Detecting The Category of Foliar Diseases in Apple Trees

Plant Pathology 2020 - FGVC7

1st Aakanksha Saini
Roll Number : MT19056
Indraprastha Institute of Information Technology (IIIT-D)
New Delhi, India
aakanksha19056@iiitd.ac.in

2nd Kaushal Sanadhya
Roll Number : MT19133
Indraprastha Institute of Information Technology (IIIT-D)
New Delhi, India
kaushal19133@iiitd.ac.in

***Abstract*—Plant leaf diseases and harmful insects are significant challenges in front of the peasants. Efficient and accurate prediction of leaf diseases in crops could help farmers in early treatment. Artificial intelligence has emerged as a revolutionary alternative for detecting these diseases in early stages. This project aims to train models which can accurately distinguish between healthy and infected leaf with specific disease category. Various Machine learning / deep learning techniques Like NuSVM, deep convolutional neural networks such as Res Net, Dense Net, etc. have been used for this purpose.**

I. Introduction

Agriculture plays a vital role in the continuous existence of human beings on earth due to people's dependence on food production.Moreover, agriculture contributes to a significant part of economic growth. Misdiagnosis of many plant diseases can lead to the emergence of resistant pathogen strains, increased input costs, and more outbreaks with substantial financial loss and environmental impacts. Currently these diseases are detected by agriculture experts, which is a highly time-consuming process. The use of Artificial intelligence techniques like deep learning is an innovative solution for such agriculture applications.

The dataset [1] of high resolution apple leaf images is taken up from Kaggle competition named Plant Pathology 2020 - FGVC7 [2].The objective of this challenge is to train a model using the given training images of apple leaves and precisely classify the test images into different disease categories or a healthy leaf.

II. LITERATURE REVIEW

Over two decades, food security has become a major problem for the world. For developing countries like India, agriculture is facing many issues, including crop loss due to deficiency of nutrients in soil, bacterial and viral infection on various parts of the plant etc. Possibly one can come up with an idea of educating farmers regarding organic farming, but the issues like disease identification is not an easy task. Moreover, some expertise is required to detect several diseases. For larger farms, this task becomes time-consuming as well.

It is necessary to review the work done by the experts in this field before we carry out our work. It would help us to evaluate various models and provide some valuable ideas for our work. A brief introduction to some of the eye-catching work is described below:

One study [3] presented disease detection in Malus Domestica through unsupervised machine learning techniques like K means clustering and color analysis to classify and recognize the texture and color features of healthy and diseased plant leaves.

In one paper, the author used an artificial neural network (ANN) and Gabor filter for feature extraction and image preprocessing. This model [14] achieved up to 91

Another author presented an in-field automatic wheat disease diagnosis model [13] based on multiple deep learning instances.VGG-FCN-VD16 and VGG-FCN-S were the architecture used to achieve the mean recognition accuracies of 97.5

These studies inspired us to take up this Kaggle challenge [2]. Moreover, the images are not pre-processed hence provided us an opportunity to work for the real-world scenario.

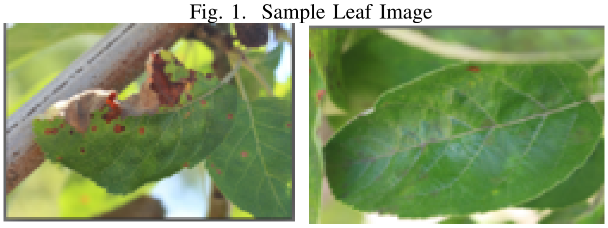
III. DATA SET DESCRIPTION AND PRE-PROCESSING

The data set [1] consists of 3642 images of apple leaves from the support of the Cornell Initiative for Digital Agriculture (CIDA). Training and test set contains 1821 images each. The training images were manually annotated using Train.csv file. All the photos are of high resolution in RGB format with size 1365 X 2048 pixels. The prominent feature of this data set is that the leaf images are not pre-processed to provide uniform light intensity and orientation of the leaf in the picture etc. unlike other famous data sets like Plant Village data set.

A. CLASS DISTRIBUTION

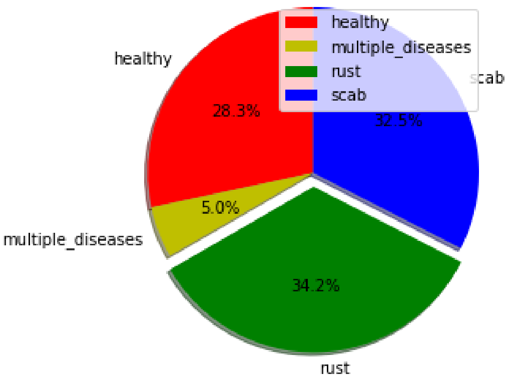
The training data set is divided into four classes, namely, healthy, rust, scab, and multiple diseased. This data is highly imbalanced, as shown in the below pie chart.

Fig. 1. Sample Leaf Image



Therefore, there is a high demand for oversampling techniques like mixup (provided by Fast.AI) and image augmentation. Since the images were captured with different light intensity conditions and various angles, hence we performed the image normalization before feeding them into CNN models for training.

Fig. 2. Training Data Distribution

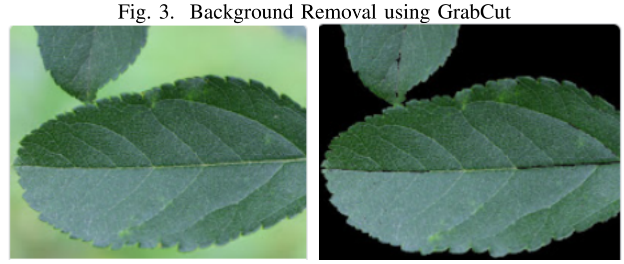


B. DATA PRE-PROCESSING

As shown in fig 1, these images were captured with non-uniform background, different light intensity, multiple angles. We also encountered the problem of data imbalance, as shown in fig2. Therefore, we performed various pre-processing steps to make these pictures suitable for training machine learning / deep learning models.

- We used the GrabCut algorithm [15] to extract the foreground of an image with minimal user interaction. Initially, we draw a rectangle around the leaf region. Then this algorithm segments it iteratively to get the best result. For some images with unsatisfactory results, we had to give some strokes on the picture as fine touch-ups. Example of the leaf image with and without background is shown in fig 3.
- The class imbalance and small data set problem was dealt with by using two techniques. First of all, we created the transformation of the images by flipping , introducing lighting effects , zoom , rotate and paffine etc.
- Mixup is one of the callbacks provided by Fast.AI for performing data augmentation. It is handy for regularizing models in computer vision. The image augmentation is performed by mixing up the linear combination of two images (not necessarily from the same class). Though one side effect was that the final training and validation loss was higher than training without it.
- All the images were normalized with the normalized transformation of Fast.AI library with imagenet stats. The reason for choosing imagenet stats rather than cifar stats or mnist stats is that all the CNN models of Fast AI are pre trained on ImageNet data set. We had chosen the pre trained model since the size of our dataset is comparatively small and the pre trained models are trained on 14 million images.

Fig. 3. Background Removal using GrabCut



IV. PROPOSED ARCHITECTURE

We used various feature extraction techniques to extract the features of the images after performing the pre processing on the image data set.These features were used to perform classification using state-of-the-art support vector machine model(SVM). This resulted in a terrible score on the public leaderboard of Kaggle since this proposed solution was highly dependent on the properties of features extracted. Therefore we moved on to deep convolutional neural networks due to their auto feature extraction properties. The features are learned while CNN is being trained on the images through hundreds of its hidden layers.

We have used Fast AI to implement CNN models due to its efficient deep learning models which can be used to solve practical problems quickly and reliably.We have used the below models:

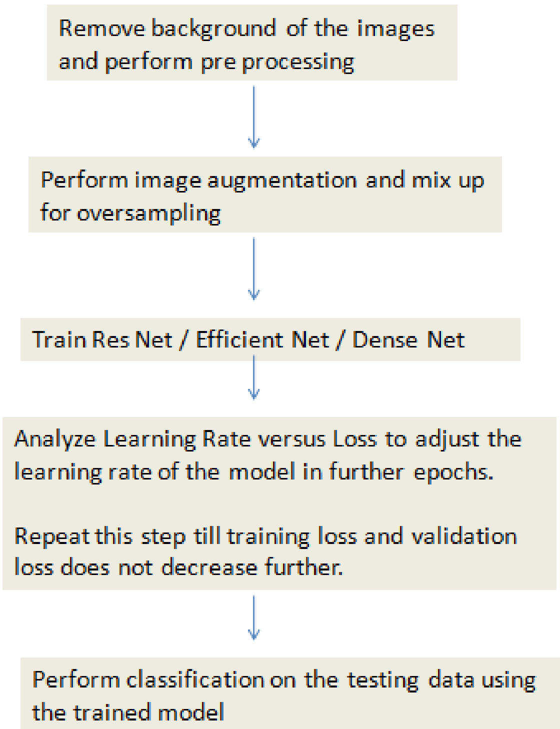
- Efficient Net (B0 and B5)
- Res Net (Res Net 50 , Res Net 152)
- Dense Net (Dense Net 161, Dense Net 121 ,Dense Net 169)

The reason behind choosing these three CNNs is that these neural networks perform exceptionally well on the image net dataset challenge [16]. These Neural Network are readily available in Pytorch/Tensorflow. The pre-trained versions of these models on the Imagenet data set are also part of FastAI libraries, which are giving high accuracy for our data set.

The flowchart of the proposed architecture is shown in figure 4.

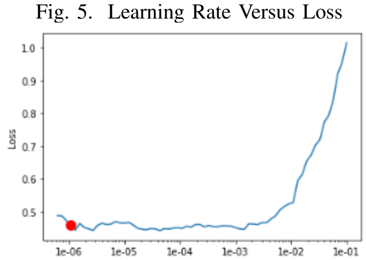
- Similar pre processing steps were followed for both training and test images.
- After pre processing , the training images were resized and divided into batches using ImageDataBunch provided by Fast.AI.The reason behind dividing the training images being the RAM constraints of the machine.

Fig. 4. Proposed Architecture



- The training and validation set were generated implicitly in the ratio of 80:20 using ImageDataBunch.
- The pre trained models were used for transfer learning. The final weights of the model were derived from the training images.
- The model was trained using ”fit one cycle” function for five epochs initially to analyze the training and validation losses.
- Then the model is unfreezed to fetch the learning rate versus loss curve. This step is performed to estimate the optimal learning rate slice for further training.The optimal learning rate is shown by red dot in figure 5.

Fig. 5. Learning Rate Versus Loss



- The model is trained further until the validation and training loss are not decreased further.
- Finally trained model is used to perform the classification of the test images.If the images’ ensemble is performed, then the average of the prediction is taken from various models.

V. RESULTS

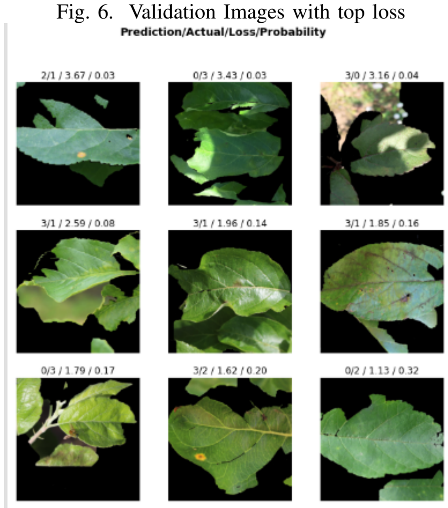
We have trained more than 20 models for the Kaggle challenge.Top few results are summarized in the below table.

Summary of all the models used			
Model Name	Model Description	Kaggle Public Leaderboard Score	Accuracy
Ensemble (DenseNet 169 + Efficient Net B5)	Pre Trained = True Image Size = 256 Transformation =True Mixup = True	95.6	98.6
Ensemble(Dense Net 169 + Efficient Net B5)	Pretrained = True Image Size = 256 Augmentation = True Mixup = True	95.3	98.6
Efficient Net B5	PreTrained = True Image Size = 256 Augmentation = False Mixup = True	95.3	98.6
Ensemble(Dense Net 121 + Res Net 50)	Pre Trained = True Image Size = 400 Mixup = False Augmentation = True	94.4	97.6
Dense Net 121	Pre Trained = True Image Size = 400 Mixup = False Augmentation =True	93.3	96.4
Res Net 50	Pre Trained = True Image size = 512 Mixup = false Augmentation = True	93	95.87

VI. ANALYSIS OF RESULTS

- We can conclude from figure 6 that the leaves with multiple diseases are not easy to identify. They are miss classified most often.
- Although the images are highly normalized, the classifier yet often misunderstands the sunlight spots as disease spots.

Fig. 6. Validation Images with top loss



- The ensemble of models can rectify this problem up to a certain extent. We have used ensemble because the individual learners were proved to be weak learners. They were not able to differentiate sunlight spots and diseased spots.
- Some of the ensembles listed above were performing equally well as compared to the individual models on the Kaggle leaderboard.But the prediction confidence of ensembles was quite high.

VII. INFERENCES DRAWN

- We have used pre-trained CNN models which are trained on 14 Million images of image net data set belonging to 1000 different classes. There are several substantial benefits to leverage pre-trained models like transfer learning, dependable prediction accuracy, and excellent feature extraction for our data set.
- Decreasing the size of an image is undoubtedly degrading the performance of CNN up to a certain level due to the lesser availability of finer features in leaf images.
- But decreasing the sizes of images up to a certain level and dividing the training images into batches were necessary as well since we have limited RAM and we received CUDA memory errors while working on the original size of the images. While reducing the size of the image we reduced the dimensions proportionally so that the resulting image has square shape and the leaf is at the central part of it.
- The accuracy could still be hampered by reducing the size as it might have cropped out some part of the leaf in focus. To overcome this, the model was trained using differential sizes which lead improvement in scores.
- The models trained with the mixup functionality of FASTAI for image augmentation are giving higher accuracy than without it. Although the final loss will be higher than when training without it.
- Moreover, the models trained with mixup functionality are making predictions with less confidence.
- Ensembles of these models are performing better on the public leaderboard. These ensembles are designed by taking the average of the outputs by various CNN models which perform better because of independence of each classifier and diversification of features.
- Discriminative fine-tuning or Differential learning rates are giving optimal performance when used with models. The possible reason could be that the first few layers would deal with very granular details of the leaf image such as the edges etc — of which we usually wouldn't want to change much and like to retain it's information. As such, there's not much need to change their weights by a big amount.On the other hand, the later layers are responsible to get detailed features of the leaf such as color pattern and shape of disease spot.

VIII. Individual Contribution

Both the team members collaborated equally in all the tasks since the project is implementation based.

- Kaushal Sanadhya: Pre-processing of Test Images, Training variations of all 3 models, application of ensemble methods, literature reading, documentation
- Aakanksha Saini: Pre-processing of Train Images, Training variations of all three models, application of ensemble methods, literature reading, documentation

References

[1] Project Dataset (https://arxiv.org/abs/2004.11958)

[2] https://www.kaggle.com/c/plant-pathology-2020-fgvc7/overview

[3] M. Akila, P. Deepan, 2018, Detection and Classification of Plant Leaf Diseases by using Deep Learning Algorithm, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH TECHNOLOGY (IJERT) ICONNECT – 2018 (Volume 6 – Issue 07)

[4] Plant Disease Classification Using Deep Learning Methods ,ICMLSC 2020: Proceedings of the 4th International Conference on Machine Learning and Soft ComputingJanuary 2020 Pages 5–9https://doi.org/10.1145/3380688.3380697

[5] M. Sardogan, A. Tuncer and Y. Ozen, "Plant Leaf Disease Detection and Classification Based on CNN with LVQ Algorithm," 2018 3rd International Conference on Computer Science and Engineering (UBMK), Sarajevo, 2018, pp. 382-385.

[6] A. S. Tulshan and N. Raul, "Plant Leaf Disease Detection using Machine Learning," 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kanpur, India, 2019, pp. 1-6.

[7] Jeon, Wang-Su Rhee, Sang-Yong. (2017). Plant Leaf Recognition Using a Convolution Neural Network. The International Journal of Fuzzy Logic and Intelligent Systems. 17. 26-34. 10.5391/IJFIS.2017.17.1.26.

[8] Azlah, M.A.F.; Chua, L.S.; Rahmad, F.R.; Abdullah, F.I.; Wan Alwi, S.R. Review on Techniques for Plant Leaf Classification and Recognition. Computers 2019, 8, 77.

[9] V. Pooja, R. Das and V. Kanchana, "Identification of plant leaf diseases using image processing techniques," 2017 IEEE Technological Innovations in ICT for Agriculture and Rural Development (TIAR), Chennai, 2017, pp. 130-133.

[10] Manu BN (2020). Plant Leaf Disease Detection and Classification using Multiclass SVM Classifier (https://www.mathworks.com/matlabcentral/fileexchange/55098-plant-leaf-disease-detection-and-classification-using-multiclass-svm-classifier), MATLAB Central File Exchange. Retrieved April 21, 2020.

[11] M. Chanda and M. Biswas, "Plant disease identification and classification using Back-Propagation Neural Network with Particle Swarm Optimization," 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 2019, pp. 1029-1036.

[12] Jakjoud, Fatimazahra Anas, Hatim Bouaaddi, Abella. (2019). Deep Learning application for plant diseases detection. 1-6. 10.1145/3372938.3372983.
[13] Jiang Lu, Jie Hu, Guannan Zhao, Fenghua Mei, Changshui Zhang, An in-field automatic wheat disease diagnosis system, Computers and Electronics in Agriculture 142 (2017) 369–379.
[14] Kulkarni Anand H, Ashwin Patil RK. Applying image processing technique to detect plant diseases. Int J Mod Eng Res 2012;2(5):3661–4.
[15] https://courses.engr.illinois.edu/cs445/fa2015/projects/final/grabcut cs445.pdf
[16] Olga Russakovsky*, Jia Deng*, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg and Li Fei-Fei. (* = equal contribution) ImageNet Large Scale Visual Recognition Challenge. IJCV, 2015.