Plant Disease Explanation

Kausik Lakkaraju, Doctoral Student, University of South Carolina

December 2021

Abstract

Many countries lose their annual crop yield due to various plant diseases. The detection of these diseases is difficult due to lack of proper lab equipment. Though many researchers tried to solve this problem using various neural network classifiers, it is difficult to know the decision-making process of the neural network model. The aim of the project is to build a neural network classifier which would detect the name of the plant and identify the disease, if it exists. In addition to this, the system would also provide the explanation on why the system had to take that particular decision for a certain sample.

1 Problem

Many countries lose some part of their annual crop yield due to various plant diseases. In the case of India, it is 35% every year. Detection of plant diseases is difficult due to lack of proper lab infrastructure or equipment. In order to overcome this problem, many researchers have come up with datasets like PlantVillage [6] and PlantDoc [11] for the detection of various plant diseases. Despite their high accuracy when trained on such datasets, it is difficult to know the decision making process of a neural network model. Especially, the features and the logics it considers to classify a particular image. In order to overcome this problem, explainability methods can be used to justify the decision taken by the neural network model in order to make a prediction. This will make the system more transparent and trustworthy to the user.

[7] An AI-powered medical diagnosis system which wrongly diagnoses a patient to not have a cancer when they actually have cancer or an autonomous vehicle that drives erratically and causes fatal accidents despite

regular road conditions are some of the cases in which transparency of the system is very essential. If the reason for the AI system's mistake is known, AI system developers could learn why the model took a wrong decision and they could build better systems in the future.

The aim of the project is to build a system which would predict the plant disease, if it exists, when given an image and would also provide an explanation on why the system had to take that particular decision.

2 Related work

To tackle the problem of early identification of plant diseases, many researchers have developed several neural network classifiers. Despite their high accuracy, they are still black-box models whose decision making process is not known. For example, in [1] the authors have built a neural netowrk classifier called EfficientNet. They trained this on the PlantVillage dataset which is one of the datasets that I used for this project. Though they achieved a very high accuracy of around 99%, this would not make their system trustworthy. In the recent years, focus on the transparency of an AI system has increased when building a robust AI system. [13] We can see an increase in trend for XAI.

In a survey paper [4], the authors have explained various XAI algorithms, evaluation metrics along with an interesting case study which showed the growth of XAI from 2007 to 2020.

In [8], the authors have built a deep neural network classifier and used explanation methods like GradCAM++ which was introduced in [3] was used here to detect plant diseases. The authors used Masked R-CNN model [5] for training. The prediction from the neural network is given to Grad-CAM explainer to localize the disease on the leaf. This is one of the papers that motivated my work. In this project, I have compared LIME explanation result obtained on a diseased leaf when I trained a Convolutional Neural Network (CNN) model on both the datasets that I used which will be explained in the data section of the methodology.

Another recent work in this area is [9] where the authors have used Single Layer Perceptron (SLP) as an Explainable AI (XAI) module. The main focus of the paper was the early diagnosis of drought stress in wheat plants. They have made use of RGB and TIR-based sensors. They have replaced the images with their feature vectors while training which led to faster training

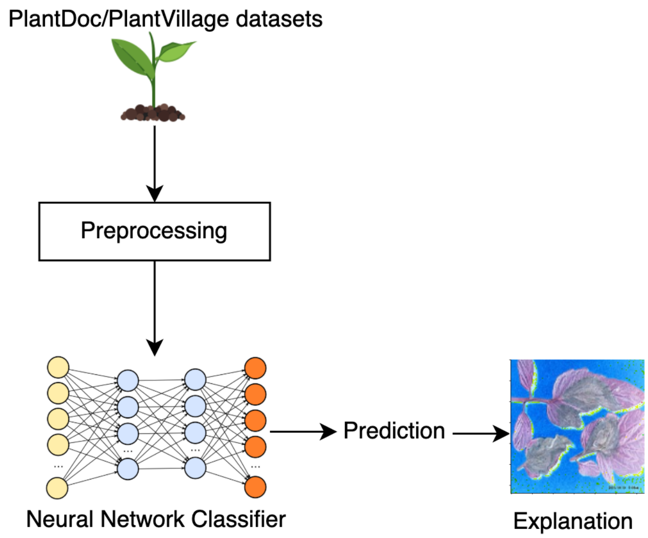


Figure 1: Overall workflow of the system

process.

Though the number of works done in the area of XAI in the plant disease detection setting is quiet low, various researchers have used XAI in medical diagnosis and in detection of special conditions in humans. One such recent work is [2] where the authors have built a model to detect a condition called Autism Spectrum Disorder (ASD). They have built a model which generates autism subtypes and identify discriminatory factors among them. They have used logistic regression and built a SHAP explainer on top of that.

Interesting works like these motivate AI researchers all around the world to pay more attention to trust aspect while building an AI system.

3 Methodology

Figure 1 shows the overall methodology of the system. The dataset will be pre-processed. The neural network model will be trained on the pre-processed data. The prediction of the neural network model will be used to provide a local explanation.

3.1 Data

Two datasets have been used in this project: PlantVillage and PlantDoc.

- **PlantVillage**: A well-known dataset with 54303 healthy and unhealthy leaf images divided into 38 categories by species and disease. All the images were taken in a laboratory setting with consistent lighting. I have taken only a subset of this dataset for the project as the whole dataset is very large. The subset that I have considered has 15 different classes of diseases and I have extracted 250 images from each of these classes.

- **PlantDoc**: The dataset contains 2,598 data points in total across 13 plant species and up to 17 classes of diseases. Unlike PlantVillage, PlantDoc dataset has images taken in real-world. In the PlantDoc paper, the authors have pointed out that their dataset performed better on real-world images compared to PlantVillage. I have considered all the disease classes given in the PlantDoc dataset but I have extracted 250 images from each of these classes.

3.2 User

The end-user can be a person with no knowledge in the domain. This system can be deployed as a mobile application which can be used by such people. They could take a picture of the plant and the model would predict the plant type, the disease (if it is diseased) and also the explanation for the model prediction.

This system can also be used by the government for constant monitoring of large agricultural fields for the welfare of farmers. This wold be possible by hovering drones over such fields.

3.3 Pre-processing

The pre-processing steps that have been followed before training the model are:

- Converting images to arrays so that it can be processed by neural networks. The images cannot be converted to grayscale for this problem as the color of the spots also plays an important role to determine whether or not it is diseased. For example, a leaf might be under shadow or has a big brown bacterial spot on it. This cannot be differentiated properly in grayscale.

- Converting the image labels to binary arrays for easier training.
- Normalizing the pixel values by dividing it by 225 so that the values can be comparable.
- Augmented the dataset by applying rotation, shifting of pixel values, shearing the image so that the image will be distorted in one angle and will have different perception angles, zooming and horizontal flipping. These augmentation techniques add more variations to the image data so that even if the user takes an image from a different angle or position, the model would still be able to recognize it.

3.4 Classification

For this project, I have used 2 kinds of classifiers: CNN and InceptionV3.

- **Convolutional Neural Network**: A Convolutional Neural Network is a class of artificial neural network, most commonly applied to analyze visual imagery. I have used a 5-layer CNN with 5-sets of 2D Convolutional layers, ReLU Activation layers, Batch Normalization layers, Max Pooling layers and Dropout layers. At the end, I have added some Dense and Batch Normalization layers. The whole architecture can be seen in the provided colab notebooks. I have trained this model on the
- **InceptionV3**: Inception v3 [12] is a convolutional neural network for assisting in image analysis and object detection, and got its start as a module for Googlenet. It is the third edition of Google's Inception Convolutional Neural Network, originally introduced during the ImageNet Recognition Challenge [14]. I initialized the model with pre-trained imagenet weights and applied transfer learning to use it for plant disease detection. I have applied 3 dense layers at the end. Two with ReLU activation functions and one with softmax activation function which gives out a multinomial probability distribution as output.

3.5 Explanation

To provide an explanation for the black-box models, I have used the LIME (Local Interpretable Model-agnostic Explanations) explainer [10]. It creates a local linear approximation for the model behavior. After training the model on the datasets, I have obtained the prediction of the model on an image which is not present in either of the datasets. The results can be seen in section 4.1.2. It highlights the portions of the image which have contributed to the decision taken by the model for that particular image.

3.6 Human Values

As explanation adds transparency to the model, this makes the user trust the system more. This will make the user appreciate the system more and take right decision by verifying the model prediction with an explanation.

3.7 Human-AI issues

As many other AI systems, this system can also be prone to misclassification and adversarial attacks. Having just a robust model architecture would not be enough, the data should be suitable as well. The authors in the PlantDoc paper have pointed out that the PlantVillage dataset does not produce accurate results on real-world images as all the images in the PlantVillage dataset were taken in laboratory conditions. These kind of issues might make it hard for the user to use this system when deployed.

4 Evaluation

I have evaluated the two models: CNN and InceptionV3 on both PlantVillage and PlantDoc datasets. Section 4.0.1 shows the model performance on their test sets after training. In Section 4.0.2, I will be discussing about the results that I obtained from the explainers when tested on an image which is not present in either of the datasets.

4.0.1 Test Results

The metrics that I used here are: Accuracy of the model on test set after splitting the data into training set and testing set with 80:20 ratio and train-

Dataset	Model	Recall	Precision	F1-score	Accuracy (%)	BCE Loss
PlantDoc	CNN	0.22	0.43	0.28	31.10	0.13
	InceptionV3	0.48	0.55	0.51	49.61	0.13
PlantVillage	CNN	0.78	0.82	0.80	80.16	0.09
	InceptionV3	0.81	0.83	0.82	81.81	0.07

Table 1: Recall, Precision, F1-score, Test set accuracy and BCE Loss are the metrics that I have included while training the models. These are the results when the models are tested on the test datasets.

ing the model on the training set, BCE loss which is the binary cross-entropy loss while testing, F1-Score, Precision and Recall. I consider the F1-score to be more prominent than accuracy in case of the data I am using as there might be some uneven class distribution. I have trained the CNN and InceptionV3 on PlantDoc for 60 epochs each. I have trained the same models on PlantVillage for 30 epochs each. These results can be seen in Table 1.

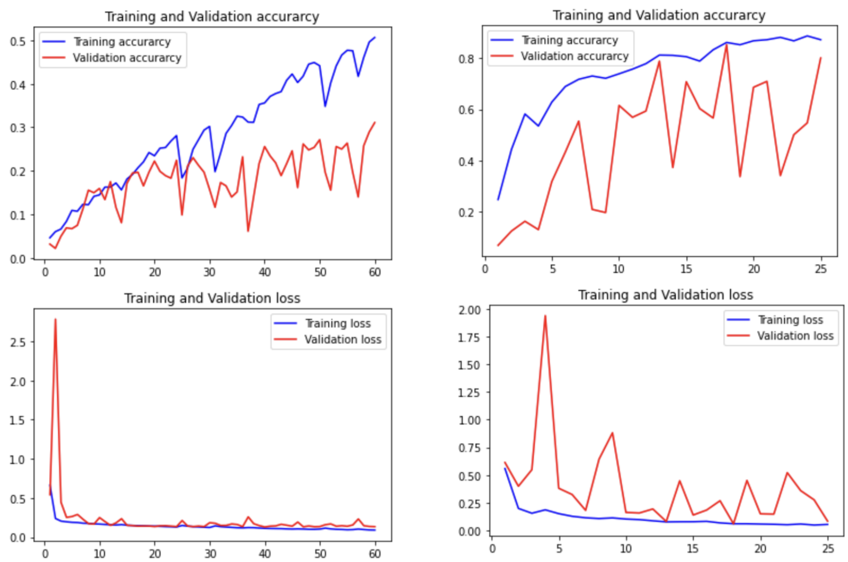
The Accuracy Vs. Epochs and BCE Loss Vs. Epochs graphs for both the datasets and both the models can be seen in Figure 2 and Figure 3 respectively.

From the tabular results and the training progression, we can say that the models performed better when trained on the PlantVillage dataset. These scores say how well the model performed on those datasets but does not give us any insights on the decision making process of the decision making process of the neural network model. In the next section, I will present a small case study which would introduce the transparency aspect of the project.

4.0.2 Case Study

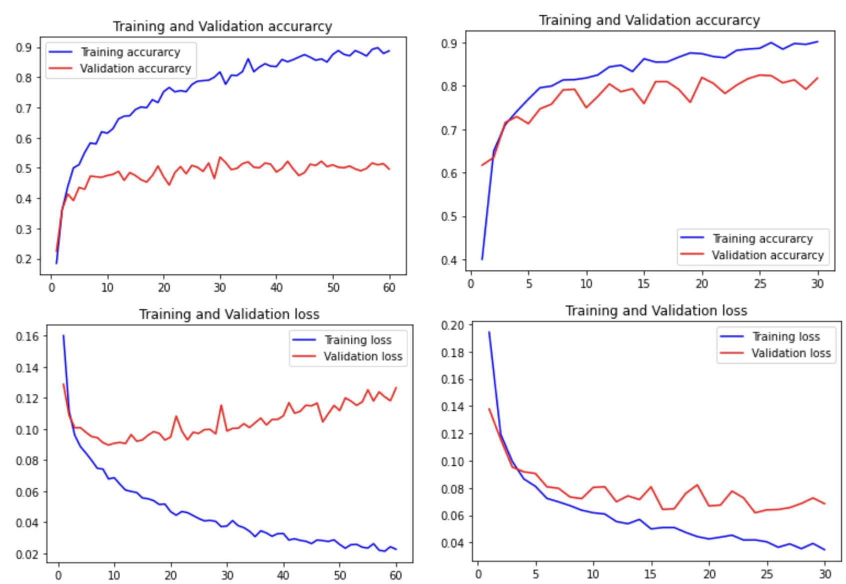
I will use a simpler notation of '¡dataset name¿-¡model name¿' while describing the models. For example, a CNN model trained on PlantDoc dataset will be called as PD-CNN and InceptionV3 trained on PlantDoc will be called PD-IV3.

In this section, I present a small case study in which I have tested all the four trained models on a sample image which is not present in either of the datasets. I used LIME explainer to get insights on what features contributed for the model prediction. Figure 4 shows the results from the explainer.



(a) PlantDoc (b) PlantVillage

Figure 2: PlantDoc and PlantVillage CNN training progression.



(a) PlantDoc (b) PlantVillage

Figure 3: PlantDoc and PlantVillage InceptionV3 training progression

The first figure is the original image that I used for testing. It is a tomato leaf affected by late blight. The second and third figures show the LIME explainer's output from PD-CNN and PD-IV3 respectively. They have predicted the image to be tomato leaf and blueberry leaf respectively. The fourth and fifth figures show the LIME explainer's output from PV-CNN and PV-IV3 respectively. They have predicted the image to be tomato septoria leaf spot and tomato late blight. The models that predicted the plant names correctly are PD-CNN, PV-CNN and PV-IV3. The only model that predicted the disease name also correctly is PV-IV3. The LIME has highlighted a part of the leaf's boundary for all the 3 whereas the PV-IV3 has higlighted the diseased spots along with the plant shape. That is the reason why PV-IV3 was able to predict the disease name along with the plant name correctly.

5 Conclusion

The conclusions that I were able to draw from the results that I obtained are PlantDoc dataset is not as good as they described in the paper as the images were not processed/cropped properly. Most of the images have watermark. Few of the images were not taken in natural lighting. They have collected these images from various other sources.PlantVillage dataset is huge and consistent. The only drawback is its lack of accuracy on real-world images as pointed out in the PlantDoc paper.

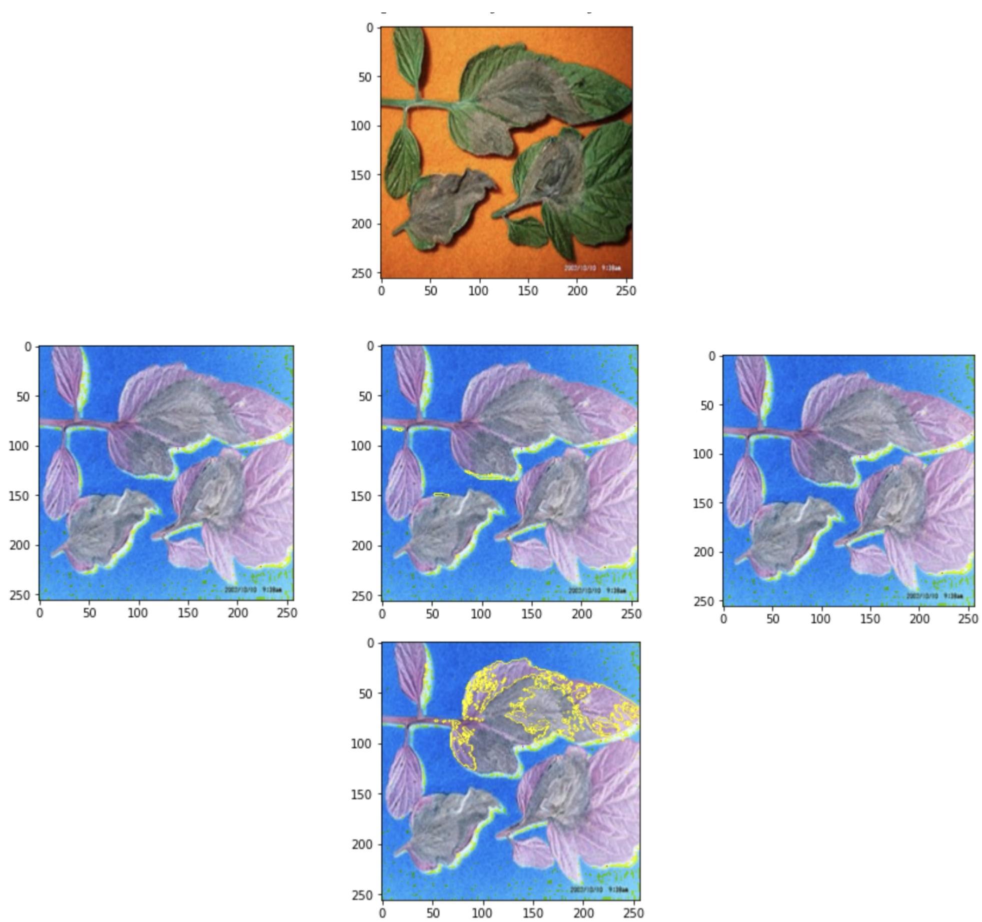


Figure 4: LIME explainer output from each of the trained models.

References

[1] Ümit Atila, Murat Uçar, Kemal Akyol, and Emine Uçar. Plant leaf disease classification using efficientnet deep learning model. *Ecological Informatics*, 61:101182, 2021.

[2] Milon Biswas, M. Shamim Kaiser, Mufti Mahmud, Shamim Al Mamun, Md. Shahadat Hossain, and Muhammad Arifur Rahman. An xai based autism detection: The context behind the detection. In Mufti Mahmud, M. Shamim Kaiser, Stefano Vassanelli, Qionghai Dai, and Ning Zhong, editors, *Brain Informatics*, pages 448–459, Cham, 2021. Springer International Publishing.

[3] Aditya Chattopadhyay, Anirban Sarkar, Prantik Howlader, and Vineeth N. Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. *CoRR*, abs/1710.11063, 2017.

[4] Arun Das and Paul Rad. Opportunities and challenges in explainable artificial intelligence (XAI): A survey. *CoRR*, abs/2006.11371, 2020.

[5] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017.

[6] David P. Hughes and Marcel Salath'e . An open access repository of images on plant health to enable the development of mobile disease diagnostics through machine learning and crowdsourcing. *CoRR*, abs/1511.08060, 2015.

[7] Bahador Khaleghi. The why of explainable ai. 2019.

[8] Shakti Kinger and Vrushali Kulkarni. Explainable ai for deep learning based disease detection. In *2021 Thirteenth International Conference on Contemporary Computing (IC3-2021)*, pages 209–216, 2021.

[9] Irina Maximova, Evgeny Vasiliev, Alexandra Getmanskaya, Dmitry Kior, Vladimir Sukhov, Vladimir Vodeneev, and Vadim Turlapov. Study of xai-capabilities for early diagnosis of plant drought. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2021.

[10] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 1135–1144, 2016.

[11] Davinder Singh, Naman Jain, Pranjali Jain, Pratik Kayal, Sudhakar Kumawat, and Nipun Batra. Plantdoc: a dataset for visual plant disease detection. In *Proceedings of the 7th ACM IKDD CoDS and 25th COMAD*, pages 249–253. 2020.

[12] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. *CoRR*, abs/1512.00567, 2015.

[13] Google Trends. https://trends.google.com/trends/explore?date=2016-01-01%202019-07-07q=explainable%20ai.

[14] Wikipedia. https://en.wikipedia.org/wiki/inceptionv3.