Detecting Plant Diseases in Browser Using Efficient Deep Learning

Griffin Davis
The University of Texas at Dallas
800 W Campbell Rd, Richardson, TX 75080
gcd@utdallas.edu

Canton Zhou
The University of Texas at Dallas
800 W Campbell Rd, Richardson, TX 75080
canton.zhou@utdallas.edu

Arshia Elahi
The University of Texas at Dallas
800 W Campbell Rd, Richardson, TX 75080
arshia.elahi@utdallas.edu

Abstract

We examine the comparative performance of EfficientNet and MobileNet, two lightweight image classification neural networks, and AlexNet, a large-scale image classification network in detecting disease on the leaves of plants of 21 species. Both the predictive performance of the models as well as their computational efficiency are tested in multiple environments. To determine the ability of each of these models to be used away from computers with high computational resources, the models will be run in browsers on mobile devices.

1. Introduction

Costing the global economy an estimated $220 million yearly [5], plant disease has remarkable effects on the lives of both farmers and consumers. Plant disease management is a critical aspect of ensuring food production can meet the demands of a growing human population [4]. The most fundamental part of disease management is determining if a plant is diseased at all. Especially at large-scale farms, scrutinizing each and every leaf to check for signs of disease can be difficult and prohibitively expensive. Modern computer vision techniques allow for this process to be automated, allowing farmers to rapidly determine if their plants are suffering from a problem they can work to solve.

One of the difficulties in applying these machine learning techniques is the computational resources required to train and utilize them. In recent years, substantial research efforts in artificial intelligence have focused on rapidly expanding the scale and power of neural networks. These efforts have resulted in breakthrough inference performance from such models, but the advancements require extraordinary computational power [3].

Efforts to make neural network inference more efficient have often come through post-training pruning of the network. Intelligent pruning techniques allow for the size of the network to be reduced without harming accuracy, as demonstrated in various prior works [8, 13, 17]. Post-training pruning techniques, however, do not reduce the significant resources required to train large models and instead introduce an additional step to the model-building process. Efforts to prune networks at the time of initialization have not yet achieved the success of other pruning methods [7].

The difficulties of large models and pruning techniques can be avoided entirely by training small, efficient models from the start. In this work, we will present a comparison of traditional, large image classification models with smaller, more efficient models. We will assess the differences in accuracy while also considering the performance, both in training and inference. This will allow us to determine if smaller models can obtain sufficient accuracy while maintaining a size that enables their usage in a web browser on a mobile device.

By building models that can run within a browser on a mobile device, we can bring the powerful prediction capabilities of modern convolutional neural networks to the devices that farmers and other users carry with them. This will help close the gap between technologies farmers can easily use and the technologies that can innovate how they care for their crops.

2. Related Work

While plant disease detection has been widely studied, including through the application of convolutional neural networks, these methods often focus on individual plant types and large models that require significant computational resources. In this work we will aim to compare the performance of large models to that of small, efficient

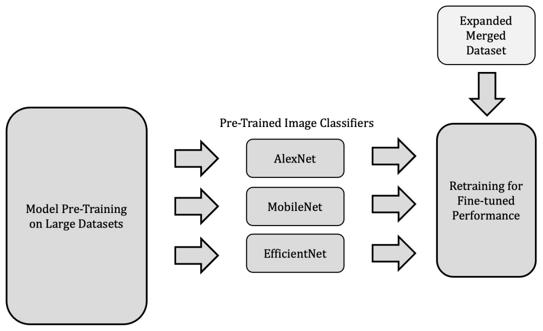
Model Pre-Training on Large Datasets
Pre-Trained Image Classifiers
AlexNet
MobileNet
EfficientNet
Expanded Merged Dataset
Retraining for Fine-tuned Performance

Figure 1. Model training process

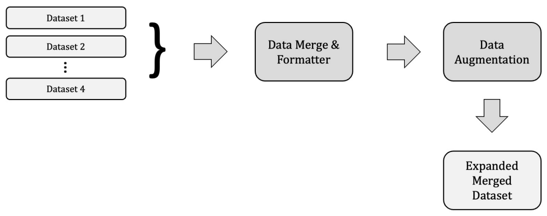

Figure 2. Data preparation and augmentation process

models on the task of detecting disease on a wide array of crop types. Mohanty et al. expand their study to 14 crop species and 26 diseases [15], but the authors rely on AlexNet and GoogleLeNet [20], which are not designed for mobile or browser environments. AlexNet in particular relies on a network of 60 million parameters and 650,000 neurons to power its high prediction accuracy [12]. Differently, Agarwal et al. and Mishra et al. each focus on a single crop species, focusing on tomato and corn plants respectively [1, 14].

3. Method

By combining multiple plant leaf disease datasets, we have constructed a dataset including multiple plant types. This data will be provided to pre-trained models to fine-tune their performance to the task of detecting plant disease. Model pre-training allows for the use and re-use of models that have been trained on large sets of data. In the case of image classification, these pre-training datasets assist the model in learning fundamental image classification tasks, preparing the model for future fine-tuning towards specific types of images or classes. The benefits of model pre-training have been widely demonstrated for a variety of tasks [9]. By utilizing pre-training, along with other techniques to expand the available data, we aim to construct models with high prediction performance with relatively minimal fine-tuning training time. The pre-training and re-training processes used to construct our models are described in Figure 1.

To expand the size of the training and testing data, we merged multiple publicly available plant leaf disease data sets and applied various data augmentation strategies. For the first trial, these strategies will be 30-degree rotations and image mirroring, in total expanding the dataset by over 8 times, excluding duplicates. In later trials, we will apply noise to the images and convert them to grayscale for a total of approximately 32 times the original data size. For a complete enumeration of these techniques and their effects on the data size, see Table 1. The data preparation process utilized in this work is illustrated in Figure 2. We will use a combination of publicly available data sets that have both annotated plant diseases and healthy leaves across a range of different plant species. By using images from different sources, the model will ideally generalize better to different conditions and plant diseases not seen at training time. Furthermore, by expanding the data with image augmentation, we can avoid biases that may be present in individual datasets [16] and improve generalization performance [18]. The wide array of plant types found in our merged dataset is vital to ensuring the model is usable in the context of a farm where multiple crop types may be present.

Table 1. Data augmentation strategies

Strategy	**Data Expansion**
30-degree rotations	4x
Image mirroring	2x
Image noise	2x
Grayscale	2x
Total augmentation	32x

This data will be used to train convolutional neural networks with differing architectures. We will train neural networks designed for usage in environments with low computational capacity, to determine both model performance and efficiency in environments that may be usable by a farmer in the field. The models selected for this portion of the project are MobileNet [10] and EfficientNet [21], both of which are designed to be lightweight and efficient. We will test both the accuracy of these models as well as the performance inside internet browsers on multiple device types to determine the optimal model for this usage. In future work, the application can allow for user input regarding the correctness of the model prediction to add new training data.

In addition to the lightweight models, we will train a large-scale, traditional model: AlexNet. This model, with its 65 million parameters [12], showcases the forefront of large-scale, powerful image classification models. By training and testing this model we can determine the difference in prediction performance between lightweight and heavyweight models. Moreover, we can determine the difference in computational performance in these types of models on multiple environment types.

4. Experiments

The data sources used for this project will include the PlantVillage data set, introduced by Hughes and Salathé [11]; Identification of Plant Leaf Diseases data from Arun and Geetharamani [2]; the PlantDoc data set, introduced by Singh et al. [19]; and a plant disease data set by Frabotta [6]. Prior to applying the augmentation process (as described in Figure 2), the merged datasets have a combined size of 62,682 images, covering 21 species. After the first round of augmentation, the dataset has an expanded size of over 500,000 images (see Figure 3). For future trials, we will apply additional image augmentation techniques to further expand the dataset (see Figure 4). The images will also be resized, cropped, and normalized in preparation for usage in training. The models will be trained using a binary cross entropy loss with Adam optimization.

The performance of the model will be evaluated based on both its accuracy and computational efficiency. For the accuracy of the model, we will compute metrics the precision (see Equation 1), recall (see Equation 2), and F_1 score (see Equation 3). This will be done by comparing true positives, true negatives with incorrectly classified examples on a testing data set. Efficiency will be measured by the time required to train the model initially, and then the amount of time it takes for the deployed model to output a result. A successful model is one that has both high accuracy and can be deployed reasonably quickly in a browser on commodity hardware.

$$\text{Precision(model)} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \quad (1)$$

$$\text{Recall(model)} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \quad (2)$$

$$F_1(\text{model}) = 2\frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \quad (3)$$

At this stage in our work, we have only trained and tested one trial of the AlexNet model using an 80-20 train-test data split. After 100 epochs of training, we obtained the performance noted in Table 2 on withheld test data. The model's loss curve in training is noted in Figure 5.

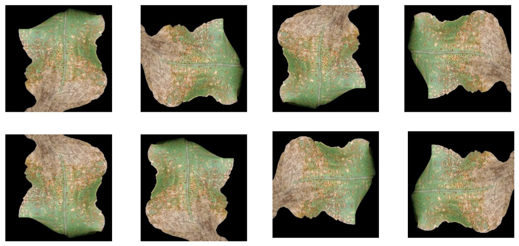

Figure 3. Sample image augmentation for first trial

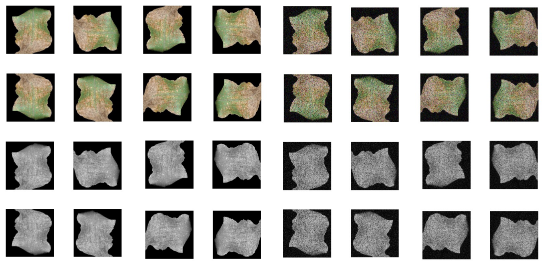

Figure 4. Sample image augmentation for second trial

Table 2. Trial 1 AlexNet Performance

Metric	**Performance**
Accuracy	97.1%
Precision	0.960069
Recall	0.970077
F_1 Score	0.965047
Training Time (CUDA)	324 minutes
Inference Time (CUDA)	0.0023 sec
Inference Time (CPU)	0.0253 sec

5. Next Steps

Prior to applying the next round of data augmentations, we will first train and test the A and B networks on the trial 1 data. We will compare the performance of these models using the metrics described in Section 3. Next, we will continue to expand the dataset through further data augmentation strategies. These will include applying image noise and grayscale transformations (see Table 1).

After these additional augmentations, we will re-apply the fine-tuning training to the pre-trained models to determine any change in performance. The models will be tested

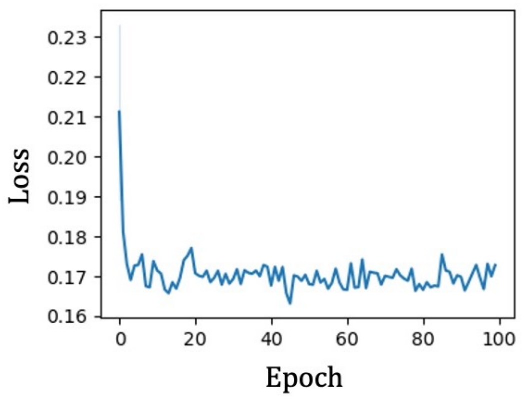

Figure 5. AlexNet loss across training epochs

on withheld test data to assess three predictors of prediction performance: precision, recall, and F_1 score. We will also run these models in the Internet browser of a mobile device to assess changes to their inference time in lightweight computational environments. Through each of these steps, the large-scale AlexNet model will be compared to the lightweight EfficientNet and MobileNet models.

References

[1] Mohit Agarwal, Abhishek Singh, Siddhartha Arjaria, Amit Sinha, and Suneet Gupta. ToLeD: Tomato Leaf Disease Detection using Convolution Neural Network. *Procedia Computer Science*, 167:293–301, Jan. 2020. 2

[2] J Arun Pandian and Gopal Geetharaamani. Data for: Identification of Plant Leaf Diseases Using a 9-layer Deep Convolutional Neural Network, Apr. 2019. Type: dataset. 3

[3] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners, July 2020. arXiv:2005.14165 [cs]. 1

[4] Jeremy J Burdon, Luke G Barrett, Li-Na yang, Dun-Chun He, and Jiasui Zhan. Maximizing World Food Production through Disease Control. *BioScience*, 70(2):126–128, Feb. 2020. 1

[5] Food and Argiculture Organization of the United Nations. FAO - News Article: New standards to curb the global spread of plant pests and diseases. 1

[6] Jessica Frabotta. Plant Diseases Dataset With Augmentation. 3

[7] Jonathan Frankle, Gintare Karolina Dziugaite, Daniel M. Roy, and Michael Carbin. Pruning Neural Networks at Initialization: Why are We Missing the Mark?, Mar. 2021. arXiv:2009.08576 [cs, stat]. 1

[8] Song Han, Jeff Pool, John Tran, and William J. Dally. Learning both Weights and Connections for Efficient Neural Networks, Oct. 2015. arXiv:1506.02626 [cs]. 1

[9] Xu Han, Zhengyan Zhang, Ning Ding, Yuxian Gu, Xiao Liu, Yuqi Huo, Jiezhong Qiu, Yuan Yao, Ao Zhang, Liang Zhang, Wentao Han, Minlie Huang, Qin Jin, Yanyan Lan, Yang Liu, Zhiyuan Liu, Zhiwu Lu, Xipeng Qiu, Ruihua Song, Jie Tang, Ji-Rong Wen, Jinhui Yuan, Wayne Xin Zhao, and Jun Zhu. Pre-trained models: Past, present and future. *AI Open*, 2:225–250, Jan. 2021. 2

[10] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications, Apr. 2017. arXiv:1704.04861 [cs]. 2

[11] David P Hughes and Marcel Salathé. An open access repository of images on plant health to enable the development of mobile disease diagnostics, 2015. 3

[12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. 2, 3

[13] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning Filters for Efficient ConvNets, Mar. 2017. arXiv:1608.08710 [cs]. 1

[14] Sumita Mishra, Rishabh Sachan, and Diksha Rajpal. Deep Convolutional Neural Network based Detection System for Real-time Corn Plant Disease Recognition. *Procedia Computer Science*, 167:2003–2010, Jan. 2020. 2

[15] Sharada P. Mohanty, David P. Hughes, and Marcel Salathé. Using Deep Learning for Image-Based Plant Disease Detection. *Frontiers in Plant Science*, 7:1419, Sept. 2016. 2

[16] Mehmet Alican Noyan. Uncovering bias in the PlantVillage dataset, June 2022. arXiv:2206.04374 [cs]. 2

[17] R. Reed. Pruning algorithms-a survey. *IEEE Transactions on Neural Networks*, 4(5):740–747, Sept. 1993. Conference Name: IEEE Transactions on Neural Networks. 1

[18] Connor Shorten and Taghi M. Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):60, 2019. 2

[19] 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, Jan. 2020. arXiv:1911.10317 [cs, eess]. 3

[20] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, Boston, MA, USA, June 2015. IEEE. 2

[21] Mingxing Tan and Quoc V. Le. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks, Sept. 2020. arXiv:1905.11946 [cs, stat]. 2