Plant Disease Classification from Leaf Images using Deep Learning

Hemanth Krishna Sai Ram Aradhyula [1] **Koushik Ponugoti** [1] **Lohith Reddy Nimma** [1]

Github Link: https://github.com/lohithreddy15/Plant-Disease-Classification

Abstract

Crop diseases are a major threat to food security, but their rapid identification remains difficult in many parts of the world due to the lack of the necessary infrastructure. The combination of increasing global smartphone penetration and recent advances in computer vision made possible by deep learning has paved the way for smartphone-assisted disease diagnosis. Using a public dataset of 54,306 images of diseased and healthy plant leaves collected under controlled conditions, we train a deep convolutional neural network to identify 14 crop species and 26 diseases (or absence thereof). The trained model achieves an accuracy of 99.684% on a held-out test set, demonstrating the feasibility of this approach. This wok also creates a GUI application where users can directly upload plant images and get information about the disease it may have. Overall, the approach of training deep learning models on increasingly large and publicly available image datasets presents a clear path toward smartphone-assisted crop disease diagnosis on a massive global scale.

1. Introduction

The development of new technologies has provided human civilization with the capability of producing sufficient food to satisfy the requirements of more than 7 billion people. Climate change (Tai et al., 2014), pollinator decline (Annex), plant diseases (Strange & Scott, 2005), and others continue to endanger food security . Plant diseases threaten global food security and can devastate smallholder farmers who rely on healthy crops. Smallholder farmers produce over 80% of agricultural output in developing countries (IFAD, 2013), and pests and diseases often cause yield losses of over 50%. (Harvey et al., 2014). Smallholder farmers, who account for 50% of hungry people (Sanchez & Swaminathan, 2005), are especially prone to pathogen-induced food supply disruptions.

Many initiatives have been created to stop agricultural damage from diseases. Integrated pest management (IPM) techniques have replaced traditional methods of applying poisons widely over the past ten years (Ehler, 2006). Whatever the method, the first stage in effective disease management is accurate illness identification when it first manifests.

Historically, entities like neighbourhood plant clinics or farm education groups have supported disease identification. More recently, these initiatives have also been helped by the availability of online resources for illness detection, taking advantage of the global increase in Internet usage. Even more lately, mobile phone-based tools have multiplied, capitalising on the historically unprecedented global adoption of mobile phone technology (Bureau, 2017).

Because of their processing power, high-resolution screens, and extensive built-in accessory sets, such as sophisticated HD cameras, smartphones in particular offer very innovative methods to aid in the identification of illnesses. By the end of 2015, mobile broadband service was available to 69 percent of the world's population, and mobile broadband usage hit 47 percent in 2015, a 12-fold rise since 2007. (Bureau, 2017). If automatic picture identification for disease detection is theoretically possible, it may be made widely accessible thanks to variables such as high definition cameras, high performance computers, and broad smartphone adoption. Here, we use 54,306 pictures of 14 crop species with 26 diseases (or healthy) that were made publicly accessible through the project PlantVillage (Hughes et al., 2015) to show the technological viability of a deep learning method. Fig. 1 shows a sample of each crop-disease combination.

Computer vision, and object recognition in particular, has made tremendous advances in the past few years. The PASCAL VOC Challenge (Everingham et al., 2010), and more recently the Large Scale Visual Recognition Challenge (ILSVRC) (Russakovsky et al., 2015) based on the ImageNet dataset (Deng et al., 2009) have been widely used as benchmarks for numerous visualization-related problems in computer vision, including object classification. In 2012,

[1]Department of Computer Science, University at Buffalo. Correspondence to: Hemanth Krishna Sai Ram Aradhyula, Koushik Ponugoti, Lohith Reddy Nimma < >.

This report is based on the ICML 2023 paper format

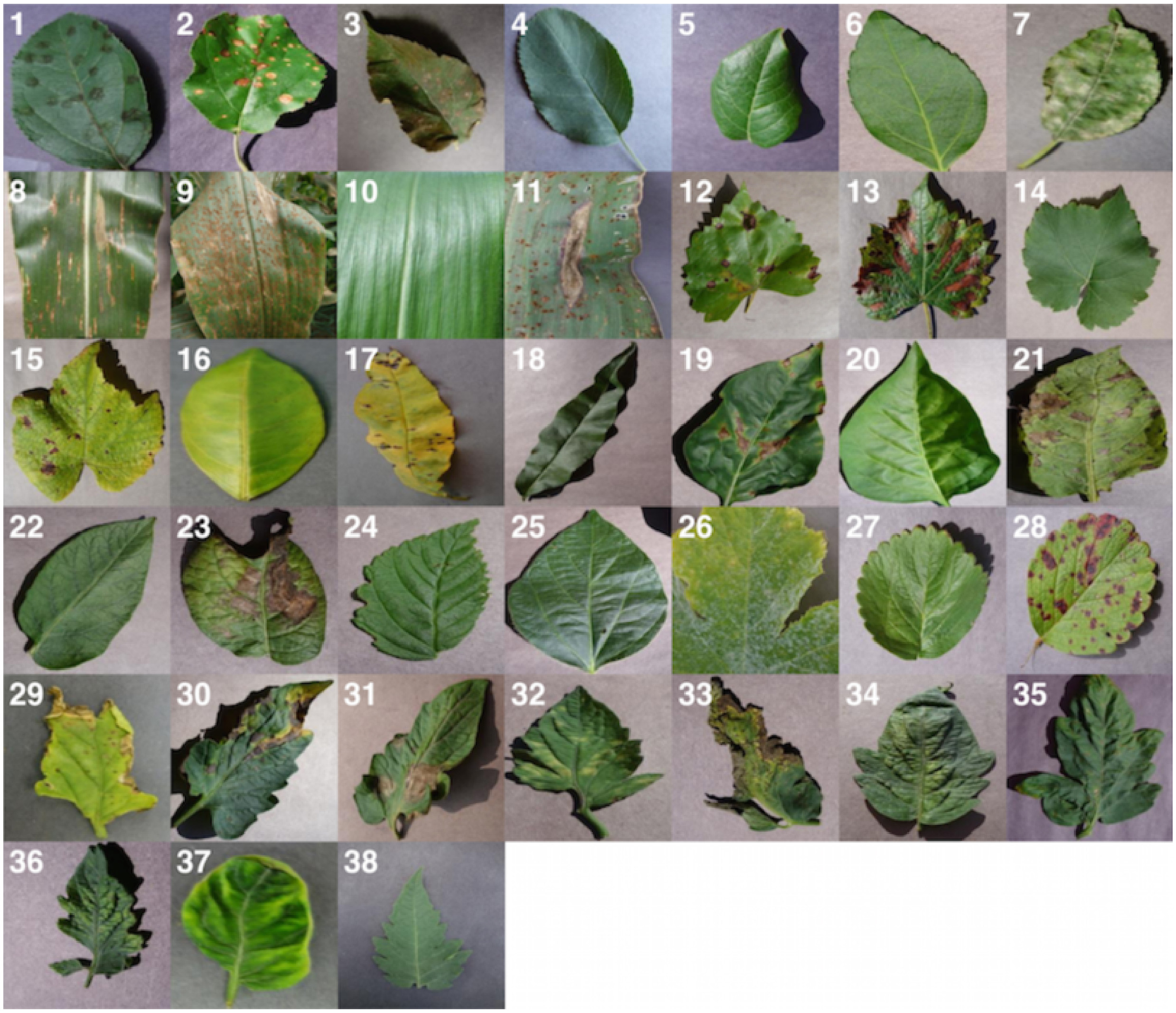


Figure 1. **Example of leaf images from the PlantVillage dataset, representing every crop-disease pair used.** (1) Apple Scab, Venturia inaequalis (2) Apple Black Rot, Botryosphaeria obtusa (3) Apple Cedar Rust, Gymnosporangium juniperi-virginianae (4) Apple healthy (5) Blueberry healthy (6) Cherry healthy (7) Cherry Powdery Mildew, Podoshaera clandestine (8) Corn Gray Leaf Spot, Cercospora zeae-maydis (9) Corn Common Rust, Puccinia sorghi (10) Corn healthy (11) Corn Northern Leaf Blight, Exserohilum turcicum (12) Grape Black Rot, Guignardia bidwellii, (13) Grape Black Measles (Esca), Phaeomoniella aleophilum, Phaeomoniella chlamydospora (14) Grape Healthy (15) Grape Leaf Blight, Pseudocercospora vitis (16) Orange Huanglongbing (Citrus Greening), Candidatus Liberibacter spp. (17) Peach Bacterial Spot, Xanthomonas campestris (18) Peach healthy (19) Bell Pepper Bacterial Spot, Xanthomonas campestris (20) Bell Pepper healthy (21) Potato Early Blight, Alternaria solani (22) Potato healthy (23) Potato Late Blight, Phytophthora infestans (24) Raspberry healthy (25) Soybean healthy (26) Squash Powdery Mildew, Erysiphe cichoracearum (27) Strawberry Healthy (28) Strawberry Leaf Scorch, Diplocarpon earlianum (29) Tomato Bacterial Spot, Xanthomonas campestris pv. vesicatoria (30) Tomato Early Blight, Alternaria solani (31) Tomato Late Blight, Phytophthora infestans (32) Tomato Leaf Mold, Passalora fulva (33) Tomato Septoria Leaf Spot, Septoria lycopersici (34) Tomato Two Spotted Spider Mite, Tetranychus urticae (35) Tomato Target Spot, Corynespora cassiicola (36) Tomato Mosaic Virus (37) Tomato Yellow Leaf Curl Virus (38) Tomato healthy.

a large, deep convolutional neural network achieved a top-5 error of 16.4% for the classification of images into 1000 possible categories (Krizhevsky et al., 2017). In the following 3 years, various advances in deep convolutional neural networks lowered the error rate to 3.57% (Krizhevsky et al., 2017; Simonyan & Zisserman, 2014; Zeiler & Fergus, 2014; He et al., 2016; Szegedy et al., 2015). Even though training large neural networks can take a long time, the trained models can quickly classify images, which makes them good for consumer applications on smartphones. Deep neural networks, which are a type of end-to-end learning, have recently been used successfully in many different fields. Neural networks make a connection between an input, like a picture of a sick plant, and an output, like a crop and a disease. In a neural network, the nodes are mathematical functions that take in numbers from the edges and output numbers on the edges going out. Deep neural networks are just a stack of layers of nodes that map the input layer to the output layer. The challenge is to make a deep network where both the structure of the network and the functions (nodes) and edge weights correctly map the input to the output. Deep neural networks are trained by making changes to the network parameters so that the mapping gets better as the network is trained. This process is hard to calculate, but it has gotten a lot better in recent years thanks to a number of breakthroughs in both ideas and technology (LeCun et al., 2015; Schmidhuber, 2015). To make accurate image classifiers for diagnosing plant diseases, we needed a large set of images of both sick and healthy plants that had been checked. Until not too long ago, there was no such dataset, and even smaller datasets were not free to use. To solve this problem, the PlantVillage project has started collecting tens of thousands of pictures of healthy and sick crop plants (Hughes et al., 2015) and has made them openly and freely available. Here, we describe how 54,306 images and a convolutional neural network were used to sort 26 diseases in 14 crop species. We judge how well our models work by how well they can pick the right crop-disease pair out of 38 possible classes. The best model gets 99.684% accuratecy overall. This shows that our approach is technically possible.

2. Methods

In this section, we first describe the dataset in detail followed by how we arrange the datasets and the performance metrics we use and finally we describe the methodology we used to classify among the diseases.

2.1. Dataset Description

We look at 54,306 pictures of plant leaves that have been put into 38 different classes. Each class label is a pair of a crop and a disease, and we try to predict the crop-disease pair

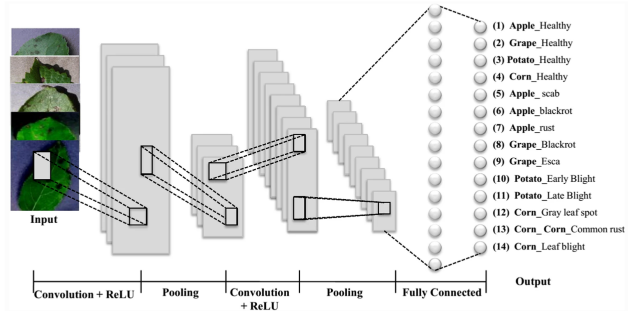


Figure 2. Computational graph of the model

from just an image of a plant leaf. Fig. 1 shows one example for every pair of crop and disease in the PlantVillage dataset. In all of the methods described in this paper, the images are resized to 256 $\times$ 256 pixels, and the model is optimised and predictions are made using these smaller images.

Across all our experiments, we use only the color images from the PlantVillage dataset. These images are taken in particular conditions which may differ from the real world scenarios (for example, the background has a simple texture and different color from the leaves). The distribution of images and these classes are not uniform; some classes have more than 5000 images while others have less than 200 samples. This imbalanced distribution makes the classification task more challenging.

2.2. Performance plan

To get a sense of how our approaches will perform on new, unseen data, and also to monitor whether any of our approaches are overfitting, we run all of our experiments with a variety of train-test set splits, including 80–20 (80% of the whole dataset used for training, and 20% for testing), 60–40 (60% of the whole dataset used for training, and 40% for testing), 50–50 (50% of the whole dataset used for training, and 50% for testing), 40–60 (40% of the whole dataset used for training, and 60% for testing) and finally 20–80 (20% of the whole dataset used for training, and 80% for testing).

In addition, for each experiment, we compute the mean precision, the mean recall, the mean F1 score, and the overall accuracy over the entire training period at predetermined intervals (at the end of every epoch). We utilise the final mean F1 score to compare results across all of the various experimental configurations.

2.3. Proposed Approach

We evaluate the applicability of deep convolutional neural networks for the classification problem described above. We focus on two popular architectures, namely DenseNet(Huang et al., 2017) and EfficientNet (Tan & Le, 2019).

2.3.1. DENSENET

The DenseNet architecture is a deep neural network designed for image classification tasks. It is built on the principle of densely connected convolutional layers, where each layer receives inputs not only from the previous layer, but also from all preceding layers. This enables the network to efficiently reuse features learned from earlier layers, and results in a compact yet powerful model.

The core building block of DenseNet is the dense block, which consists of multiple convolutional layers that are densely connected to each other. Each layer receives the feature maps from all preceding layers as input, and its output is passed on to all subsequent layers. The dense block is followed by a transition layer, which reduces the spatial dimensions of the feature maps using a pooling operation and a 1x1 convolutional layer. This helps to control the growth of feature maps and reduce the computational cost of the network.The output of the layer would be:

$$a_{k+1}(x, y) = \mathrm{ReLU}\left(\sum_{y=0}^{N} \omega_k x^* A_k(y) + \beta_{kx}\right)$$

where x represents the filter, k represents the layer, β indicates bias, ω indicates weight, N is the kernal width, $A_k(y)$ represents the y^{th} local region in k^{th} layer

DenseNet is characterized by its high accuracy and efficiency, particularly on small to medium-sized datasets. It achieves state-of-the-art performance on various benchmark datasets, such as CIFAR-10, CIFAR-100 (Krizhevsky et al., 2009), and ImageNet(Deng et al., 2009). In addition, DenseNet is able to learn more discriminative features than traditional architectures with fewer parameters, making it more memory-efficient and easier to train.

2.3.2. EFFICIENTNET

EfficientNet is a novel deep neural network architecture designed to achieve state-of-the-art performance on image classification tasks while maintaining computational efficiency. The architecture is based on a novel compound scaling method, which scales the network dimensions in a systematic and balanced way, leading to improved accuracy and efficiency compared to previous state-of-the-art models.

The key insight behind the EfficientNet architecture is that the optimal network architecture depends on the tradeoff between model size, computational cost, and accuracy. To find the optimal balance between these factors, the authors introduce a new scaling method that uniformly scales network width, depth, and resolution in a balanced way. The scaling coefficients are learned through a grid search over a range of possible values, resulting in a family of models that are efficient at different computational budgets.The number of parameters in the efficientnet can be calculated as parameters = $w^2 \cdot d^2 \cdot r \cdot c$ where w is the width multiplier, d is the depth multiplier, r is the resolution multiplier, c is the number of channels in the input image.

The EfficientNet architecture is composed of a backbone network and a top network. The backbone network consists of a series of blocks, each of which contains a sequence of convolutional layers, batch normalization, and activation functions. The authors propose a new block design, called the Mobile Inverted Residual Block (MBConv), which is based on depthwise separable convolutions and linear bottlenecks. The MBConv block reduces the number of parameters and computation while improving accuracy, and is used as the building block for the entire network.

The top network of EfficientNet consists of a global average pooling layer, a fully connected layer, and a softmax layer. For calculationg the floating-point operations(FLOPs) in an efficient model $flops = w^2 \cdot d^2 \cdot r^2 \cdot c \cdot kernel\_size^2$ .The network is trained end-to-end using a standard cross-entropy loss function, and is optimized using stochastic gradient descent with momentum.At the end. At the end of the CNN model the last features are extracted and are converted into one vector through a pooling operation where the feature vectors are added.Global Average Pooling(GAP) performs a simple average with equal weights which does not give any special attention to particular respective fields or regions of the input image.To calculate
$L_j = \mathrm{GAP}(L_{j-1}) = \frac{1}{M \times M} \sum_{i=0}^{M \times M-1} F_i$
$L_j = \mathrm{GAP\_ATN}(L_{j-1}) = \sum_{i=1}^{M \times M} w_i F_i$

where L_j and L_{j-1} are the input and output layers of the GAP function, w_i are the weights

The authors evaluate the performance of EfficientNet on several standard image classification datasets, including ImageNet, CIFAR-10, and CIFAR-100. They show that EfficientNet achieves state-of-the-art performance on these datasets while being significantly more efficient than previous models. Specifically, EfficientNet achieves a top-1 accuracy of 84.3% on ImageNet, while being 8.4x smaller and 6.1x faster than the previous state-of-the-art model.

Overall, EfficientNet is a highly efficient and effective architecture for image classification tasks, and represents a significant advance in the field of deep learning. The novel compound scaling method used in the architecture provides a systematic and balanced approach to optimizing the trade-off between model size, computational cost, and accuracy, and has the potential to be applied to other types of neural networks as well.

In terms of performance, both DenseNet and EfficientNet achieve state-of-the-art results on standard image classification datasets, such as ImageNet. However, EfficientNet typically achieves better accuracy and efficiency than DenseNet, especially on larger datasets where the benefits of compound scaling become more pronounced. For example, EfficientNet achieves a top-1 accuracy of 84.3% on ImageNet, while being significantly smaller and faster than previous state-of-the-art models. In contrast, DenseNet achieves a top-1 accuracy of 81.8% on ImageNet, which is slightly lower than EfficientNet, but still very competitive.

2.3.3. Training Configurations

We analyze the performance of both these architectures on the PlantVillage dataset by adapting pre-trained models (trained on the ImageNet dataset) using transfer learning. We just re-train the last fully connected layers of both the models to fine tune the weights so that the model can predict well on PlantVillage dataset. The key difference between transfer learning vs. training from scratch is in the initial state of weights of a few layers, which lets the transfer learning approach exploit the large amount of visual knowledge already learned by the pre-trained DenseNet and EfficientNet models extracted from ImageNet (Russakovsky et al., 2015). To summarize, we have a total of 10 experimental configurations (2 different architectures × 5 different train-test splits).

2.3.4. Hyperparameters

Each of these 10 experiments runs for a total of 10 epochs, where one epoch is defined as the number of training iterations in which the particular neural network has completed a full pass of the whole training set. The choice of the number of epochs was made based on the empirical observation that in all of these experiments, the learning always converged well within 10 epochs (as is evident from the aggregated plots (Figure 3) across all the experiments). To enable a fair comparison between the results of all the experimental configurations, we also tried to standardize the hyper-parameters across all the experiments, and we used the following hyper-parameters in all of the experiments:

- **Solver type**: Stochastic Gradient Descent,
- **Learning rate**: 0.005
- **Batch size:** 32

3. Results

The accuracy across all our arrangements is shown in Table 1. Each and every trial set-up run each for a total of 10 epochs. We alter the test set to address the over-fitting problem. To determine the best train set ratio and see that even in the worst situation of evaluating the trained models after training on only 20% of the data. On the remaining 80% of the input, the EfficientNet and DenseNet generates 96.018% and 96.011% accuracy respectively which is just 2% less than their corresponding best accuracy with a train-test split of 80:20.

Table 1. Accuracy of EfficientNet and DenseNet on five different train-test splits. The reported accuracy is of the test set.

Train:Test	Model	
	Efficientnet	DenseNet
80:20	98.886%	98.886%
60:40	98.798%	98.875
50:50	98.313%	98.317%
40:60	97.803%	97.798%
20:80	96.018%	96.011%

Also according to Fig. 3, we see that the loss gradually decreases and accuracy gradually increases while training the DenseNet model with a train-test split of 20:80 whereas the loss steeply decreases and accuracy steeply increases when the train-test split is 80:20. Similar results were also found for EfficientNet model. This shows that the model face more difficulty to learn when the number of training data were less.

4. Web App Implementation

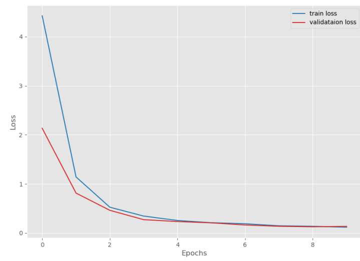
We have also implemented a web application using the Flask web framework and python. In this web application an user is given the choice to upload an image. The user can upload an image of a diseased leaf. The image can be in .png, .jpg, .jpeg formats. After uploading the image, the image will be shown in the webpage and the option to predict the image will appear. Clicking the predict button will output the plant name and the disease that is predicted by the trained model. We have hard coded the app to predict using the trained densenet model. A screenshot of the web application is shown in Fig. 4.

5. Model Results

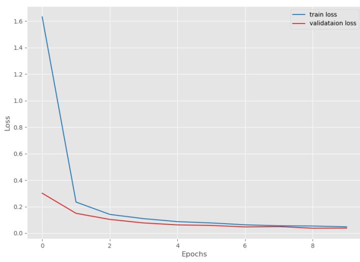
Model predicts the leaf species and predicts the leaf is healthy or unhealthy and determines the disease the leaf gets effected.Image of the results were shown in the Fig. 5

6. Conclusion

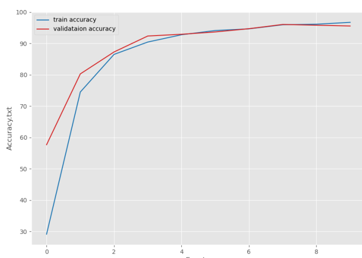
Convolutional neural networks have improved in object detection and image classification in recent years. (Krizhevsky et al., 2017; Simonyan & Zisserman, 2014; Zeiler & Fergus, 2014; He et al., 2016; Szegedy et al., 2015). Traditional image categorization methods have used hand-engineered features like SIFT (Lowe, 2004), HoG (Dalal & Triggs, 2005), SURF (Bay et al., 2008), and others, followed by a learning algorithm.Thus, predefined features greatly influ-



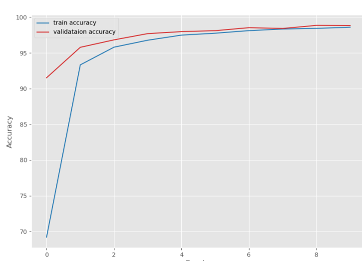
(a) Loss 20:80 split



(b) Loss 80:20 split



(c) Accuracy 20:80 split



(d) Accuracy 80:20 split

Figure 3. The loss and accuracy curve of DenseNet training with train-test splits of 20:80 and 80:20

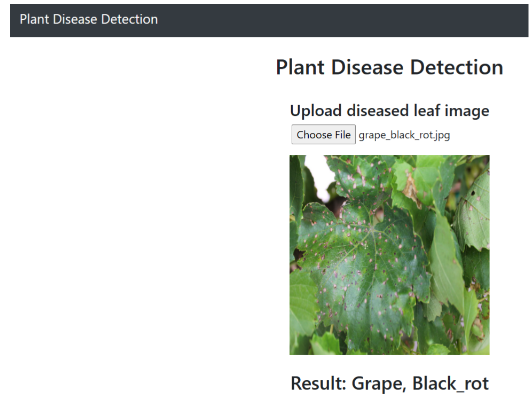


Figure 4. Screenshot of the web application to detect plant disease

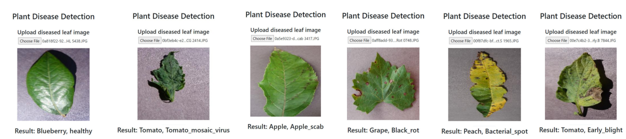


Figure 5. Results of the model

enced these approaches' performances. Feature engineering is difficult and must be repeated whenever the issue or dataset changes. All standard computer vision methods for plant disease detection use hand-engineered features, image enhancement, and other labor-intensive methods.Traditional machine learning disease classification methods focus on a few groups, usually within a crop. Examples include a feature extraction and classification pipeline using thermal and stereo images to classify tomato powdery mildew against healthy tomato leaves (Raza et al., 2015); the detection of powdery mildew in uncontrolled environments using RGB images (Hernández-Rabadán et al., 2014); the detection of apple scab using RGBD images (Chéné et al., 2012); and the detection of citrus huanglongbing using fluorescence imaging spectroscopy (Wetterich et al., 2013) reviewed machine learning for plant phenotyping. Neural networks have been used to classify and detect Phalaenopsis seedling diseases like bacterial soft rot, bacterial brown spot, and Phytophthora black rot (Huang, 2007), but the method required representing images using a carefully selected list of texture features before the neural network.classification. Our approach is based on recent work by (Krizhevsky et al., 2009), which showed for the first time that end-to-end supervised training using a deep convolutional neural network architecture is a practical possibility even for image classification problems with a very large number of classes, beating traditional approaches using hand-engineered features by a large margin in standard benchmarks. Their generalizability and lack of feature engineering make them a promising option for computational inference of plant diseases.We trained a deep convolutional neural network model on plant leaf images to identify crop species and disease on new images. The PlantVillage data set of 54,306 images, comprising 38 classes of 14 crop species and 26 diseases (or lack thereof), has achieved this goal with a top accuracy of 99.35%. The model properly classifies crop and disease from 38 possible classes in 993 of 1000 images without feature engineering. Importantly, while model training takes many hours on a high-performance GPU cluster computer, classification is fast (less than a second on a CPU) and can be done on a smartphone. This enables global smartphone-assisted crop disease detection.However, future work must resolve several limitations. First, on photos taken under different conditions than those used for training, the model's accuracy drops to just above 31%. This accuracy is much better than the one based on a random selection of 38 classes (2.6%), but a more diverse set of training data is needed to improve it. Our results suggest that more (and more variable) data alone will significantly increase accuracy, and data collection efforts are underway.We can only classify single leaves facing up on a homogeneous backdrop. A real-world application should be able to identify images of plant disease real-world application should be able to identify images of plant disease. Many diseases appear on multiple plant parts, not just the upper half of leaves. Thus, new image gathering efforts should aim to capture images from many angles and in realistic settings.As producers are expected to know which crops they are growing, using 38 classes that contain both crop species and disease status makes the task harder than necessary. Limiting the classification task to disease status won't affect the PlantVillage dataset's high accuracy. Real-world datasets show accuracy increases. With more training data, the presented approach should work better with many crop species and diseases.Finally, this method supplements current disease diagnosis solutions and does not replace them. Laboratory tests are always more accurate than visual findings, which can be difficult in the early stages. However, we think the strategy could help prevent yield loss. In the future, smartphone image data may be combined with position and time data for greater accuracy. Last but not least, remember the incredible speed at which mobile technology has evolved in recent years and will continue to do so. With the number and quality of sensors on smartphones growing, we expect highly accurate smartphone diagnoses soon.

References

Annex, V. Report of the plenary of the intergovernmental science-policy platform on biodiversity and ecosystem services on the work of its third session.

Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L. Speeded-up robust features (surf). *Computer vision and image understanding*, 110(3):346–359, 2008.

Bureau, R. International telecommunication union. *ICT Facts and Figures*, 2017.

Chéné, Y., Rousseau, D., Lucidarme, P., Bertheloot, J., Caffier, V., Morel, P., Belin, É., and Chapeau-Blondeau, F. On the use of depth camera for 3d phenotyping of entire plants. *Computers and Electronics in Agriculture*, 82:122–127, 2012.

Dalal, N. and Triggs, B. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pp. 886–893. Ieee, 2005.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Ehler, L. E. Integrated pest management (ipm): definition, historical development and implementation, and the other ipm. *Pest management science*, 62(9):787–789, 2006.

Everingham, M., Van Gool, L., Williams, C. K., Winn, J., and Zisserman, A. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88: 303–338, 2010.

Harvey, C. A., Rakotobe, Z. L., Rao, N. S., Dave, R., Razafimahatratra, H., Rabarijohn, R. H., Rajaofara, H., and MacKinnon, J. L. Extreme vulnerability of smallholder farmers to agricultural risks and climate change in madagascar. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 369(1639):20130089, 2014.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Hernández-Rabadán, D. L., Ramos-Quintana, F., and Guerrero Juk, J. Integrating soms and a bayesian classifier for segmenting diseased plants in uncontrolled environments. *The Scientific World Journal*, 2014, 2014.

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

Huang, K.-Y. Application of artificial neural network for detecting phalaenopsis seedling diseases using color and texture features. *Computers and Electronics in agriculture*, 57(1):3–11, 2007.

Hughes, D., Salathé, M., et al. An open access repository of images on plant health to enable the development of mobile disease diagnostics. *arXiv preprint arXiv:1511.08060*, 2015.

IFAD, U. Smallholders, food security and the environment. *Rome: International Fund for Agricultural Development*, 29, 2013.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

LeCun, Y., Bengio, Y., and Hinton, G. Deep learning. *nature*, 521(7553):436–444, 2015.

Lowe, D. G. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60: 91–110, 2004.

Raza, S.-e.-A., Prince, G., Clarkson, J. P., and Rajpoot, N. M. Automatic detection of diseased tomato plants using thermal and stereo visible light images. *PloS one*, 10(4):e0123262, 2015.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115: 211–252, 2015.

Sanchez, P. A. and Swaminathan, M. S. Cutting world hunger in half. *Science*, 307(5708):357–359, 2005.

Schmidhuber, J. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Strange, R. N. and Scott, P. R. Plant disease: a threat to global food security. *Annu. Rev. Phytopathol.*, 43:83–116, 2005.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

Tai, A. P., Martin, M. V., and Heald, C. L. Threat to future global food security from climate change and ozone air pollution. *Nature Climate Change*, 4(9):817–821, 2014.

Tan, M. and Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pp. 6105–6114. PMLR, 2019.

Wetterich, C. B., Kumar, R., Sankaran, S., Junior, J. B., Ehsani, R., and Marcassa, L. G. A comparative study on application of computer vision and fluorescence imaging spectroscopy for detection of citrus huanglongbing disease in usa and brazil. In *Frontiers in Optics*, pp. JW3A–26. Optica Publishing Group, 2013.

Zeiler, M. D. and Fergus, R. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*, pp. 818–833. Springer, 2014.