Turkish Journal of Agriculture and Forestry

http://journals.tubitak.gov.tr/agriculture/

Research Article

Turk J Agric For
(2021) 45: 775-783
© TÜBİTAK
doi:10.3906/tar-2010-100

Apple leaf disease detection and classification based on transfer learning

Cevher ÖZDEN[1,2,*]
[1]Department of Computer Science Engineering, Faculty of Engineering, Akdeniz University, Antalya, Turkey
[2]Department of Agronomics, Faculty of Agriculture, Çukurova University, Adana, Turkey

Received: 26.10.2020 • **Accepted/Published Online:** 27.09.2021 • **Final Version:** 16.12.2021

Abstract: The world population and the number of people affected by hunger constantly increases. Precision farming offers new solutions to a modern and more fertile production in agriculture. Early and in-place disease detection is one of the fields that recent studies have focused on. The present paper introduces a new approach to transfer learning in that training, validating and testing of the model have been made on images from different sources to see its effectiveness. Several optimization methods including the adaptation of a recent custom PowerSign optimization algorithm are compared in the study. Accordingly, the model with Adagrad optimizer produced more consistent training, validation and testing accuracies as 92%, 91% and 91%, respectively. The final model is transformed into a mobile application and tested on the field. The app showed high accuracy in the real environment on condition that the phone camera should be kept close to the leaf and focus should be clear on the image.

Key words: Precision agriculture, disease detection, deep learning, image processing

1. Introduction

The ongoing development in the area of deep learning offers new opportunities for many fields. Early recognition of crop leaf diseases is one of the hottest areas where researchers introduce more reliable and robust models. A number of studies in this area have employed image processing techniques and different structures of convolutional neural networks (CNNs) for this purpose. Rehman et al. (2020) proposed a hybrid contrast stretching method to improve the quality of apple leaf images in PlantVillage dataset. Then, they employed Mask RCNN for image segmentation and ResNet-50 pretrained architecture for classification. They compared the results with other classification methods and reported that their approach outperformed with over 99% accuracy. Sibiya and Sumbwanyambe (2021) first applied threshold-segmentation on images of diseased maize leaves in PlantVillage dataset to obtain the percentage of the diseased leaf area and partitioned images into four severity classes. They trained a VGG-16 architecture network to classify the images according to their severity classes. They reported 95.6% validation accuracy and 89% test accuracy. Afzaal et al. (2021) collected 5199 images of healthy and early blight diseased potato plants from four different fields. They employed GoogleNet, VGGNet and EfficientNet architectures, and as a result, they reported that EfficientNet yielded the best performance in the classification of early blight disease with 0.98 F-score. Kamal et al. (2019) created two versions of depthwise separable convolutional network based on MobileNet, which they called Reduced MobileNet and Modified MobileNet, respectively. They used a subset of PlantVillage dataset for performance comparison, and they reported that Reduced MobileNet attained 98.34% accuracy with 29 times fewer parameters than VGG and 6 times lesser than MobileNet. Hossain et al. (2021) proposed a custom CNN architecture consisting of 10 layers to recognize rice leaf diseases. They used a total of 323 RGB colored images of five rice leaf diseases collected by International and Bangladesh Rice Research Institutes. They applied various augmentation techniques such as rotation, flipping, shifting, scaling and zooming and increased the number of images to 3876. They reported that the model achieved 99.78% training accuracy, 97.35% validation accuracy and 97.82% accuracy on independent rice images. Radha et al. (2021) compared various machine learning methods and deep learning architectures. They used a dataset that consists of diseased and healthy citrus leaves and fruits manually collected with the help of experts from Citrus Research Center in Punjab, Pakistan. They implemented SqueezeNet, linear support vector machine, stochastic gradient descent, random forest, Inception-V3 and VGG-16. Accordingly, they reported that deep learning (DL) architectures outperformed machine learning models and VGG-16 achieved highest classification accuracy of

* Correspondence: efeozden@gmail.com

89.5%, which was followed by Inception-V3 with 89%. Saleem et al. (2019) published a comprehensive review of DL models used for the detection of various plant diseases. The authors gave a detailed information about the chronological development of pretrained architectures and visualization techniques. They also provided brief information about the studies that used the pretrained and modified deep learning architectures along with the dataset and performance metrics. Accordingly, they concluded that datasets should be designed to represent the real environment and consider different field scenarios. Saleem et al. (2020) compared some of the well-known CNN architectures on the PlantVillage dataset. They used all the images (54.306) of 14 plant species in the dataset. For image preprocessing, they only applied normalization and changed the image size to 224 × 224 × 3. Upon detecting the best performing architecture, they tried to further improve the results by using various optimizers. As a result, they reported that Xception with Adam optimizer obtained the highest validation accuracy and F1-score of 99.81% and 0.9978, respectively.

Many studies in literature have used this and derived versions of the dataset with various methods (DeChant et al., 2017; Fuentes et al., 2017; Ferentinos 2018; Wspanialy and Moussa, 2020). However, most of the models have not been turned into applications that can be tried on the real environment. And the few developed apps provided rather poor results because the images in the dataset could not represent the noisy images taken in the open field. Another important point is that most studies employed models on the validation or testing sets that belong to the very same dataset used for training and the resulting models mostly have not been tried on the new datasets or in the real environment.

This paper presents a three-step approach to the classification of apple leaf diseases by combining two different datasets. In the first step, background removal and certain augmentation techniques are applied to approximate two different imaging approaches of the datasets. Then, a pretrained model (MobileNetV2) is employed on the combined dataset with different hyperparameters and optimizers (Sandler et al., 2019). In the second step, the most promising combination is used solely for testing purposes with the Plant Pathology dataset. And in the third step, final model is converted into TFLite model and a mobile application is developed and tested in the real environment.

In the study, the PowerSign optimizer presented by Irwan et al. in late 2017 is tested. The PowerSign is a relatively new and promising optimizer that has not been able to attract much attention (Kamsing et al., 2019; Kamsing et al., 2020). The reason can be the difficulty of coding from scratch and incorporating custom optimizers into present deep learning frameworks. In this paper, the PowerSign algorithm is coded and adapted for use in TensorFlow v2.

Paper contributions:

1. Precision farming has not gained enough importance in Turkey; however, major countries in agriculture have already tested and adopted the new technological products of the deep learning era. These technologies help to increase the yield and output of agriculture. In this respect, this paper is one of the first studies that have been implemented in Turkey.

2. The paper utilizes two different datasets to observe the performance of the developed models on new data. In this way, the model used for transfer learning is trained on the images that represent the real environment conditions.

3. A new promising custom optimizer (PowerSign) is used for the first time in leaf disease classification. And its performance is compared to commonly used optimizers present in famous deep learning frameworks.

A mobile application is developed to test the performance of the final model in real-world scenarios. The mobile app works offline and does not depend on a remote server. This is the main advantage of the app as plant growing areas in many developing countries might have limited or no access to mobile network. The preliminary results verify the high accuracy of the final model; however, the downside of the model is that it obliges to hold the camera focused on leaves and its performance deteriorates slightly below 80% when the leaf loses focus or does not cover much of the screen. This indicates that despite background removal and augmentation techniques used in the study, the performance of the model still needs to be improved.

2. Materials and methods

2.1. Dataset

This study uses two different datasets that contain images with the same labels. The first one is Plant Pathology dataset, which consists of 3651 images captured in an apple orchard in US. The images were categorized into 4 classes by experts that are rust, scab, healthy and multiple diseases. The images in Plant Pathology dataset were taken at different angles, illumination and background with different shapes and sizes. This makes dataset rather complex and close to real world conditions.

The second dataset is PlantVillage dataset that has been extensively used by many previous studies on image classification. The dataset contains 54,303 leaf images of 14 different plant species which are categorized into 38 different classes, 12 healthy and 26 unhealthy (spot, rust, blight, mite, etc.). This dataset contains images of apple leaves which have the same disease attributes as the plant pathology dataset. However, it has certain discrepancies due to the rather controlled structure of photographing process. Samples from both datasets are depicted in Figure 1.

In order to eliminate the discrepancies between two datasets, all images are resized to 224 × 224 × 3 using geometric transformation without any loss in image quality. Then, iterative GrabCut algorithm in OpenCV is used to remove the background from the images. The resulting images are illustrated in Figure 2 below.

2.2. Transfer learning

Transfer learning focuses on transferring the knowledge across different domains and has found a large application area in the recent years. This method is based on the adaptation of a model trained on a large image database for a new target usage. A pretrained model either can be transferred as the input of the next task, or its weights and layers can be fine-tuned to adapt it to the new task (Gonthier et al., 2020). Many deep learning architectures have been introduced and used for this purpose. Some well-known and successful architectures include AlexNet, VGG, ResNet, DenseNet, Inception, GoogleNet, Xception, MobileNet and EfficientNet. Different versions of MobileNet and EfficientNet were considered for this study. Both models are more suitable and mostly used for mobile phone applications because of their relatively low number of parameters, so they can run with limited computational sources that a standard smart phone can offer. The parameter numbers of the pretrained architectures are given in the Table 1 below.

Another aim of this study is to develop a mobile application based on the resulting model. Therefore, the model size and inference time are other important factors in selecting the pretrained model and deep learning architecture. MobileNet V2 has smaller size when turned into TFLite model with relatively better inference time. For this reason, it is used for transfer learning in the study. A comparison between MobileNet V2 and Efficient Net Lite models is provided in the Table 2 below.

MobileNet has introduced depthwise separable convolution that significantly reduces the complexity of neural networks. The idea is based on dividing convolution operation into two separate layers: the first one performs lightweight filtering with a single filter per input channel, while the second layer performs pointwise convolution (1 × 1) and builds new features from input channels. The upgraded MobileNet V2 has introduced a novel layer: inverted residual with linear bottleneck (Sandler et al., 2019). In this layer, low dimensional representation is taken as an input, expanded to high dimension and filtered with a lightweight depthwise convolution. Then, resulting features are compressed back to a low dimension with a linear convolution. The residual block structures are illustrated in Figure 3.

Input and output layers of MobileNet V2 are pruned prior to its use for transfer learning. Then, an input layer of size (224 × 224 × 3) is added in front of MobileNet V2, also global average pooling layer, Dense Layer with Relu activation and Dense Layer with Softmax activation for four classes are included. The final model has over 5 million trainable parameters.

2.3. Deep learning optimizers

The characteristics of the optimizers used in the study can be summarized as follows:

- Adam: This optimizer combines the advantages of RMSProp and SGD optimizers by using both momentum

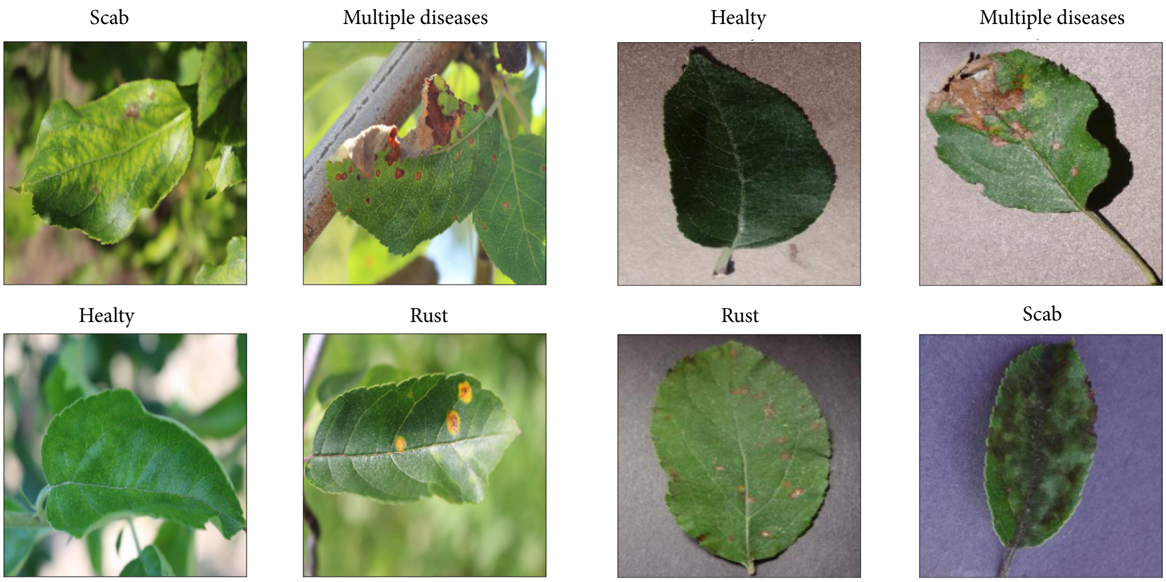

Figure 1. Sample images from Plant Pathology and plant PlantVillage datasets.

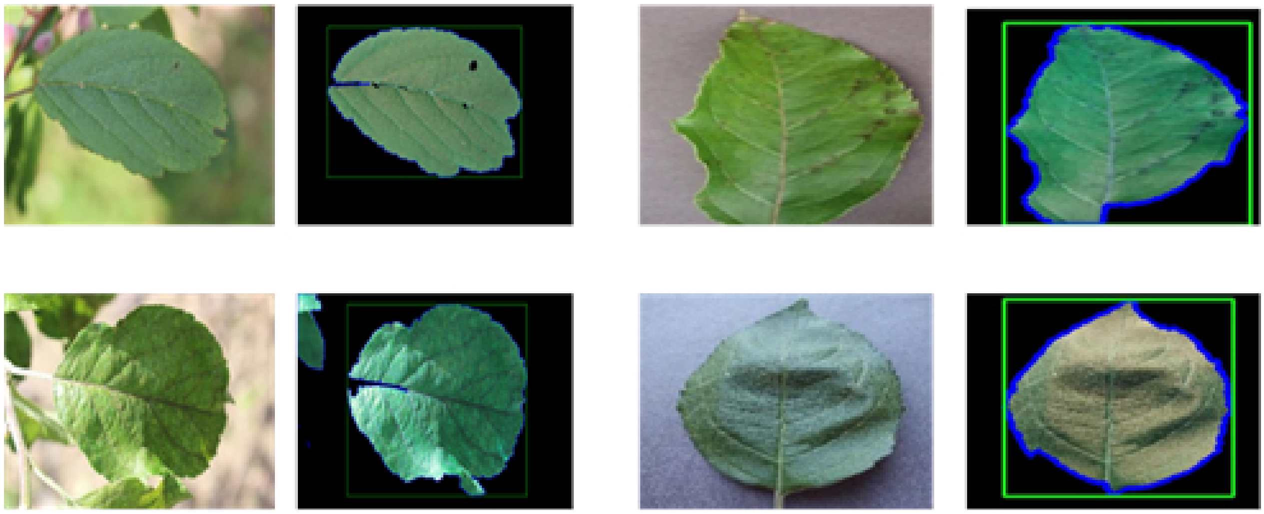

Figure 2. Background removal.

Table 1. Some popular deep learning architectures and their parameter numbers.

Deep learning models	Parameters
AlexNet	60M
VGG	133–144M
Xception	22.8M
Inception	132M
MobileNet V1	4.2M
MobileNet V2	3.4M
EfficientNet-B0	5.3M
EfficientNet-B7	66M

Table 2. Comparison of MobileNet V2 and EfficientNet Lite.

Model	Model size (MB)	Inference time (s)
MobileNet V2	8.54	0.035
EfficientNet Lite-0	12.58	0.042
EfficientNet Lite-4	44.69	0.221

and scaling. It is primarily designed for nonstationary and noisy problems (Kingma and Ba, 2014).

- Adagrad: This optimizer is primarily designed for high dimensional problems. It scales the learning rate for each dimension using the knowledge of past iterations. It lowers learning rate for more frequent features and increases it for less frequent features (Duchi et al., 2011).

- Adadelta: It is developed to address two problems of Adagrad. One problem is the constantly decaying learning rate during training so that it becomes too small after a number of iterations. The other problem is the manual selection of global learning rate. To solve these problems, Adadelta accumulates the sum of squared gradients over a limited time rather than over all time and it uses Hessian approximation to ensure that the update direction always follows the negative direction (Zeiler, 2012).

- RMSProp: It uses a moving average of the squared gradient for each weight and adjusts the weights accordingly (Hinton et al., 2012).

- PowerSign: This optimizer implements reinforcement learning to obtain a suitable operation that enables itself to reach the optimum point. For each update, this optimizer compares the sign of the gradient and running average, and then adjust the step size with respect to the agreement between these two values. The fast early convergence of PowerSign makes it an interesting optimizer to combine with others such as Adam (Irwan et al., 2017).

The specification of the optimizers is given in Table 3.

The process followed in the study is summarized in Figure 4 below.

- The images in PlantVillage and Plant Pathology datasets are resized to 224 × 224 × 3. GrabCut algorithm in OpenCV framework is used for background removal. The resulting images are randomly merged into a single database and split into 70% training, 15% validation and %15 testing.

- The images are fed into the input layer of the model Architecture. In order to eliminate the imbalanced structure of the datasets, weighted class approach is employed. Weighted class approach sets the output layer's bias to reflect the imbalanced structure of the dataset it is trained on. This approach is reported to be especially useful when overfitting is concerned due to lack of training data (Justin and Taghi, 2019). An alternative approach could be

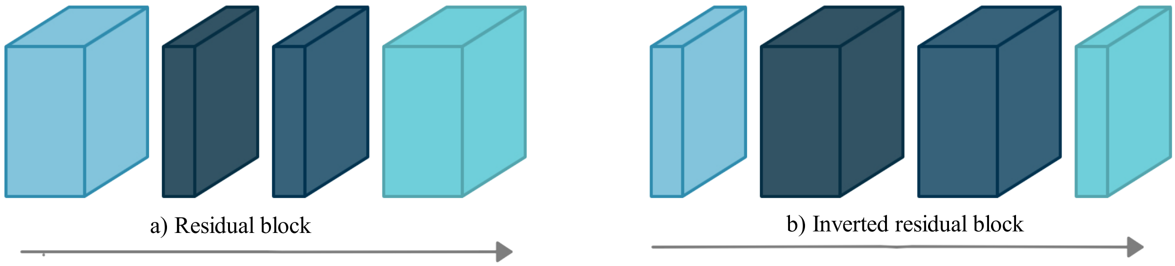

Figure 3. a) Traditional residual block, b) inverted residual block.

Table 3. Hyperparameters of the optimizers.

Optimizers	Hyperparameters
Adam	Learning rate = 0.001, beta1 = 0.9, beta2 = 0.999, epsilon = 1e-07, amsgrad = False
Adagrad	Learning rate = 0.001, initial accumulator = 0.1, epsilon = 1e-07
Adadelta	Learning rate = 0.001, rho = 0.95, epsilon = 1e-07
RMSProp	Learning rate = 0.001, rho = 0.9, momentum = 0.0, epsilon = 1e-08, centered = False
PowerSign	Learning rate = 0.001, beta = 0.9, sign decay = None, use locking = False

data augmentation; however, it is not preferred due to its additional burden on storage and computation.

- The model architecture is trained with various optimizers (Adam, Adagrad, Adadelta, PowerSign, RMSProp).
- The model that provides best accuracy is turned into mobile application using TFLite converter and Android Studio.
- The application is tested in the real environment.

3. Results

The model architecture is applied on the combined dataset with various optimizers (Adam, Adagrad, Adadelta, PowerSign, RMSProp). The validation and training accuracies are the final results after 20 epochs. Accordingly, it is noteworthy that PowerSign optimizer has attained the highest accuracy on training set and surpassed RMSProp in test accuracy, however, it overfits the data as its validation and test accuracies are lower. The results are summarized in Table 4.

The pretrained model yielded more consistent validation, training and testing accuracies with Adagrad optimization. The prediction performance of the model on test dataset is depicted as confusion matrices. One important point is that all optimizers have produced their lowest scores for the classification of multiple diseases class. This could be attributed to the vagueness of the term. Each leaf in multiple diseases class could carry different proportions of rust, scab and rot, which further complicates the classification of this class. The results on test dataset are given in Figure 5 below.

The best model was selected by F1-score and test accuracy and it was transformed into TFLite model to work with Android OS phones. One of the base templates of TensorFlow mobile application has been utilized to develop mobile application in this study. The resulting app was tested on PlantVillage test dataset as well as the images downloaded from the internet and taken in an apple orchard in Antalya, Turkey. The preliminary results indicated that the mobile app makes highly accurate classification for healthy, rust and scab classes, however, it produces poor results for multiple diseases class, classifying them either scab or rust. One other important point is that the camera should be kept close to the leaf and focus should be clear on the image. Otherwise, the classification accuracy of the model endures high degradation. Example screenshots of the application is provided in the Figure 6.

A recent study by Ngugi et al. (2020) has proposed a new automatic background removal method for mobile phone applications as an alternative to GrabCut algorithm, which has reportedly outperformed all competitor background removal techniques. It has not been employed in this paper because their method is primarily intended for web-based and centralized applications that require network condition. However, it should be incorporated and tested in a further study.

4. Discussion and conclusion

This paper has presented several novelties in image classification. The pretrained models yield high accuracies

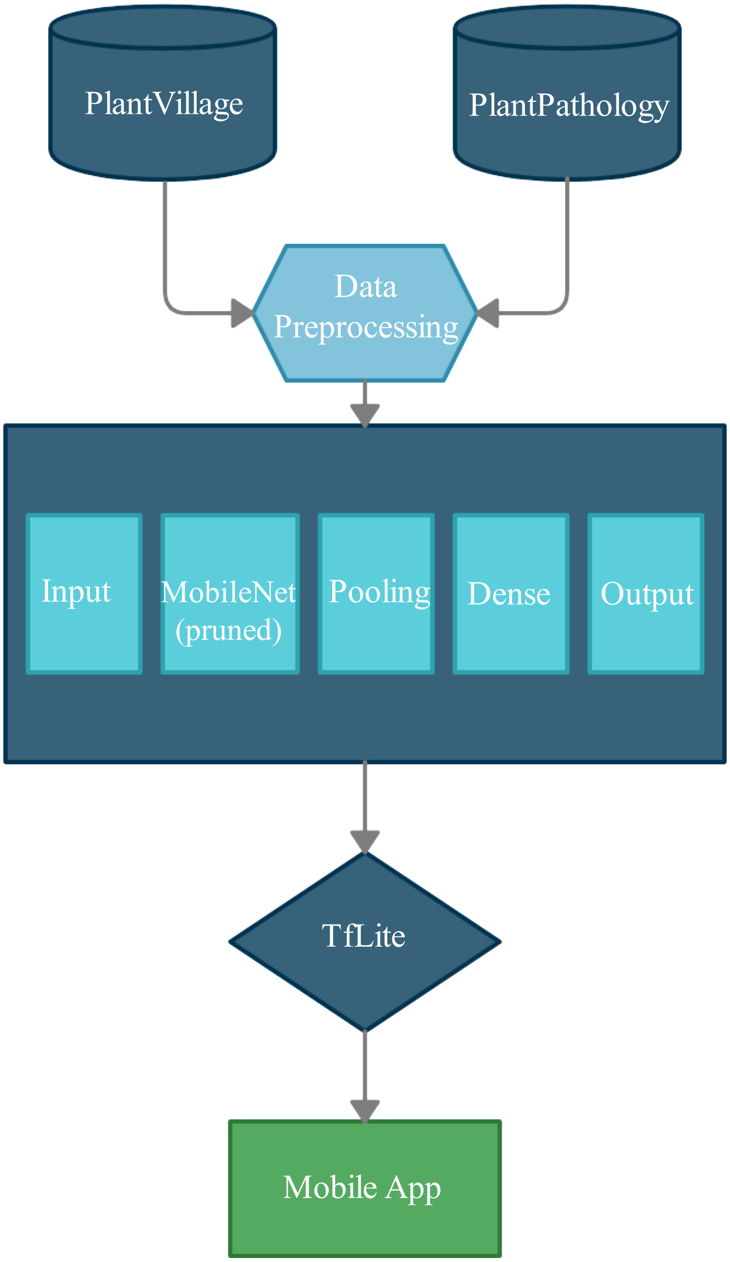

Figure 4. Block diagram of the process steps.

Table 4. Summary results of model.

Optimizer	Training accuracy	Validation accuracy	Test accuracy	F1-score
Adam	0.97	0.88	0.87	0.86
Adagrad	0.92	0.92	0.91	0.91
PowerSign	0.98	0.85	0.82	0.83
Adadelta	0.92	0.90	0.88	0.88
RMSProp	0.96	0.75	0.71	0.69

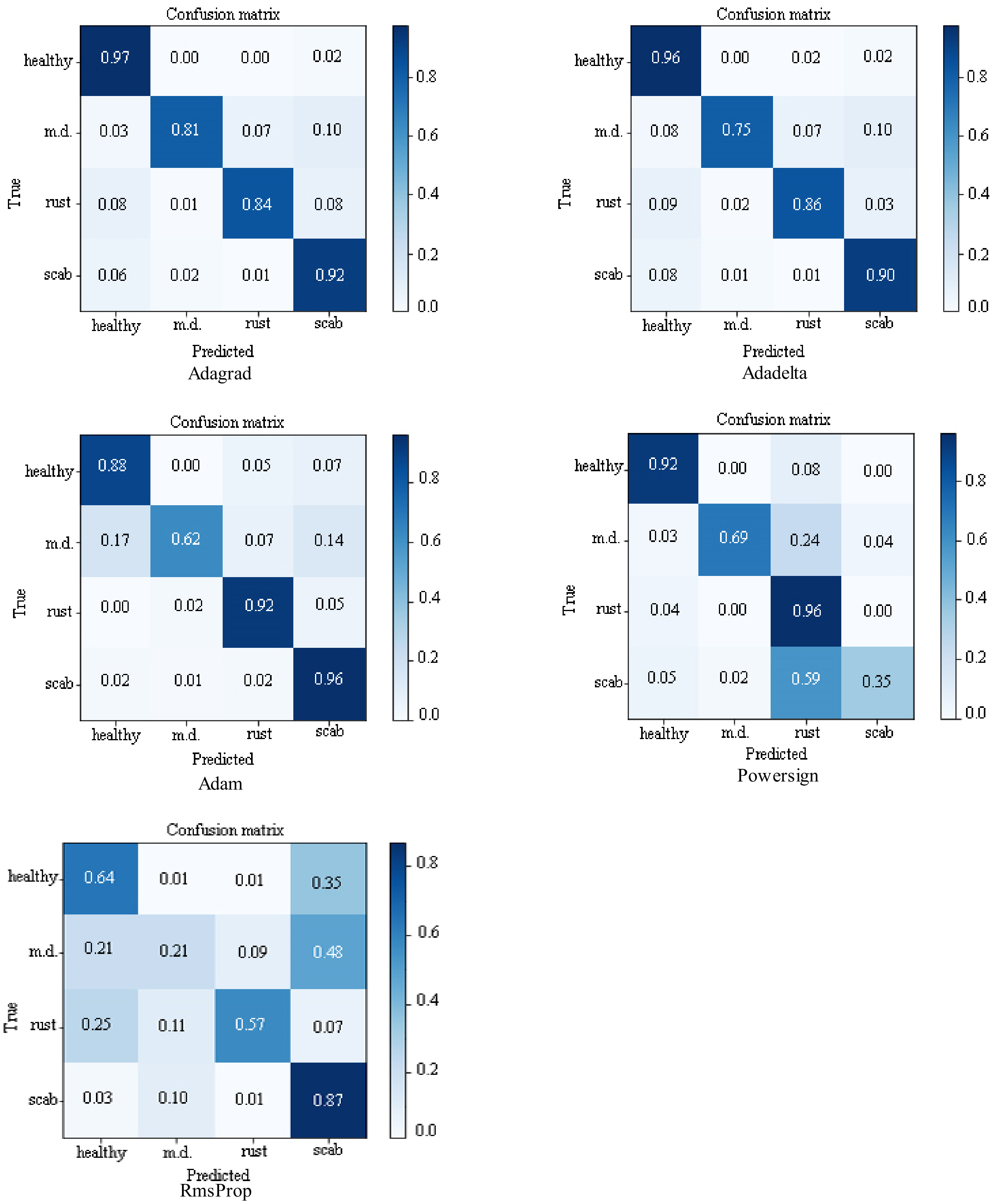

Figure 5. Confusion matrices on test dataset.

in image classification if the images belong to the same dataset, in other words, if the images are collected with the same conditions. Furthermore, the pretrained models are trained on images from thousands of different and unrelated fields. However, mobile applications are intended for open production fields with different conditions and

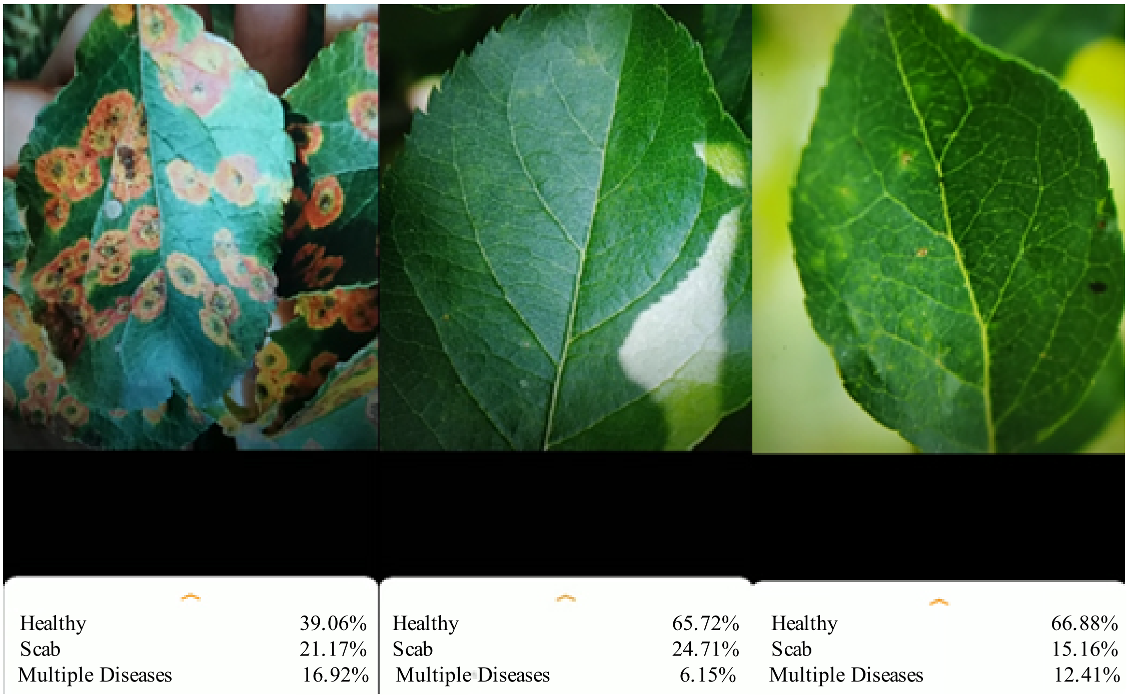

Figure 6. Screenshots of the mobile app.

they will be used by different users. Therefore, the models to be used in transfer learning should be trained on the images from the same field. For this purpose, two similar datasets are combined in the paper. And the developed model is tested on images taken from different sources. The final mobile app has certain advantages in that it does not need network connection or a centralized processor to run and it produces high accuracies. The downside of the application is that it obliges users to hold the camera in a certain position to decrease the interference of surrounding environment. Another important contribution of the paper is that a relatively new custom PowerSign optimizer has been tested on TensorFlow V2 and it attained certain success especially on training dataset. However, it rapidly overfits the data. This paper adopted class weight approach to overcome imbalanced structure of the dataset. The PowerSign optimizer might as well be tried on oversampled data to see how its performance changes and certain amendments can be added to prevent it from memorizing the dataset.

References

Afzaal H, Aitazaz F, Schumann A, Nazar H, McKenzie-Gopsill A et al. (2021). Detection of a potato disease (early blight) using artificial intelligence. Remote Sensing. doi: 13.411.10.3390/rs13030411

DeChant C, Wiesner-Hanks T, Chen S, Stewart EL, Yosinski J et al. (2017). Automated identification of northern leaf blight-infected maize plants from field imagery using deep learning. Phytopathology. 107 (11): 1426-1432. doi: 10.1094/PHYTO-11-16-0417-R

Duchi J, Hazan E, Singer Y (2011). Adaptive subgradient methods for online learning and stochastic optimization. Journal of Machine Learning Research. 12 (61): 2121-2159.

Ferentinos KP (2018). Deep learning models for plant disease detection and diagnosis. Computers and Electronics in Agriculture 145: 311-318.

Fuentes A, Yoon S, Kim SC, Park DS (2017). A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. Sensors 17 (9): 2022. doi: 10.3390/s17092022.

Gonthier N, Gousseau Y, Ladjal S (2021). An analysis of the transfer learning of convolutional neural networks for artistic images. Lecture Notes in Computer Science, 12663. Springer. doi: 0.1007/978-3-030-68796-0_39

Hassan A, Aitazaz F, Arnold S, Nazar H, Andrew M (2021). Detection of a potato disease (early blight) using artificial intelligence. Remote Sensing. 13.

Hinton G, Srivastava N, Swersky K (2012). Neural networks for machine learning course. Slide 29 of lecture 6.

Hossain S, Tanjil M, Mohammed A, Mohammad I, Mobassirin S (2021). Rice leaf diseases recognition using convolutional neural networks. doi: 10.1007/978-3-030-65390-3_23

Hughes D, Marcel S (2015). An open access repository of images on plant health to enable the development of mobile disease diagnostics through machine learning and crowdsourcing.

Irwan B, Barret Z, Vijay V, Quoc L (2017). Neural optimizer search with reinforcement learning.

Johnson J, Khoshgoftaar T (2019). Survey on deep learning with class imbalance. Journal of Big Data. doi: 10.1186/s40537-019-0192-5.

Justin J, Taghi K (2020). The effects of data sampling with deep learning and highly imbalanced big data. Information Systems Frontiers, 22. doi: 10.1007/s10796-020-10022-7.

Kamal K, Zhendong Y, Mingyang W, Zhilu W (2019). Depthwise separable convolution architectures for plant disease classification. Computers and Electronics in Agriculture, 165. doi: 10.1016/j.compag.2019.104948

Kamsing P, Torteeka P, Soemsak Y (2020). An enhanced learning algorithm with a particle filter-based gradient descent optimizer method. Neural Computing and Applications, 32. doi: 10.1007/s00521-020-04726-9

Kamsing P, Torteeka P, Yooyen S (2019). Deep convolutional neural networks for plane identification on Satellite imagery by exploiting transfer learning with a different optimizer. IGARSS 2019 – 2019 IEEE International Geoscience and Remote Sensing Symposium: 9788-9791. Japan. 2019. doi: 10.1109/IGARSS.2019.8899206

Kingma D, Ba J (2014). Adam: A method for stochastic optimization. International Conference on Learning Representations.

Ngugi L, Moataz A, Abo-Zahhad M (2020). Tomato leaf segmentation algorithms for mobile phone applications using deep learning. Computers and Electronics in Agriculture 178: 105788. doi:10.1016/j.compag.2020.105788

Radha S, Jyotir C, Noor Z, Sarfraz B (2021). Performance of deep learning vs machine learning in plant leaf disease detection. Microprocessors and Microsystems 80: 103615. doi: 10.1016/j.micpro.2020.103615

Rehman Z, Khan MA, Ahmed F (2021). Recognizing apple leaf diseases using a novel parallel real-time processing framework based on MASK RCNN and transfer learning: An application for smart agriculture. IET Image Processing 2021: 1– 12. doi: 10.1049/ipr2.12183

Saleem M, Potgieter J, Arif K (2019). Plant disease detection and classification by deep learning. Plants 2019, *8*, 468. doi: 10.3390/plants8110468

Saleem M, Potgieter J, Arif K (2020). Plant disease classification: A comparative evaluation of convolutional neural networks and deep learning optimizers. Plants (9): 1319. doi: 10.3390/plants9101319

Sandler M, Howard A, Menglong Z, Andrey Z, Liang-Chieh C (2019). MobileNetV2: Inverted residuals and linear bottlenecks. 4510-4520. doi: 10.1109/CVPR.2018.00474

Sibiya M, Sumbwanyambe M (2021). Automatic fuzzy logic-based maize common rust disease severity predictions with thresholding and deep learning. Pathogens. doi: 10.131.10.3390/pathogens10020131

Stevo B, Fulgosi A (1976). The influence of pattern similarity and transfer learning upon the training of a base perceptron B2. Proceedings of Symposium Informatica 3-121-5, Bled.

Wspanialy P, Moussa M (2020). A detection and severity estimation system for generic diseases of tomato greenhouse plants. Computers and Electronics in Agriculture 178: 105701. doi: 10.1016/j.compag.2020.105701

Zeiler M (2012). Adadelta: An adaptive learning rate method, 1212.