

PlantDoc: A Dataset for Visual Plant Disease Detection

Davinder Singh*, Naman Jain*, Pranjali Jain*, Pratik Kayal*

Sudhakar Kumawat, Nipun Batra

Indian Institute of Technology Gandhinagar, Gujarat, India 382 355

{davinder.singh,naman.j,pranjali.jain,pratik.kayal,sudhakar.kumawat,nipun.batra}@iitgn.ac.in

ABSTRACT

India loses 35% of the annual crop yield due to plant diseases. Early detection of plant diseases remains difficult due to the lack of lab infrastructure and expertise. In this paper, we explore the possibility of computer vision approaches for scalable and early plant disease detection. The lack of availability of sufficiently large-scale non-lab data set remains a major challenge for enabling vision based plant disease detection. Against this background, we present PlantDoc: a dataset for visual plant disease detection. Our dataset contains 2,598 data points in total across 13 plant species and up to 17 classes of diseases, involving approximately 300 human hours of effort in annotating internet scraped images. To show the efficacy of our dataset, we learn 3 models for the task of plant disease classification. Our results show that modelling using our dataset can increase the classification accuracy by up to 31%. We believe that our dataset can help reduce the entry barrier of computer vision techniques in plant disease detection.

KEYWORDS

Deep Learning, Object Detection, Image Classification

1 INTRODUCTION

Annually the Earth's population increases by about 1.6%, and so does the demand for plant products of every kind [16]. The protection of crops against plant diseases has a vital role to play in meeting the growing demand for food quality and quantity [22]. In terms of economic value, plant diseases alone cost the global economy around US\$220 billion annually [1]. According to the Indian Council of Agricultural Research, more than 35% of crop production is lost every year due to Pests and Disease [15]. Food security is threatened by an alarming increase in the number of outbreaks of pests and plant diseases. These diseases jeopardize food security and have broad economic, social, and environmental impacts [5].

Timely disease detection in plants remains a challenging task for farmers. They do not have many options other than consulting fellow farmers or the Kisan helpline [17]. Expertise in plant diseases is necessary for an individual to be able to identify the diseased leaves. Furthermore, in most cases it is necessary to have a lab infrastructure to identify a diseased leaf.

In this work, we explore the possibility of using computer vision for scalable and cost-effective plant disease detection. Computer vision has made tremendous advances in the past few years through various advances in deep convolutional neural networks. While training large neural networks can be very time consuming, the trained models can classify images very quickly, which makes them also suitable for consumer applications on smartphones. Image

processing for detecting plant diseases opens up new avenues to combine the knowledge of deep learning approaches with real-world problems in agriculture, and hence, facilitates advancements in agricultural knowledge, the yield of crops, and disease control.

Majority of existing vision-based solutions require high-resolution images with a plain background. In contrast, as the majority of Indian farmers use low-end mobile devices with natural background and lighting conditions, we focus on images in natural environmental conditions with non-trivial background noise and provide the best possible query resolution for crops and plants. Against this background, we highlight our two main contributions: i) development of PlantDoc: a dataset of 2,598 images across 13 plant species and 27 classes(17-10, disease-healthy) ii) benchmarking the curated data set and showing its utility in disease detection in non-controlled environments. To the best of our knowledge, this is the first such dataset containing data from non-controlled settings.

We evaluated our dataset using various classification and object detection architectures mentioned in Section 4 to establish the requirement of a dataset in non-controlled settings. The results suggested that lab-controlled dataset cannot be used to classify or detect images in real-scenario. We found that fine-tuning the models on PlantDoc reduces the classification error by up to 31%. Thus, our dataset can potentially be used to build an application which detects and classifies 27 plant disease/healthy classes efficiently.

2 RELATED WORK

Our related work can be broadly categorized into: i) techniques for plant disease detection; and ii) datasets advancing research in plant disease detection.

2.1 Techniques for plant disease detection

Prior work by Sankaran et al. [19] proposed using reliable sensors for monitoring health and diseases in plants under field conditions. However, plant disease detection using sensors has the potential to benefit only a few farmers because of the substantial hardware cost and lack of expertise to operate such sensors. In contrast, prior work by Patil et al. [18] extracted shape features for disease detection in sugarcane leaves obtaining a final average accuracy of 98.60%. In a similar work, Patil et al. [3] used texture features, namely inertia, homogeneity, and correlation obtained by calculating the gray level co-occurrence matrix on the image and color extraction for disease detection on maize leaves. Recent work [8] has looked into neural networks for the identification of three different legume species based on the morphological patterns of leaves veins. Likewise, feature extraction and Neural Network Ensemble (NNE) have been used for recognizing tea leaf diseases with a final testing accuracy of 91% [25]. A host of other recent works have looked at convolutional neural network variants for disease detection using plant leaf

*Equal Contribution.

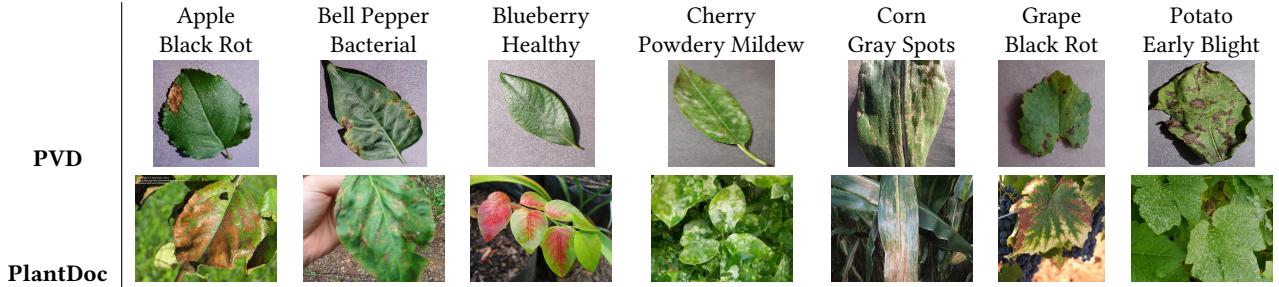


Figure 1: Samples from various classes in the PlantDoc Dataset show the gap between lab-controlled and real-life images

images [7, 21]. These works are limited to a particular crop, which is a significant limitation. Also, the datasets used in the works have not been made public, thereby, impacting reproducibility.

2.2 Datasets for plant disease detection

The PlantVillage dataset(PVD) [14] is the only public dataset for plant disease detection to the best of our knowledge. The data set curators created an automated system using GoogleNet [23] and AlexNet [12] for disease detection, achieving an accuracy of 99.35%. However, the images in PlantVillage dataset are taken in laboratory setups and not in the real conditions of cultivation fields, due to which their efficacy in real world is likely to be poor. In contrast, we curate real-life images of healthy and diseased plants to create a publicly available dataset.

3 THE PLANTDOC DATASET

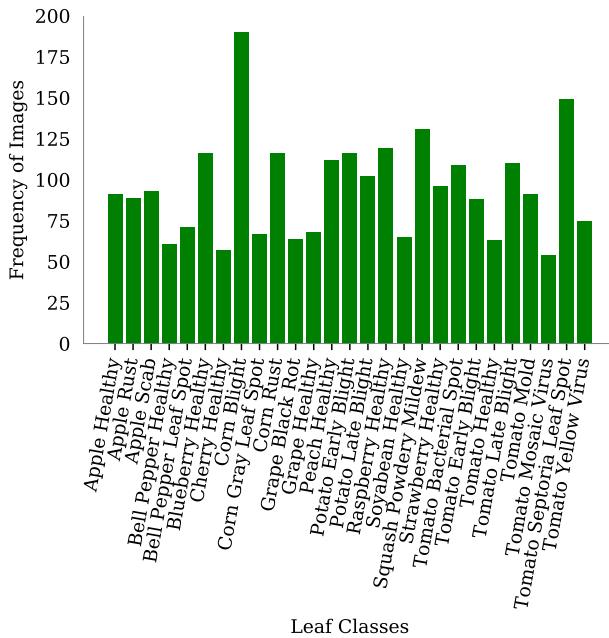


Figure 2: Statistics of PlantDoc Dataset

The PlantVillage dataset contains images taken under controlled settings. This dataset limits the effectiveness of detecting diseases because, in reality, plant images may contain multiple leaves with different types of background conditions with varying lighting conditions (shown in Figure 1). Against this background, we now describe our curated dataset and discuss the techniques used for curation.

3.1 Data Collection

To account for the intricacies of the real world, we require models trained on real-life images. This fact motivated us to create a dataset by downloading images from Google Images and Ecosia [6] for accurate plant disease detection in the farm setting. We downloaded images from the internet since collecting large-scale plant disease data through fieldwork requires enormous effort. We collected about 20,900 images by using scientific and common names of 38 classes mentioned in the dataset by Mohanty et al. [14].

Four users filtered the images by selecting images based on their metadata on the website and guidelines mentioned on APSNet [2]. APS compiled a list of peer-reviewed literature corresponding to each plant disease. We referred APS' prior literature and accordingly classified images. Some of the most important factors for classification were the color, area and density of the diseased part and shape of the species. We removed inappropriate (such as non-leaf plant, lab controlled and out-of-scope images) and duplicate images across classes downloaded due to web search. Every image was checked by two individuals according to the guidelines to reduce labeling errors. Finally, to have sufficient training samples, we removed the classes with less than 50 images. Figure 2 shows the statistics of the final dataset having a total of 27 classes spanning over 13 species with 2,598 images.

To build an application for the object detection task, we need exact bounding regions containing the leaf in the entire image. Hence, we used the LabelImg tool [24] to make the bounding boxes around the leaves (Figure 3) in all the images. In real scenarios, the image may have multiple leaves or a combination of diseased and healthy leaves. We labeled all the leaves in the image explicitly with their particular classes. While labeling the boxes, we made sure that the entire leaf should be present inside the box and the area of the bounding box should not be smaller than 1/8th (approximately) of the image size. After labeling, the information about all the coordinates of boxes in an image and their respective class label were stored separately in an XML file corresponding to each image.

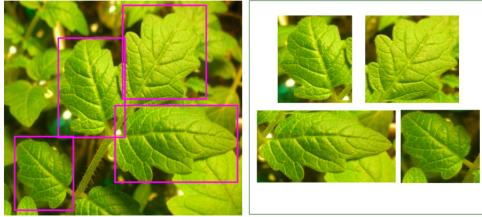


Figure 3: An Image with bounding boxes and its cropped leaves

Cropped-PlantDoc Dataset: To show the differences between our dataset and PlantVillage, we built another dataset called the *Cropped-PlantDoc (C-PD)* by cropping the images using bounding box information. Similar to PlantVillage, cropped images contains only the leaf but these images are of low-quality, have small-size and varying backgrounds. The total number of leaf images after cropping 2,598 images turns out to be 9,216 i.e. 9,216 bounding boxes.

4 BENCHMARKING PLANTDOC DATASET

We now discuss two benchmark set of experiments on our dataset: i) plant image classification; and ii) detecting leaf within an image.

PreTrained Weights	Training Set (Set %)	Test Set (Set %)	Accuracy	F1-Score
ImageNet	PlantDoc (80)	PlantDoc (20)	13.74	0.12
ImageNet	PVD	PlantDoc (100)	15.08	0.15
ImageNet+PVD	PlantDoc (80)	PlantDoc (20)	29.73	0.28

Table 1: Transfer Learning doubled the accuracy after fine-tuning on Uncropped PlantDoc dataset

4.1 System configuration

All our experiments used NVidia V100 GPU with a system of 32 GB RAM and 8 CPU cores. We used Keras with Tensorflow backend as the deep learning framework. We plan to make the fully reproducible Github repository^{1,2} for code and dataset.

4.2 Plant image classification

Our main goal was to construct a model which can detect a leaf in an image and then classify it into the particular classes shown in Figure 2. We performed two main experiments, which we discuss after describing our experimental settings.

4.2.1 Experimental settings. For training the networks, we used stochastic gradient descent with momentum 0.9, categorical cross-entropy loss, and a learning rate of 0.001. All weights were initialized with the orthogonal initializer. We applied common data augmentation techniques such as rotation, scaling, flipping etc. on the input images. All images were resized to 100×100 , before feeding into the networks. For pre-trained models, we used the weights provided in Keras trained on ImageNet.

¹<https://github.com/pratikkayal/PlantDoc-Object-Detection-Dataset>

²<https://github.com/pratikkayal/PlantDoc-Dataset>

Model	PreTrained Weights	Training Set (Set %)	Test Set (Set %)	Accuracy	F1-Score
VGG16	ImageNet	C-PD (80)	C-PD(20)	44.52	0.44
VGG16	ImageNet	PVD	C-PD (100)	19.73	0.18
VGG16	ImageNet+PVD	C-PD (80)	C-PD (20)	60.41	0.60
InceptionV3	ImageNet	C-PD (80)	C-PD (20)	46.67	0.46
InceptionV3	ImageNet	PVD	C-PD (100)	30.78	0.28
InceptionV3	ImageNet+PVD	C-PD (80)	C-PD (20)	62.06	0.61
InceptionResNet V2	ImageNet	C-PD (80)	C-PD (20)	49.04	0.49
InceptionResNet V2	ImageNet	PVD	C-PD (100)	39.87	0.38
InceptionResNet V2	ImageNet+PVD	C-PD (80)	C-PD (20)	70.53	0.70

Table 2: Training on controlled dataset (PlantVillage - PVD) gives poor performance on real world images. Performance on real world images can be improved by training on real world images from our dataset

Model	PreTrained Weights	mAP (at 50% iou)
MobileNet	COCO	32.8
MobileNet	COCO+PVD	22.4
Faster-rcnn-inception-resnet	iNaturalist	36.1
Faster-rcnn-inception-resnet	COCO	38.9

Table 3: Leaf detection mAP

* **iou** refers to intersection over union

4.2.2 Plant image classification using raw images (uncropped). Our first experiments aims to understand classification accuracy on the uncropped PlantDoc dataset. We evaluated the performance of VGG16 [20] using different training sets on PlantDoc as shown in Table 1.

4.2.3 Plant image classification using cropped images. Further, we evaluate the performance of several popular CNN architectures on the Cropped-PlantDoc dataset that have recently achieved state-of-the-art accuracies on image classification tasks on the popular datasets, such as ImageNet [4], CIFAR-10 [11], etc. Table 2 gives the complete list of the architectures that we used for benchmarking our Cropped-PlantDoc dataset. This experiment was conducted to verify the performance of PlantVillage in real-setting.

4.3 Leaf Detection

The aim of our next experiments is to evaluate the performance of Faster R-CNN with InceptionResnetV2 model and MobileNet model on our PlantDoc Dataset as shown in Table 3. We use mean average precision (mAP: higher is better) to evaluate the models and compare it with scores on COCO dataset since no evaluation exists in the domain of plant disease.

4.3.1 Experimental Setting. Object Detection models require training for a much longer duration. For training Faster R-CNN with Inception Resnet v2 network, we used Momentum optimizer keeping a degrading learning rate with an initial value of 0.0006. For training the MobileNet network, with RMSprop as optimizer – we took an initial learning rate of 0.0005 with decay steps as 25000 and decay factor as 0.95. While training, data augmentation like random horizontal flip and random SSD crop was applied on input images. We split our dataset into 2,360-238 based on training-testing. We took the pre-trained weights and fine-tuned on training set of PlantDoc. As aforementioned, we provide train-test splits of the dataset for consistent evaluation and fair comparison over the dataset in future.

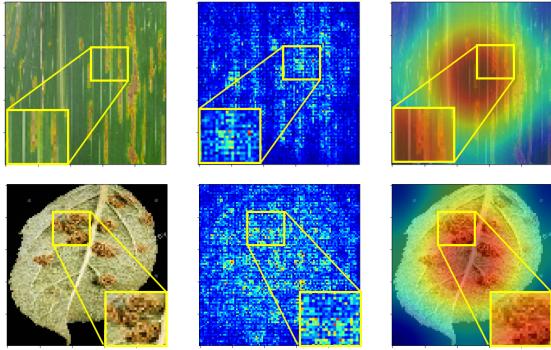


Figure 4: Saliency and Activation Maps shows the affected parts of disease in a leaf.

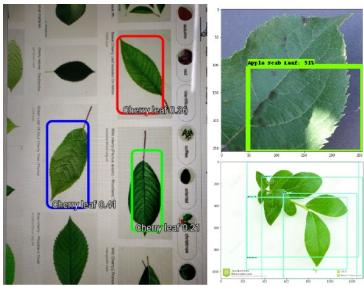


Figure 5: Leaf detection results in our mobile application

5 RESULTS AND DISCUSSION

Figure 4 shows saliency map and gradient activation map of the Corn Leaf Blight and Tomato Bacterial Spots respectively. As expected, the neural network is learning to focus on the set of visual features which are correlated with disease such as the blemishes in the leaf (lines in Corn Leaf Blight and spots in Tomato Bacterial Spots). The network even learns the shape of the leaf (shown in second row of Figure 4) to help it distinguish between species.

As predicted, the results in Table 1 clearly shows that real case scenarios have low accuracy when processed initially with ImageNet or PlantVillage. Also, Table 1 and Table 2 clearly shows low accuracy achieved by training on PlantVillage and testing on PlantDoc. Model fails to produce accurate results due to background noise, images with leaf from multiple classes in a dataset and low-resolution leaf images.

Table 3 shows that Faster R-CNN with InceptionResnetV2 performs the best with an mAP of 38.9. It is interesting to see that MobileNet performance is decreased when pre-trained on COCO+PlantVillage compared to the model where pre-training was done only on COCO. This attributes to the fact that PlantVillage is not contributing towards better results. MobileNet gives an mAP of 22 when evaluated on COCO dataset which has significantly more classes [9].



Figure 6: Tomato Bacterial leaf spot(a) and Septoria(b) looks similar and are hard to label using visual features alone

6 APPLICATION BUILDING

We were able to adapt the above solution to a mobile environment (Figure 5) by using models that very significantly reduce complexity, without sacrificing the effective accuracy. This allowed us to achieve the best possible performance, given that the application should predict the bounding boxes and classes in real time in a mobile CPU. We have built application that utilizes MobileNets Object Detection Network [10] due to its efficiency and competitive accuracy. The network builds on top of the SSD framework [13].

7 LIMITATIONS

The dataset has been curated with care, but due to lack of extensive domain expertise, there are some images in the dataset which can potentially be wrongly classified (shown in Figure 6). Further, to train highly accurate models for disease detection, we may require a dataset with more number of images in each class. But, due to non-availability of public dataset and lack of real-life scenario for field work, our approach gives a feasible direction to tackle the on-going problem of disease detection.

8 CONCLUSIONS AND FUTURE WORK

In this paper, we addressed the problem of detection of diseased/healthy leaves in images using state of the art object detection models. One of the main contributions of our work is to propose an entirely new dataset for plant disease detection called PlantDoc. Our benchmark experiments show the lack of efficacy of models learnt on controlled datasets, thereby, showing the significance of real-world datasets such as ours. Applying image segmentation techniques to extract leaf out of the images can potentially enhance the utility of the dataset. We believe that this dataset is an important first step towards computer vision enabled scalable plant disease detection.

REFERENCES

- [1] GN Agrios. 2005. Plant pathology 5th Edition: Elsevier Academic Press. *Burlington, Ma. USA* (2005), 79–103.
- [2] APSNet. 2019. Resources for Plant Diseases. <https://www.apsnet.org/edcenter/resources/commonnames/Pages/default.aspx>
- [3] Sanjay B Patil, K Shrikant, and Bodhe . 2011. Betel Leaf Area Measurement Using Image Processing. *International Journal on Computer Science and Engineering (IJCSE)* 3 (01 2011).
- [4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 248–255.
- [5] Messe Düsseldorf. [n. d.]. SAVE FOOD. ([n. d.]). https://www.messe-duesseldorf.com/cgi-bin/md_home/lib/pub/tt.cgi/SAVE_FOOD.html?oid=121&lang=2&ticket=g_ue_st
- [6] Ecosia. 2019. Search Engine. <https://www.ecosia.org/?c=en>
- [7] Alvaro Fuentes, Sook Yoon, Sang Kim, and Dong Park. 2017. A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. *Sensors* 17, 9 (2017), 2022.
- [8] Guillermo L Grinblat, Lucas C Uzal, Mónica G Larese, and Pablo M Granitto. 2016. Deep learning for plant identification using vein morphological patterns. *Computers and Electronics in Agriculture* 127 (2016), 418–424.
- [9] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861* (2017).
- [10] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. 2017. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *CoRR* abs/1704.04861 (2017). arXiv:1704.04861 <http://arxiv.org/abs/1704.04861>
- [11] Alex Krizhevsky et al. 2009. *Learning multiple layers of features from tiny images*. Technical Report. Citeseer.
- [12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*. 1097–1105.
- [13] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. 2016. Ssd: Single shot multibox detector. In *European conference on computer vision*. Springer, 21–37.
- [14] Sharada P Mohanty, David P Hughes, and Marcel Salathé. 2016. Using deep learning for image-based plant disease detection. *Frontiers in plant science* 7 (2016), 1419.
- [15] T Mohapatra. 2018. ICAR News July-September 2018. Published in monthly newsletter, <https://www.icar.org.in/sites/default/files/ICARNewsJulySeptember2018.pdf>.
- [16] E-C Oerke, H-W Dehne, Fritz Schönbeck, and Adolf Weber. 2012. *Crop production and crop protection: estimated losses in major food and cash crops*. Elsevier.
- [17] Government of India. 2019. Kisan Knowledge Management System. <https://dackkms.gov.in/account/login.aspx>
- [18] Sanjay B Patil and Shrikant K Bodhe. 2011. Leaf disease severity measurement using image processing. *International Journal of Engineering and Technology* 3, 5 (2011), 297–301.
- [19] Sindhuja Sankaran, Ashish Mishra, Reza Ehsani, and Cristina Davis. 2010. A review of advanced techniques for detecting plant diseases. *Computers and Electronics in Agriculture* 72, 1 (2010), 1–13.
- [20] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).
- [21] Srdjan Sladojevic, Marko Arsenovic, Andras Anderla, Dubravko Culibrk, and Darko Stefanovic. 2016. Deep neural networks based recognition of plant diseases by leaf image classification. *Computational intelligence and neuroscience* 2016 (2016).
- [22] Richard N Strange and Peter R Scott. 2005. Plant disease: a threat to global food security. *Annu. Rev. Phytopathol.* 43 (2005), 83–116.
- [23] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1–9.
- [24] Tzutalin. 2015. LabelImg. Free Software: MIT License. <https://github.com/tzutalin/labelImg>
- [25] Zhi-Hua Zhou and SF Chen. 2002. Neural network ensemble. *CHINESE JOURNAL OF COMPUTERS-CHINESE EDITION-* 25, 1 (2002), 1–8.