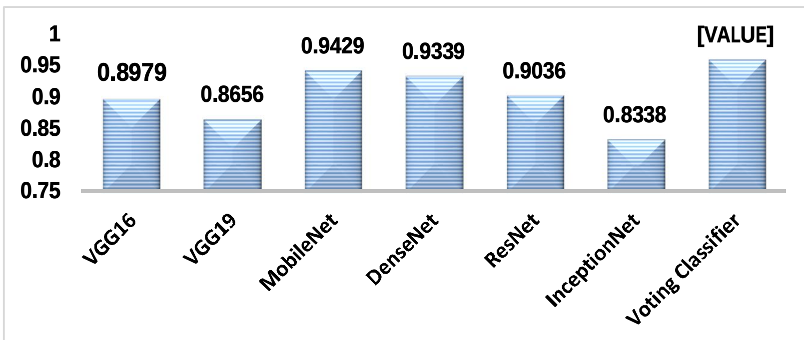

Figure 2:Comparison of accuracies of models after applying voting classifier

Performance Evaluation in terms of F1-Score

Figure 4. shows that disease Tomato_Yellow_Leaf_Curl_Virus (DS8) is easy to detect by all models as F1 score of all models for this disease is more than 0,9 in all cases. Similar is the case for DS1 disease for which F1 score of all models is more than 0.9 except for InceptionNet. It is noticed that not all models are able to classify DS2 And DS7 diseases easily. Diseases D2 (Tomato Early Blight) is easily detected by MobileNet, DenseNet and Voting Classifier. Similarly, D7 (Tomato Target Spot) DenseNet and Voting classifier gave best performance. This leads us to the conclusion that some diseases are harder to recognize (or easier to misclassify) than others and might want inclusion of additional image data to further improve Deep Learning model performance. Also, from the above graph, it can be concluded that Voting Classifier overcome the weakness of all models, easily able to detect all diseases as F1 score for all diseases is more than 0.9 in case of Voting Classifier (VC).

IV. CONCLUSION AND FUTURE WORK

Convolutional Neural Networks have made remarkable success in the field of computer vision related tasks. The paper reviews advancement in CNN architectures starting from LeNet, especially for plant disease detection and classification. In the paper, we have noted our findings after the review of Deep Learning methods and systems for detection and identification of plant diseases which we have applied for Tomato leaves. MobileNet achieved the best results: 96.5 % validation accuracy and 94.2 % testing accuracy only in 38 minutes and 8 seconds CPU time which is very less as compared to other models. The lowest accuracy can be seen in InceptionNetV3, it also takes maximum CPU time. The results show that performance of architectures is not up to the mark for all kind of diseases, hence Voting Classifier is used to minimize the loss. Results of Voting Classifier clearly shows that the accuracy is boosted as it is not dependent on one model but takes benefit of all models. The accuracy is increased to 96 % after using Voting Classifier.

The main objective of future work is to achieve higher accuracy to recognize and classify diseases using Deep Learning methods. The aim is to develop effective system that generalizes to different datasets. The future work also involves building smart mobile application which helps farmers in speedy recognition of diseases and thus aids in fast decision making when it comes to use of pesticides as right time to save plant from diseases.

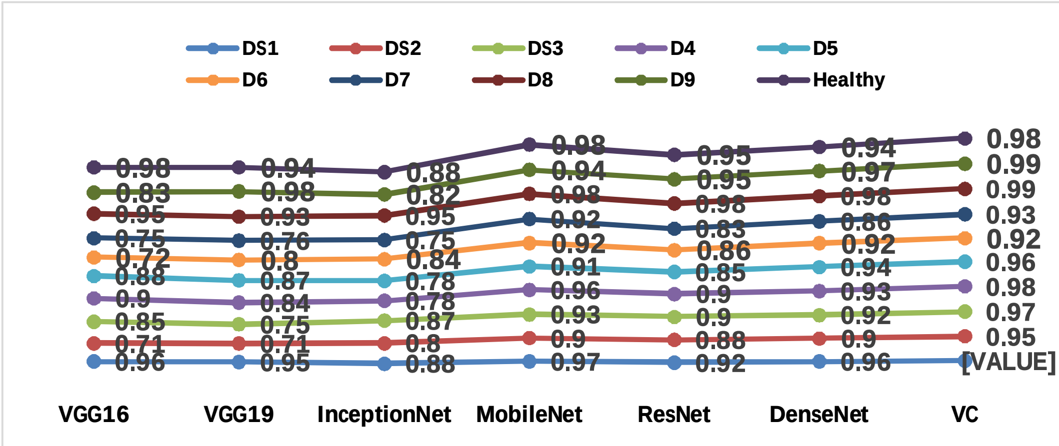

Figure 3:F1-Scores of all models for each Tomato Leaf Disease classes

REFERENCES

[1] India economic survey 2018: Farmers gain as agriculture mechanization speeds up, but more R&D needed, Financial Express, Jan 29, 2018

[2] AI, big data to be game changer in India's farm sector: Agri Sec, Times of India, Sep 26, 2019

[3] Sudduth KA, Woodward-Greene MJ, Penning BW, Locke MA, Rivers AR, Veum KS. AI down on the farm. IT Professional. 2020 May 21;22(3):22-6.

[4] Dharmaraj V, Vijayanand C. Artificial Intelligence (AI) in Agriculture. Int. J. Curr. Microbiol. App. Sci. 2018 Dec; 7(12):2122-8.

[5] Ramesh S, Hebbar R, Niveditha M, Pooja R, Shashank N, Vinod PV. Plant disease detection using machine learning. In2018 International conference on design innovations for 3Cs compute communicate control (ICDI3C) 2018 Apr 25 (pp. 41-45). IEEE.

[6] Wahab AH, Zahari R, Lim TH. Detecting diseases in chilli plants using K-means segmented support vector machine. In2019 3rd International Conference on Imaging, Signal Processing and Communication (ICISPC) 2019 Jul 27 (pp. 57-61). IEEE.

[7] Kumari S, Kumar D, Mittal M. An ensemble approach for classification and prediction of diabetes mellitus using soft voting classifier. International Journal of Cognitive Computing in Engineering. 2021 Jun 1;2:40-6.

[8] Li H, Jin Y, Zhong J, Zhao R. A Fruit Tree Disease Diagnosis Model Based on Stacking Ensemble Learning. Complexity. 2021 Sep 15;2021.

[9] Qi H, Liang Y, Ding Q, Zou J. Automatic identification of peanut-leaf diseases based on stack ensemble. Applied Sciences. 2021 Jan;11(4):1950.

[10] Deng R, Tao M, Xing H, Yang X, Liu C, Liao K, Qi L. Automatic diagnosis of rice diseases using deep learning. Frontiers in Plant Science. 2021 Aug 19:1691.

[11] Peker M. Multi-channel capsule network ensemble for plant disease detection. SN Applied Sciences. 2021 Jul;3(7):1-0.

[12] Turkoglu M, Yanikoğlu B, Hanbay D. PlantDiseaseNet: Convolutional neural network ensemble for plant disease and pest detection. Signal, Image and Video Processing. 2022 Mar;16(2):301-9.

[13] Mascarenhas S, Agarwal M. A comparison between VGG16, VGG19 and ResNet50 architecture frameworks for Image Classification. In2021 International Conference on Disruptive Technologies for Multi-Disciplinary Research and Applications (CENTCON) 2021 Nov 19 (Vol. 1, pp. 96-99). IEEE.

[14] Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A. Going deeper with convolutions. InProceedings of the IEEE conference on computer vision and pattern recognition 2015 (pp. 1-9).

[15] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. InProceedings of the IEEE conference on computer vision and pattern recognition 2016 (pp. 770-778).

[16] Huang G, Liu Z, Van Der Maaten L, Weinberger KQ. Densely connected convolutional networks. InProceedings of the IEEE conference on computer vision and pattern recognition 2017 (pp. 4700-4708).

[17] Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861. 2017 Apr 17.