Analysis of transfer learning for deep neural network based plant classification models

Article *in* Computers and Electronics in Agriculture · March 2019

DOI: 10.1016/j.compag.2019.01.041

6 authors, including:

Cagatay Catal
Bahçeşehir University

Temuçin Huseyin
Gazi University

Analysis of transfer learning for deep neural network based plant classification models

Aydin Kaya[a,*], Ali Seydi Keceli[a], Cagatay Catal[b], Hamdi Yalin Yalic[a], Huseyin Temucin[a], Bedir Tekinerdogan[b]

[a] *Department of Computer Engineering, Hacettepe University, Ankara, Turkey*
[b] *Information Technology Group, Wageningen University, Wageningen, The Netherlands*

Abstract

Plant species classification is crucial for biodiversity protection and conservation. Manual classification is time-consuming, expensive, and requires experienced experts who are often limited available. To cope with these issues, various machine learning algorithms have been proposed to support the automated classification of plant species. Among these machine learning algorithms, Deep Neural Networks (DNNs) have been applied to different data sets. DNNs have been however often applied in isolation and no effort has been made to reuse and transfer the knowledge of different applications of DNNs. Transfer learning in the context of machine learning implies the usage of the results of multiple applications of DNNs. In this article, the results of the effect of four different transfer learning models for deep neural network-based plant classification is investigated on four public datasets. Our experimental study demonstrates that transfer learning can provide important benefits for automated plant identification and can improve low-performance plant classification models.

Keywords: Plant classification, transfer learning, deep neural networks, fine-tuning, convolutional neural networks

1. Introduction

Recent studies estimate that there are currently between 220,000 and 420,000 flowering plant species on Earth [1, 2, 3]. Although, many species are under threat of extinction, or have already been extinct due to pollution and natural disasters, still many species are waiting to be discovered. For supporting the research on biodiversity conservation and ecological monitoring, accurate identification of plants is necessary in, for example, endangered species monitoring, assessment of weed control actions, and analysis of species distribution under climate change [4]. Besides of their importance for nature and ecological balance, many plant species are raw material for the medicine and chemical industries. Hence, there is a continuing need for proper plant identification.

The objective of plant classification systems is to help non-expert and non-botanist users to identify the plants automatically. Otherwise, this identification process is too long, time consuming, and expensive. Speeding up this process provides several benefits such as low cost, less effort, and more time allocation for the other tasks. Plant recognition systems can be also used in making intelligent field guides, educational tools, and agricultural practices and forestry automation. Besides of the agricultural domain, the proposed technology can be applied to other domains including healthcare, retail, and automotive.

The identification of plants is considered as a process which assigns a plant under study to a taxon based on two kinds of features, namely quantitative and qualitative characters. The

*Corresponding author
Email addresses: aliseydi@cs.hacettepe.edu.tr (Ali Seydi Keceli), aydinkaya@cs.hacettepe.edu.tr (Ali Seydi Keceli), cagatay.catal@wur.nl (Cagatay Catal), yalinyalic@cs.hacettepe.edu.tr (Hamdi Yalin Yalic), htemucin@cs.hacettepe.edu.tr (Huseyin Temucin), bedir.tekinerdogan@wur.nl (Bedir Tekinerdogan)

flower length and plant height are examples of quantitative features, while the flower color and leaf shape are examples of qualitative features [4]. Although there are many similarities between individuals of the same species, the exact identification of plants is certainly not trivial and requires sophisticated knowledge. Thus, experts in plant species identification need to be consulted, but they are however costly and not always available. Moreover, the number of plant classification experts seem be also decreasing [5].

Gaston and O'Neill [6] indicated 15 years ago that automated species identification systems was not the norm at that time [4] due to the difficulty of the implementation, the high labor intensiveness, and the high cost. Yet, with the recent dramatic advancements in Machine Learning (ML), Cloud Computing, Internet of Things (IoT), and Distributed Computing this situation has definitely changed, and automated identification systems have now become feasible and are also more and more applied.

The automated identification of plants in several domains can now replace manual processing of huge plant species catalogs by subject matter experts [7] and thus provides several benefits including reduction of cost and time, but also more accurate identification. The success of automated plant identification has led to its broad applications in biological taxonomy studies, definition of industrial raw materials, implementation of automated systems for the precision agriculture, detection of diseased plants, and educational purposes. With the help of plant identification systems in precision agriculture, for example, it is possible to spray herbicides only on weeds [8], which make the agriculture more cost-effective and eco-friendly. Food safety, remote-sensing for farming, and deforestation control are some of the other application areas which can get benefit from the automated plant classification approaches.

To realize automated classification of plant species, various machine learning algorithms have been proposed. Among these machine learning algorithms, Deep Neural Networks (DNNs) have been applied to different data sets. DNNs have been however often applied in isolation and no systematic effort has been made to reuse and transfer the knowledge of different applications of DNNs. Transfer learning in the context of machine learning implies the reuse of the earlier acquired knowledge in similar tasks. In principle, transfer learning can be applied for any machine learning algorithm in any application domain. Yet, although several studies have focused on the use of deep learning approaches for the plant classification problem [7, 9, 10, 11, 12], there is not a comprehensive study yet, which evaluates several transfer learning scenarios for deep learning algorithms. Hence, we focus on the following research questions (RQs):

- RQ1: To what extent can plant classification benefit from transfer learning in deep learning?

- RQ2: How do the different transfer learning scenarios perform at improving the performance of plant classification models using deep learning?

To answer these questions, we designed and implemented five classification models including the baseline model (end-to-end CNN model) by applying four transfer learning strategies on deep learning-based models. The first model (baseline approach) is an end-to-end Convolutional Neural Networks (CNN) model which were tested on the four datasets separately. As the second model (cross-dataset fine tuning), we trained the same algorithm with three datasets and fine-tuned the model on the remaining one. For the third model (fine tuning), we fine-tuned the pre-trained CNN models which were previously trained with ImageNet database. As the fourth model, features were extracted from pre-trained networks and then, two traditional classification algorithms were applied based on these features. Finally, a combination of RNN (Recurrent Neural Network) and CNN algorithms is tested for the classification of plants. In this paper we analyze and compare these different approaches and provide the lessons learned.

Section 2 introduces the related work, section 3 provides the transfer learning concept, section 4 presents the adopted approach, section 5 indicates the experimental results, section 6 discusses the findings and the potential threats to validity. Section 7 explains the conclusion.

2. Related Work

The use of deep learning algorithms in machine vision problems is becoming more and more popular. The plant and the leaf identification have been previously studied with different techniques by various researchers.

The initial approaches on these problems used the color information to distinguish the plant from the soil [13, 14]. Some studies utilized the vein morphology [15, 16]. The leaf veins include many textural and shape properties which can be useful for the vision-based plant identification. Some leaf-based studies also used the shape information [17, 18]. Several studies combined the shape and texture information extracted from the leaf [19] while other studies utilized the color and texture information [20]. A method is proposed by Larese et al. [21] in which computer vision techniques are used to extract the morphological leaf vein features. Extracted features are later used with machine learning algorithms to predict the three different plant species. These researchers improved this method later on in a new research project [22].

Spectroscopic methods are also applied in plant genre classification. The reflectances of the infrared, multispectral, and visible bands are used for the feature extraction in these methods [23, 24, 25]. Muthevi and Uppu [26] utilized Local Binary Patterns (LBP) in their study. Different types of LBP extraction methods, Signed component of CLBP (SCLBP)) were applied for the feature extraction. Murat et al. [27] combined different shape descriptors for the leaf classification of the tropical shrub species. The proposed method was tested with Flavia and Swedish Leaf datasets. Yousefi et al. [28] used view rotation invariant features which are extracted from the Fast Fourier Transform and Discrete Wavelet Transform. Yu et al. [29] developed a method utilizing the leaf contour and venation. Horaisova and Kukal [30] proposed a method for the invariant leaf detection.

The use of deep learning in plant and leaf identification is a relatively new approach. Grinblat et al. [7] proposed a CNN model to work with leaf veins. Leaf veins are extracted as a binary mask and an end-to-end CNN model is trained with these masks. Ferreira et al. [31] combined the superpixel segmentation algorithm and CNN to build a weed segmentation system. CNNs are used in weed detection on images which are segmented with the superpixel algorithm. Barre et al. [32] proposed a CNN architecture for the leaf identification. They applied the Foliage, LeafSnap, and Flavia datasets and built a CNN model for the classification. An architecture similar to well-known CNN models like AlexNet and CifarNet was proposed in their study. Jeon and Rhee [33] used the transfer learning from pre-trained CNN models. Pre-trained GoogleNet was used for the feature extraction. Leaf images in different scales were given to GoogleNet as input and the activation values of different layers were stored as features.

According to the literature survey outlined in this section, there is no study which evaluates the effect of several transfer learning scenarios for plant classification models in detail. Hence the work that we present in this article can be considered complementary to the earlier studies.

The classification accuracy comparison of some related studies with the best result in our experiments are shown in Table 1. In Table 1., datasets are represented in columns and methods are represented in rows. Although different experimentation setups are used in these studies like k-fold cross validation and test-train data splitting, the best results obtained are shown in Table 1. The results obtained from our experiments are promising and comparable with the recent deep based studies. The deep learning based studies[9, 11, 33] have a superiority in classification accuracy compared to traditional ones [27, 28, 34]. Usage of transfer learning and CNN in visual recognition tasks gives better results and we can observe this from our results. If a comparison is made between deep approaches, transfer learning with a pre-trained model methods are more successful than end-to-end CNN approaches. The reason of this is need of large data to obtain an accurate CNN model. Most of the related studies are tested with one or two datasets. We applied different datasets with different sizes to observe effectiveness of different transfer and deep learning approaches. In traditional methods color, shape and textural features are extracted manually and combined

Table 1: Related Studies Summary

	Datasets	Best Results	Method
Our Study	Flavia	99.00	DF - VGG16 / LDA
	Swedish	98.80	CNN - RNN
	UCI Leaf	96.20	DF - Alexnet / LDA
	Plantvillage	99.80	FT - VGG16
Lee et al. [9]	Flavia	99.40	CNN, Fine-Tuning
	MK	97.47	
Jeon & Rhee [33]	Flavia	99.60	CNN
Caglayan et al. [34]	Flavia	96.30	Hand Crafted Shape & Color Features
Yalcin et al. [11]	TARBIL	97.47	CNN, SVM
Murat et al. [27]	Flavia	95.25	HOG, Moments, ANN, RF, SVM
Yousef et al. [28]	Swedish	99.89	
	Flavia	97.50	Fourier & Wavelet Descriptors, MLP

afterwards. By using a CNN all these features are extracted and weighted. A more compact and descriptive feature set is obtained. All of these datasets consist of cropped leaf images. So all of these methods and ours are unable to work with wild (unedited and segmented) leaf, plant images. This problem can be solved with a two layer approach. Detection can be made in first layer with an R-CNN model. Then a CNN or other kind of classifier can be applied for specie recognition

3. Transfer Learning

Machine learning models are mostly built to work in isolation, which need to be rebuilt when the features and data change. However, very often the previously acquired knowledge in machine learning can be applied for similar tasks. Instead of rebuilding the models which usually requires lots of effort, transfer learning aims to reuse the model and acquired knowledge, and likewise to decrease the model development time dramatically, and improve the model performance of the isolated learning model. Transfer learning has been used in many applications such as software defect prediction [35], sentiment classification [36], and activity recognition [37].

3.1. Transfer Learning Approaches based on Domains

Pan and Yang [38] published a review paper on transfer learning, introducing four transfer learning approaches, including:

1. Instance-transfer: Re-weighting the labeled data for the target domain
2. Feature-representation-transfer: Selecting a good feature set to reduce the difference between two domains
3. Parameter transfer: Discovering parameters in one domain and reusing these parameters in the target domain
4. Relational-knowledge-transfer: Mapping of knowledge between two domains

With respect to the nature of the target task Pan and Yang [38] distinguish the following three settings:

1. Inductive transfer learning: target task is different than the source one.
2. Transductive transfer learning: target task and source task are the same, but the domains are different.
3. Unsupervised transfer learning: target task is different than the source task, but they are related to each other.

3.2. Transfer Learning Approaches based on Feature Spaces

Weiss et al. [39] published a more recent survey paper on transfer learning in which they indicate that over 700 papers were published after 2010 on transfer learning. They categorize the recent existing approaches into three categories:

1. Homogeneous transfer learning: Transferring knowledge across similar feature spaces. For instance, in software defect classification problem, when the source and target datasets consist of the same set of software metrics, homogeneous transfer learning can be applied in this case because features of the two domains are the same.
   (a) Instance-based: In this category of techniques, instances are re-weighted to reduce the effect of misleading source data.
   (b) Feature-based (symmetric or asymmetric): In symmetric feature-based approach, features of source and target data are transformed into a common space. In asymmetric feature-based approach, features are transformed into the target domain.

   (c) Parameter-based: Target model parameters are learned based on the source model.
   (d) Relational-based: Source and target domains are investigated through a relational pattern.
   (e) Hybrid-based: The above-mentioned approaches might be integrated to build a hybrid approach such as the integration of instance and parameter-based techniques.

2. Heterogeneous transfer learning: Transferring knowledge across different feature spaces. For instance, in software defect classification problem, when software metrics of the source domain are different than the software metrics of the target domain, heterogeneous transfer learning approaches are needed. For this purpose, Nam et al. [35] used Kolmogorov-Smirnov test to match the features from the source to the features of the target domain based on the closeness of the distribution between two domains. A recent survey paper on heterogeneous transfer learning investigates 38 methods and discusses those approaches in two categories as follows [40]:
   (a) Symmetric transformation: Source and target are transformed into a common feature set called domain-invariant feature subspace.
   (b) Asymmetric transformation: Features of the source are mapped into the target feature space.

3. Negative transfer: If the information from the source domain impacts the target learner adversely, that portion of the dataset should not be used for the target domain. Negative transfer techniques help to select the best information.

3.3. Transfer Learning From Pre-Trained Models

For deep neural networks, in some cases there may not be enough data to train the network or creating the labeled data might be expensive. Hence, transfer learning can be applied to adopt the knowledge that has been learned in earlier settings. For example, there are various CNN models such as AlexNet [41],GoogleNet [42], and VGG [43], which can be used later on for similar tasks.

The two common transfer learning strategies in deep learning are deep feature extraction and fine-tuning. In the deep feature extraction, the input data is provided to the pre-trained network and activation values of various layers are stored and used as features. In fine-tuning, deep neural network is trained for a similar task, in which labeling is relatively easier. While the first layers of the pre-trained network can be fixed, fine-tuning can be done on the final layers of the model to learn the properties of the new dataset. The pre-trained model is re-trained with the new small dataset and weight values of the model are updated according to a new task. Fine-tuning process occurs on the network using back-propagation with labels.

Learning to transmit is often faster than training a new neural network because all the parameters in the new network are not estimated from scratch. In the lower layers of the network, more general features exist such as color blobs and Gabor filters and they can be transferred to other tasks as well. However, in higher layers, features are more task-specific. Deep learning systems provide high performance for several problems, but they require huge amount of data and time for their training. In this case, reusing these pre-trained models for similar tasks is quite helpful.

4. Methodology

In this article, we address the two common strategies (deep feature extraction and fine-tuning) of transfer learning for deep learning based plant classification models. In our scenarios, the visual recognition is the domain for the source and the target. Therefore, our experiments apply homogeneous transfer learning methods based on the categorization of Weiss et al. [39]. Based on the Pan and Yang [38] survey paper, our setting is inductive transfer learning because source and target tasks are different. While the target task is the identification of plants, the source task is the identification of different objects.

We present five Deep Neural Networks for the plant classification problem. These models are shown in Figure 1. First we develop a CNN from scratch. Subsequently, we show the experimentation for the four different transfer learning models which are based on fine tuning and deep feature extraction. These include fine tuning, the cross-dataset fine-tuning which is applied over the trained CNN and the third approach is the fine-tuning of CNN models on leaf datasets. In the fourth approach, pre-trained CNN models are used for feature extraction and classical machine learning algorithms with the extracted deep features are applied for the classification. The fifth approach is the combination of RNN and CNN algorithms, which is explained in later sub sections.

With respect to the classification in the previous section the Fine Tuning, and Cross-Dataset Fine Tuning models are parameter-based transfer learning approaches. The Deep Feature Learning, and CNN-RNN Classification are feature-based transfer learning approaches.

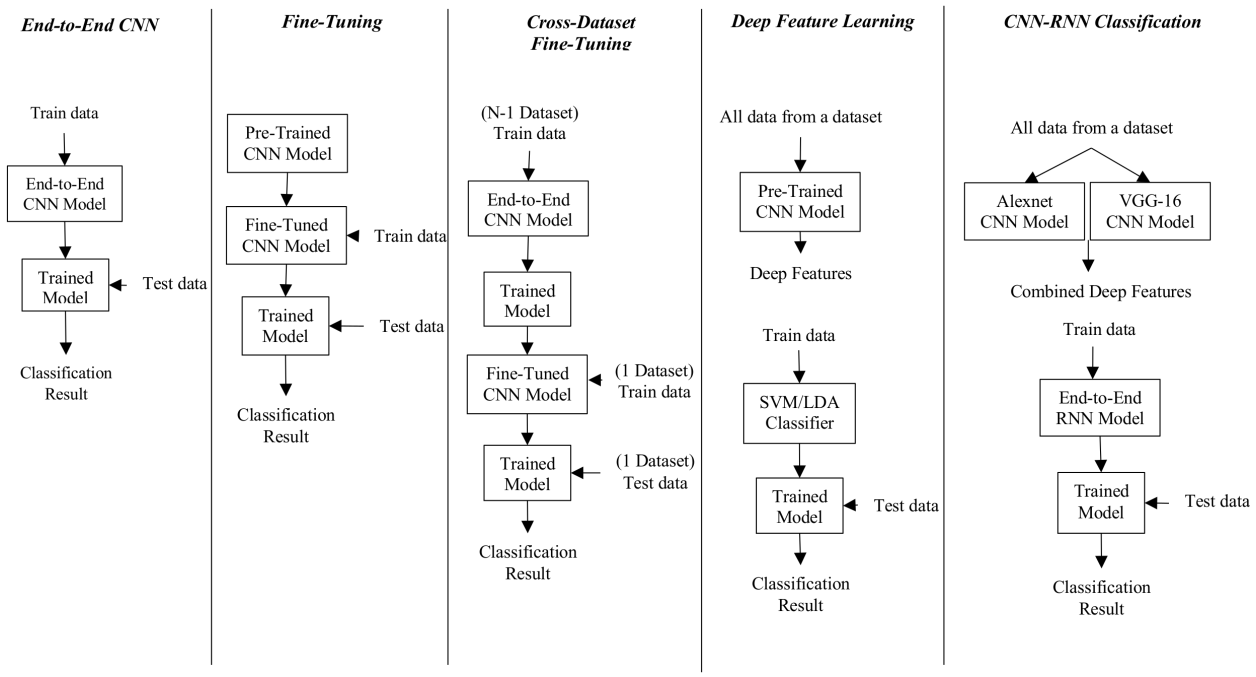

Figure 1: General schema for experimental studies

4.1. End-to-End CNN Model

An end-to-end CNN model is trained on benchmark datasets. The architecture of the designed CNN model is given in Figure 2. There are 15 layers in this CNN model. The first layer is the image input layer. Images with the size of 100x100 are provided as input. The leaf images with different width and height are re-sized before given to the CNN algorithm. There are three convolution layers in our network. Convolution layers are the main layers of a CNN. In these layers, there are filters to learn different feature types. Each filter is slid over the input images and the convolution is applied. The computed results of the convolution operations are mapped as the output. The following layers after a convolution layer are batch normalization and ReLu layers. Batch normalization layers are used for adjusting and normalizing the activation values computed by the previous layers. ReLu layers perform the threshold operation on the input to eliminate the effect of dark and noisy regions. The duty of max-pooling layers is to reduce the input dimensions to lower the computational complexity. This process is done by applying a mathematical MAX operation over the values, which corresponds to the filter. The fully connected layers are the final layers of a CNN model. These layers can be considered as the layers of standard neural networks. The class values for a given input are computed in these layers. Activation values of these layers correspond to a different layer of abstractions. Top end layers are the softmax and classification layers. Softmax layer applies softmax function and the classification layer selects the label with the maximum possibility as its output.

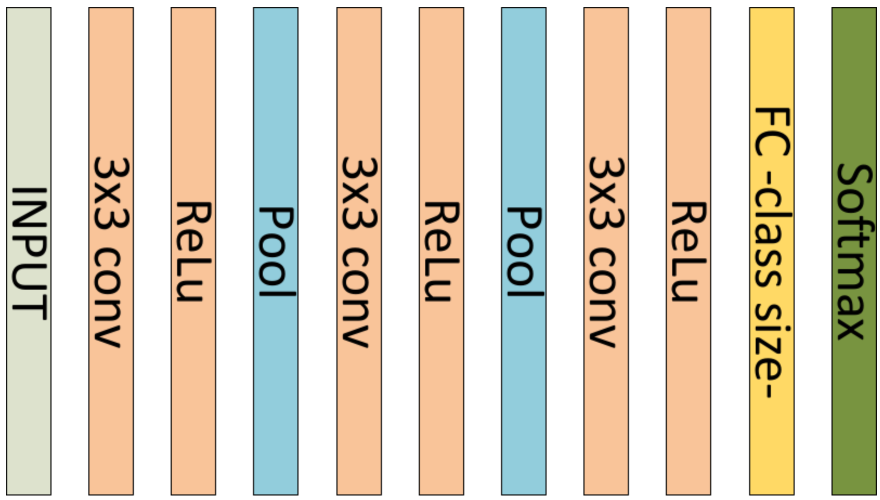

Figure 2: The architecture of the proposed CNN model

4.2. Cross-Dataset Fine Tuning

In this model, the CNN model built from scratch is trained with a combination of datasets. During this process, one of the datasets is excluded from the training sets and used for testing. All the datasets are used in testing once. Fine tuning is a widely used concept in transfer learning. In transfer learning, a model trained for a purposed task is used for a second task. In this approach, we train CNN models by using three of our four datasets and then, fine-tune the trained model with the excluded dataset. The fine-tuning operation is made by removing the last three layers and replacing with the new ones. These layers are fully connected, softmax, and classification layers. The number of outputs of the fully-connected layer should be equal to the number of the class count in the training dataset. The softmax and the classification layers are the same as the previous ones. The CNN model with new layers is trained and updated according to the new dataset.

4.3. Fine Tuning

If there is not a huge amount of training data in a problem domain, training a Convolutional Neural Network from scratch is not efficient. For this kind of cases, the common method is to use a pre-trained model obtained from a large dataset for similar problems [44, 45]. This time we apply fine-tuning for VGG-16 and AlexNet models trained with a large scale dataset. These models are well-known CNN models trained with ImageNet image database. The architectures of these models are given in Figure 3 and Figure 4. With the use of these models, we transfer the knowledge obtained from a large dataset to our own problem domain. The fine-tuning approach is similar to the method applied in cross-dataset fine tuning. Both VGG16 and AlexNet have an output of 1000 classes. Last three layers of these models are removed and replaced with the layers suitable for leaf datasets. The last fully connected , softmax, and classification layers are removed. A new fully connected layer that has the same number of outputs with the number of classes in the new training dataset is placed. Options for the new fully connected layer are set according to the new data. Finally, the new network structure is trained with the new training datasets that contain leaf images. AlexNet and VGG-16 are fine-tuned with datasets separately.

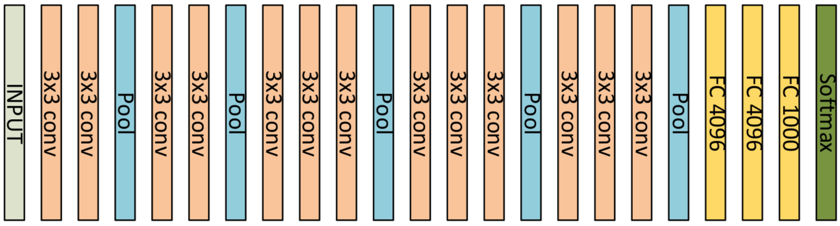

Figure 3: The architecture of VGG16 network

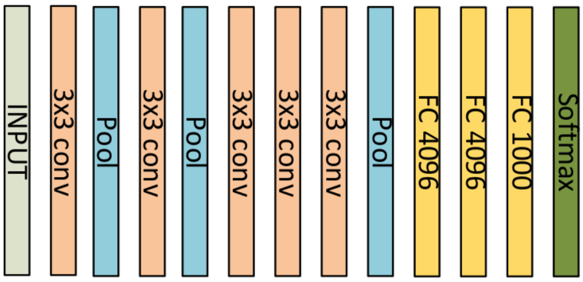

Figure 4: Architecture of the AlexNet network

4.4. Deep Feature Extraction

Another approach is extracting deep features from pre-trained models. In this approach, input images are given to VGG-16 and AlexNet directly. Then, activation values of the last fully connected layers of these pre-trained CNNs are obtained (fully connected layer 8 (Fc8)). Input layers of the AlexNet and VGG-16 are fixed and therefore, all the input data is re-sized to 227x227 for AlexNet and 224x224 for VGG-16. 1000 features are gathered from the Fc8 layer. After the deep feature extraction, classical machine learning algorithms are applied for the classification model development. Linear Discriminant Analysis(LDA) [46] and linear kernel SVM [47] are used for the training of models because they were previously applied in this problem successfully.

4.5. CNN-RNN

The final approach is to extract deep features from the pre-trained models and combine them with an RNN (Recurrent Neural Network) model. Although RNNs perform well for the sequential data, there are many studies which show that they can also be used for the non-sequential data [48]. In this approach, input images are given to VGG-16 and AlexNet directly. The images are re-sized to fit to the input layers of the pre-trained models. The activation values of the last fully connected layers of these pre-trained CNNs are obtained (fully connected layer 8 (Fc8)) as in the previous approach. 1000 features are gathered from these layers. After the deep feature extraction, deep features from different CNNs are concatenated and reshaped as a matrix with dimensions 40x50. These matrices are given to an RNN model as input. The RNN model used in this step has an input layer with the size 20, an LSTM layer that contain 1000 hidden units, a fully connected layer that has the number of outputs as the number of classes, and a softmax layer.

5. Experimental Results

In this section, we first explain the datasets and then, provide our experimental results and the statistical analysis.

5.1. Datasets

Four publicly available plant datasets are used during our experiments. We first performed our experiments on Flavia and Swedish Leaf datasets because they mostly contain clean images and no variations of luminance. Also, they are widely used datasets in this domain and publicly available. Later, we added the UCI Leaf dataset, which is hosted in UCI Machine Learning Repository because it is mostly applied by researchers for the comparison of algorithms. We continued to our experiments with Plantvillage dataset because it includes 50 times more samples compared to the samples in Flavia and Swedish Leaf and we aimed to analyse our models on this large dataset. However, our models are not dependent on these datasets and can be applied on different plant datasets.

The first one is the Flavia Dataset created by Wu et al. [49], which contains 32 species and 1900 images. Sample images are presented in Figure 5. The second one is the Swedish Leaf Dataset generated by Sderkvist [50]. There are leaf images which belong to 15 tree classes. Sample images are presented in Figure 6. The third one is the UCI Leaf Dataset built by Silva et al. [51]. There are 40 different species of plants included in this dataset with a total number of 443 images. Sample images are presented in Figure 7. The fourth dataset is PlantVillage dataset created by Mohanty et al. [52]. There are 14 different species of plants and 38 classes (healthy-diseased), having a total number of 54,306 images. The healthy plant images are included during the experiments. Sample images are presented in Figure 8. The properties of the datasets are presented in Table 2.

Table 2: Properties of the experimented datasets

Dataset	# of Classes	# of Samples	Image Background	Color
Flavia	32	1907	White	RGB
Swedish Leaf	15	1125	White	RGB
UCI Leaf	40	443	Pink	RGB
Plantvillage	14*(38**)	54306	Transparent	RGB

** healthy, ** healthy+diseased*

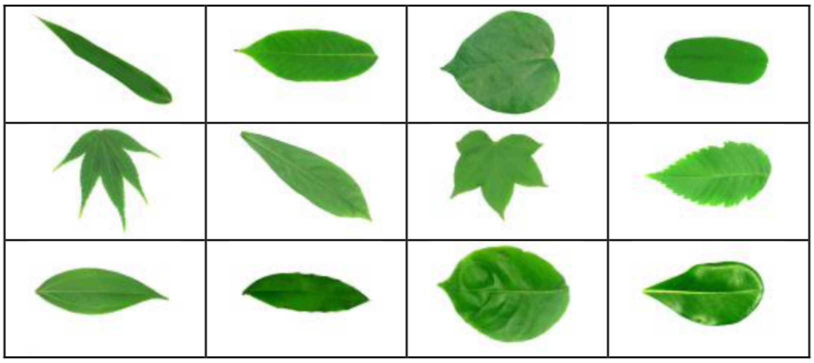

Figure 5: Sample images from Flavia Dataset

5.2. Results

All the deep learning based models have 100 epochs during the training phase. Datasets are divided into test and training sets with a ratio of 30%, and 70% respectively. During deep feature learning experiments, linear kernel SVM and LDA classifiers are used with five-fold (k=5) cross-validation. As explained in the previous sections, experiments were performed on several datasets with different sizes. Some of the datasets have a smaller number of samples. The use of a large k value would cause to have a limited number of samples in the

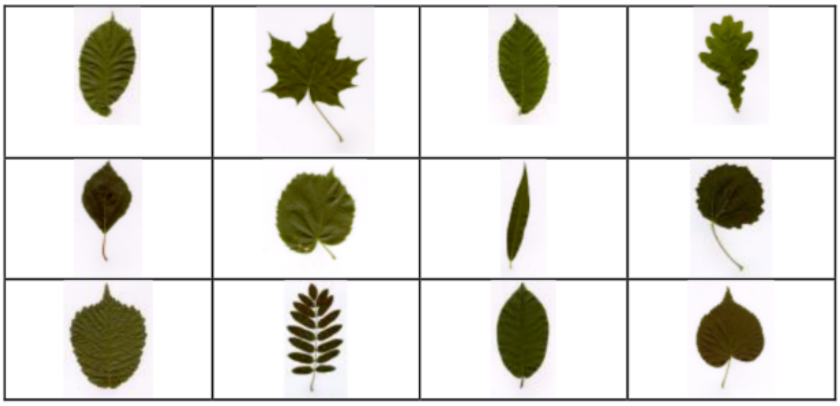

Figure 6: Sample images from Swedish Leaf Dataset

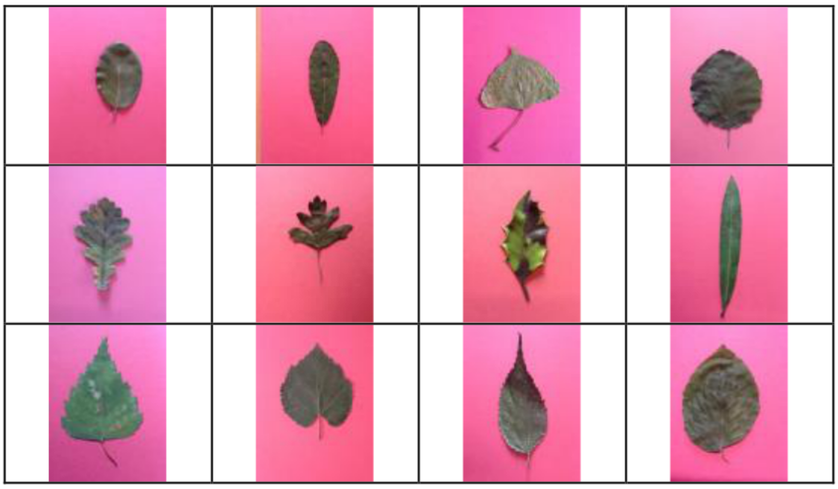

Figure 7: Sample images from UCI Leaf Dataset

test set for the classes with a small number of samples. Increasing the number of k would probably reduce the overfitting, but this time computational cost would increase. To cope with the trade-off between the accuracy and the performance, we set the k value to 5.

Since each of these approaches were discussed in the previous section, we provide only a figure to reflect the differences between different configurations.

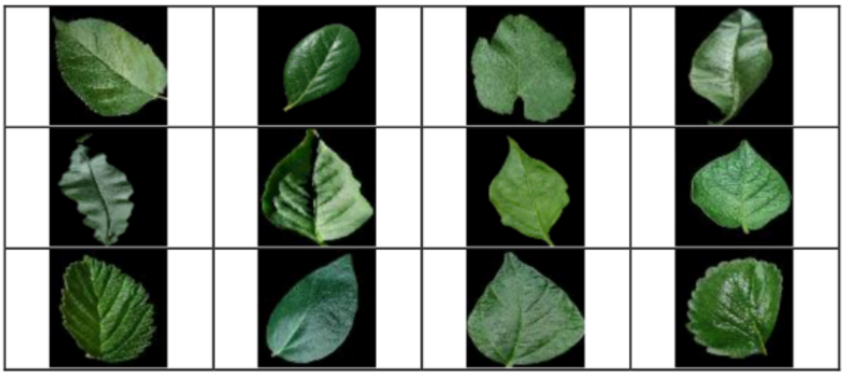

Figure 8: Sample images from Plantvillage Dataset

Results of the end-to-end CNN model is provided in Table 3. According to this table, it is observed that the same CNN model performs better on the datasets with higher sample amount, and this observation is consistent with the general idea of deep learning models, which state that the models trained with huge amount of data provide better classification outcomes. UCI Leaf dataset has the smallest size and therefore, the classification accuracy on this dataset is not as good as the performance of the model on the other datasets.

Table 3: Classification Accuracy of the End-To-End CNN Model

Dataset	**CA**
Flavia	91.08
SwedishLeaf	96.06
UCI Leaf	76.15
PlantVillage	97.40

The cross-dataset fine-tuning experiment results are presented in Table 4. UCI Leaf Dataset result is improved by 4%, however, the performance on other datasets is similar to the end-to-end CNN results. This experiment demonstrated that increasing the data size for training positively affects the performance of the model on datasets which have lower sample size.

Table 4: Classification Accuracy of the Cross-Dataset Fine-Tuning of the CNN Model

Test Dataset	**Training Datasets**	**CA**
Flavia	Swedish Leaf UCI Leaf Plantvillage	91.43
Swedish Leaf	Flavia UCI Leaf Plantvillage	96.06
UCI Leaf	Flavia Swedish Leaf Plantvillage	80.60
Plantvillage	Flavia Swedish Leaf UCI Leaf	96.93

In Table 5, results of the fine-tuning of the pre-trained models are presented. The classification performance of all the datasets improves compared to the previous approaches. Pre-trained models are trained with millions of images from the ImageNet dataset. Even if models are trained with images from different domains, fine-tuning of these models provides better results than the end-to-end models on especially datasets with limited elements. According to this experimental analysis, it is observed that the VGG16 network provides relatively better performance than the Alexnet, but the difference between these two approaches is not very significant.

Table 5: Classification Accuracy of the Fine-Tuning of Pre-Trained Models

Dataset	**Pre-Trained Model**	**CA**
Flavia	Alexnet	97.89
	VGG16	98.16
Swedish Leaf	Alexnet	95.56
	VGG16	99.11
UCI Leaf	Alexnet	89.41
	VGG16	90.56
Plantvillage	Alexnet	98.60
	VGG16	99.80

Experimental results of classification with deep features are provided in Table 6. LDA and SVM methods perform similarly on all the datasets except the UCI Leaf dataset. The best classification outcomes are obtained from these experiments, especially for the UCI dataset. In the first experiment, the performance of CNN was around 76 percent on the UCI leaf dataset, but the performance of the model with deep features is beyond this performance value.

Table 6: Classification Accuracy of Classifiers with Deep Features

Dataset	**PT-Model / Classifier**	**CA**
Flavia	Alexnet / LDA	99.00
	Alexnet / SVM	97.50
	VGG16 / LDA	99.10
	VGG16 / SVM	97.70
Swedish Leaf	Alexnet / LDA	95.80
	Alexnet / SVM	97.80
	VGG16 / LDA	96.10
	VGG16 / SVM	98.80
UCI Leaf	Alexnet / LDA	96.20
	Alexnet / SVM	88.90
	VGG16 / LDA	94.80
	VGG16 / SVM	89.60
Plantvillage	Alexnet / LDA	98.70
	Alexnet / SVM	97.80
	VGG16 / LDA	98.70
	VGG16 / SVM	98.00

Results of the classification when the CNN-RNN combination is applied are shown in Table 7. According to this table, the same classification model performed better on datasets which have higher sample amount as in the results of end-to-end CNN experiments. The model trained with the UCI Leaf dataset provides the lowest classification accuracy (70.79%). The performance of the models developed here is comparable with the performance of our other experiments.

Table 7: Classification Accuracy of CNN-RNN Models on Datasets

Dataset	**CA**
Flavia	92.65
Swedish Leaf	99.11
UCI Leaf	70.79
Plantvillage	98.77

In Figure 9, all the results are presented with a bar graph. In Table 8 - Table 11, the statistical significance analysis is performed for the analyzed methods which are represented as follows: M1: End-to-end CNN, M2: cross-dataset, M3: FT - Alexnet, M4: FT - VGG16, M5: DF - Alexnet / LDA, M6: DF - Alexnet / SVM, M7: DF - VGG16 / LDA, M8: DF - VGG16 / SVM, M9: CNN - RNN. McNemar's test is used to measure the statistical significance of the methods according to classification outcomes. All the classes are tested one-versus-all manner (one taken as positive, others are considered as negative class). Significant values ($p < 0.05$) are implied with boldface font.

From Table 4-11, we demonstrated that the transfer learning approaches perform better than the end-to-end models for all the benchmarking datasets. This difference is very clear for the UCI Leaf dataset which has a smaller sample size compared to the other datasets.

VGG16/LDA methods overall performance is the best one. For Flavia, Plantvillage, and UCI Leaf datasets, VGG16/LDA provide relatively higher performance. The performance is lower in Swedish Leaf dataset, but this method can be considered as the most successful method on these datasets. The end-to-end CNN model provides the worst performance for all the datasets. For Flavia, Plantvillage, and UCI Leaf, all the methods provide better results compared to the performance of end-to-end CNN and the cross-dataset methods.

The Plantvillage is the dataset which has the highest number of samples. The difference between deep learning based methods trained on this dataset is marginal. FT-VGG16 is the best performing one in this dataset, and all comparisons with this method are statistically significant, thus it is the most preferable method.

The similar situation is also valid for the Swedish Leaf dataset. The difference and the statistical significance between methods are lower in these datasets compared to the other ones. When compared to the End-to-End CNN models, nearly all of the methods provided statistically significant results in all benchmark datasets. Considering the classification performance of this approach, its preference is low compared to other methods.

When the all comparisons are considered, the least statistical significance is seen in the comparisons with the FT-Alexnet method. This method provided close but not the best results in many of the datasets, except UCI Leaf.

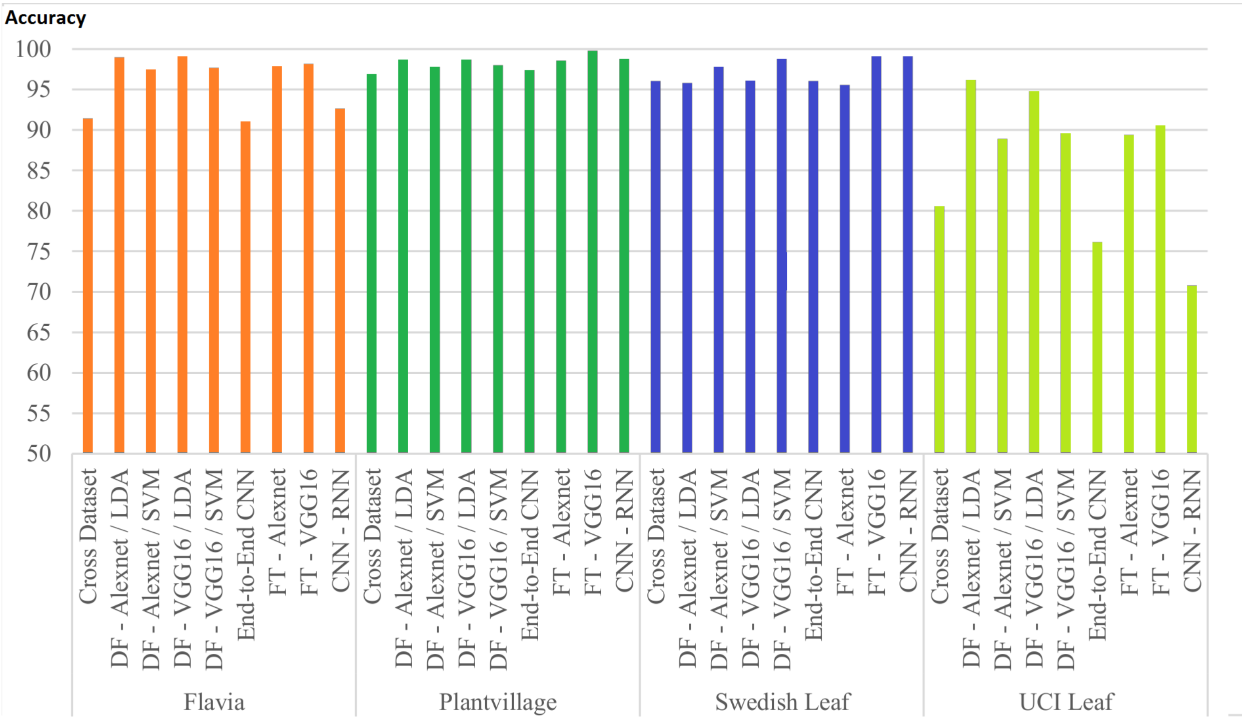

Figure 9: Graph of experimental results

Table 8: Statistical significance test results on Flavia dataset

	M2	M3	M4	M5	M6	M7	M8	M9
M1	0.75	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.14
M2	-	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.25
M3	-	-	0.62	**0.02**	0.51	**0.01**	0.74	**0.00**
M4	-	-	-	0.07	0.25	**0.04**	0.41	**0.00**
M5	-	-	-	-	**0.00**	0.79	**0.01**	**0.00**
M6	-	-	-	-	-	**0.00**	0.74	**0.00**
M7	-	-	-	-	-	-	**0.00**	**0.00**
M8	-	-	-	-	-	-	-	**0.00**

Table 9: Statistical significance test results on Plantvillage dataset

	M2	**M3**	**M4**	**M5**	**M6**	**M7**	**M8**	**M9**
M1	0.47	**0.03**	**0.00**	**0.02**	0.50	**0.02**	0.31	**0.01**
M2	-	**0.00**	**0.00**	**0.00**	0.17	**0.00**	0.08	**0.00**
M3	-	-	**0.00**	0.82	0.12	0.82	0.24	0.70
M4	-	-	-	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
M5	-	-	-	-	0.08	1.00	0.16	0.87
M6	-	-	-	-	-	0.08	0.72	0.06
M7	-	-	-	-	-	-	0.16	0.87
M8	-	-	-	-	-	-	-	0.12

Table 10: Statistical significance test results on the Swedish Leaf dataset

	M2	**M3**	**M4**	**M5**	**M6**	**M7**	**M8**	**M9**
M1	1.00	0.52	**0.00**	0.74	**0.01**	0.96	**0.00**	**0.00**
M2	-	0.52	**0.00**	0.74	**0.01**	0.96	**0.00**	**0.00**
M3	-	-	**0.00**	0.76	**0.00**	0.49	**0.00**	**0.00**
M4	-	-	-	**0.00**	**0.01**	0.00	0.44	1.00
M5	-	-	-	-	**0.00**	0.70	**0.00**	**0.00**
M6	-	-	-	-	-	**0.01**	**0.05**	**0.01**
M7	-	-	-	-	-	-	**0.00**	**0.00**
M8	-	-	-	-	-	-	-	0.44

Table 11: Statistical significance test results on the UCI Leaf dataset

	M2	M3	M4	M5	M6	M7	M8	M9
M1	**0.01**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
M2	-	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
M3	-	-	0.33	**0.00**	0.68	**0.00**	0.87	**0.00**
M4	-	-	-	**0.00**	0.16	**0.00**	0.41	**0.00**
M5	-	-	-	-	**0.00**	0.08	**0.00**	**0.00**
M6	-	-	-	-	-	**0.00**	0.56	**0.00**
M7	-	-	-	-	-	-	**0.00**	**0.00**
M8	-	-	-	-	-	-	-	**0.00**

6. Discussion

The related work section showed that the effect of transfer learning on the performance of deep learning models has not been investigated. We comparatively assessed the performance of five transfer learning scenarios on the deep learning models. We presented our Research Questions (RQs) in Section 1 and here, we discuss our research findings. To repeat the research questions were the following:

- RQ1: To what extent can plant classification benefit from transfer learning in deep learning?
- RQ-2: How do the different transfer learning scenarios perform at improving the performance of plant classification models using deep learning?

To answer these questions we have compared four different transfer learning approaches with the base, end-to-end CNN approach. For our comparison we mainly focused on classification accuracy (CA). Our study showed that transfer learning does have an impact on the classification accuracy. We have shown that transfer learning approaches perform better than the end-to-end models for the benchmarking datasets. In particular, a clear performance benefit can be observed for datasets which have less number of data points.

Experimental studies consist of potential threats to validity [53]. Regarding the internal validity, we did not apply just one transfer learning approach to evaluate the impact of transfer learning on deep learning models. Instead, we designed and implemented four different transfer learning scenarios for comparative assessment of models. While we covered several transfer learning scenarios during our experiments, new studies might apply other approaches with different settings and reach new results. We focused on deep learning models instead of traditional machine learning models because we aimed to benefit pre-trained deep learning-based plant classification models.

Regarding the external validity, our conclusions are valid for the datasets explained in Datasets subsection and observations might be different on a new set of datasets. We preferred the datasets which follow FAIR (Findable / Accessible / Interoperable / Reusable) principles [54] and therefore, results might be different on datasets which do not adopt FAIR principles.

Regarding the construct validity, we did our experiments on widely-used datasets which are still applied by other researchers in this field. Conclusion validity addresses threats which impact the ability to conclude appropriately. We split the datasets into training and test sets by using the widely-used ratio (70%-30%) and five-fold cross-validation was applied during deep feature learning experiments with linear kernel SVM and LDA classifiers. Also, 100 epochs were used for all the deep learning based models. These validation techniques were applied to avoid the randomness in data. Apart from this techniques, a statistical analysis was also performed to check the statistical significance of our experimental results. Results were reported based on accuracy parameter. Four public datasets were analyzed during our experiments and further experiments are required for new datasets.

7. Conclusion

Correct classification of plant species has many advantages not only in agriculture, but also in several other domains such as, for example biodiversity, health and forest studies. Instead of manually processing of plants by experts, automated plant identification systems enable stakeholders to quickly deal with the huge amount of plants and lessen the required time and cost of these operations.

While there are some studies on the use of deep learning algorithms for the plant classification, there has not been an in-depth study which applies several transfer learning scenarios for deep learning models. Hence our study can be considered as complementary to the existing studies paving the way for further research on transfer learning for plant classification.

This study demonstrated that the transfer learning improves the performance of deep learning models and especially, models which apply deep features and use fine-tuning provide better performance compared to the other transfer learning strategies. This result implies that instead of only applying an end-to-end CNN model for the plant classification, the other transfer learning approaches must also be considered in the low accuracy performance case.

In the near future, we are planning to investigate the performance of these models for different applications in agriculture such as plant disease detection and weeds detection. Another dimension is to perform new experiments when more public datasets become available.

Acknowledgements

The authors are grateful to the infrastructure support of Wageningen University and Hacettepe University.

References

[1] R. W. Scotland, A. H. Wortley, How many species of seed plants are there?, Taxon 52 (1) (2003) 101–104.

[2] C. Mora, D. P. Tittensor, S. Adl, A. G. Simpson, B. Worm, How many species are there on earth and in the ocean?, PLoS biology 9 (8) (2011) e1001127.

[3] R. Govaerts, How many species of seed plants are there?, Taxon 50 (4) (2001) 1085–1090.

[4] J. Wäldchen, M. Rzanny, M. Seeland, P. Mäder, Automated plant species identificationtrends and future directions, PLoS computational biology 14 (4) (2018) e1005993.

[5] G. Hopkins, R. P. Freckleton, Declines in the numbers of amateur and professional taxonomists: implications for conservation, Animal Conservation 5 (3) (2002) 245–249.

[6] K. J. Gaston, M. A. O'Neill, Automated species identification: why not?, Philosophical Transactions of the Royal Society of London B: Biological Sciences 359 (1444) (2004) 655–667.

[7] G. L. Grinblat, L. C. Uzal, M. G. Larese, P. M. Granitto, Deep learning for plant identification using vein morphological patterns, Computers and Electronics in Agriculture 127 (2016) 418–424.

[8] T. Pahikkala, K. Kari, H. Mattila, A. Lepistö, J. Teuhola, O. S. Nevalainen, E. Tyystjärvi, Classification of plant species from images of overlapping leaves, Computers and Electronics in Agriculture 118 (2015) 186–192.

[9] S. H. Lee, C. S. Chan, S. J. Mayo, P. Remagnino, How deep learning extracts and learns leaf features for plant classification, Pattern Recognition 71 (2017) 1–13.

[10] M. Dyrmann, H. Karstoft, H. S. Midtiby, Plant species classification using deep convolutional neural network, Biosystems Engineering 151 (2016) 72–80.

[11] H. Yalcin, S. Razavi, Plant classification using convolutional neural networks, in: Agro-Geoinformatics (Agro-Geoinformatics), 2016 Fifth International Conference on, IEEE, 2016, pp. 1–5.

[12] W. Strothmann, A. Ruckelshausen, J. Hertzberg, C. Scholz, F. Langsenkamp, Plant classification with in-field-labeling for crop/weed discrimination using spectral features and 3d surface features from a multi-wavelength laser line profile system, Computers and Electronics in Agriculture 134 (2017) 79–93.

[13] D. M. Woebbecke, G. E. Meyer, K. Von Bargen, D. Mortensen, Color indices for weed identification under various soil, residue, and lighting conditions, Transactions of the ASAE 38 (1) (1995) 259–269.

[14] R. Gerhards, S. Christensen, Real-time weed detection, decision making and patch spraying in maize, sugarbeet, winter wheat and winter barley, Weed research 43 (6) (2003) 385–392.

[15] L. Sack, E. M. Dietrich, C. M. Streeter, D. Sánchez-Gómez, N. M. Holbrook, Leaf palmate venation and vascular redundancy confer tolerance of hydraulic disruption, Proceedings of the National Academy of Sciences 105 (5) (2008) 1567–1572.

[16] C. Scoffoni, M. Rawls, A. McKown, H. Cochard, L. Sack, Decline of leaf hydraulic conductance with dehydration: relationship to leaf size and venation architecture, Plant Physiology 156 (2) (2011) 832–843.

[17] G. Agarwal, P. Belhumeur, S. Feiner, D. Jacobs, W. J. Kress, R. Ramamoorthi, N. A. Bourg, N. Dixit, H. Ling, D. Mahajan, et al., First steps toward an electronic field guide for plants, Taxon 55 (3) (2006) 597–610.

[18] J. C. Neto, G. E. Meyer, D. D. Jones, A. K. Samal, Plant species identification using elliptic fourier leaf shape analysis, Computers and electronics in agriculture 50 (2) (2006) 121–134.

[19] Z. Husin, A. Shakaff, A. Aziz, R. Farook, M. Jaafar, U. Hashim, A. Harun, Embedded portable device for herb leaves recognition using image processing techniques and neural network algorithm, Computers and electronics in agriculture 89 (2012) 18–29.

[20] R. Pydipati, T. Burks, W. Lee, Identification of citrus disease using color texture features and discriminant analysis, Computers and electronics in agriculture 52 (1-2) (2006) 49–59.

[21] M. G. Larese, R. Namías, R. M. Craviotto, M. R. Arango, C. Gallo, P. M. Granitto, Automatic

classification of legumes using leaf vein image features, Pattern Recognition 47 (1) (2014) 158–168.

[22] M. G. Larese, A. E. Baya, R. M. Craviotto, M. R. Arango, C. Gallo, P. M. Granitto, Multiscale recognition of legume varieties based on leaf venation images, Expert Systems with Applications 41 (10) (2014) 4638–4647.

[23] H. Mattila, P. Valli, T. Pahikkala, J. Teuhola, O. S. Nevalainen, E. Tyystjärvi, Comparison of chlorophyll fluorescence curves and texture analysis for automatic plant identification, Precision agriculture 14 (6) (2013) 621–636.

[24] N. Wang, N. Zhang, J. Wei, Q. Stoll, D. Peterson, A real-time, embedded, weed-detection system for use in wheat fields, Biosystems Engineering 98 (3) (2007) 276–285.

[25] E. Tyystjärvi, M. Nørremark, H. Mattila, M. Keränen, M. Hakala-Yatkin, C.-O. Ottosen, E. Rosenqvist, Automatic identification of crop and weed species with chlorophyll fluorescence induction curves, Precision Agriculture 12 (4) (2011) 546–563.

[26] A. Muthevi, R. B. Uppu, Leaf classification using completed local binary pattern of textures, in: Advance Computing Conference (IACC), 2017 IEEE 7th International, IEEE, 2017, pp. 870–874.

[27] M. Murat, S.-W. Chang, A. Abu, H. J. Yap, K.-T. Yong, Automated classification of tropical shrub species: a hybrid of leaf shape and machine learning approach, PeerJ 5 (2017) e3792.

[28] E. Yousefi, Y. Baleghi, S. M. Sakhaei, Rotation invariant wavelet descriptors, a new set of features to enhance plant leaves classification, Computers and Electronics in Agriculture 140 (2017) 70–76.

[29] X. Yu, S. Xiong, Y. Gao, Y. Zhao, X. Yuan, Multiscale crossing representation using combined feature of contour and venation for leaf image identification, in: Digital Image Computing: Techniques and Applications (DICTA), 2016 International Conference on, IEEE, 2016, pp. 1–6.

[30] K. Horaisová, J. Kukal, Leaf classification from binary image via artificial intelligence, Biosystems Engineering 142 (2016) 83–100.

[31] A. dos Santos Ferreira, D. M. Freitas, G. G. da Silva, H. Pistori, M. T. Folhes, Weed detection in soybean crops using convnets, Computers and Electronics in Agriculture 143 (2017) 314–324.

[32] P. Barré, B. C. Stöver, K. F. Müller, V. Steinhage, Leafnet: A computer vision system for automatic

plant species identification, Ecological Informatics 40 (2017) 50–56.

[33] W.-S. Jeon, S.-Y. Rhee, Plant leaf recognition using a convolution neural network, International Journal of Fuzzy Logic and Intelligent Systems 17 (1) (2017) 26–34.

[34] A. Caglayan, O. Guclu, A. B. Can, A plant recognition approach using shape and color features in leaf images, in: International Conference on Image Analysis and Processing, Springer, 2013, pp. 161–170.

[35] J. Nam, W. Fu, S. Kim, T. Menzies, L. Tan, Heterogeneous defect prediction, IEEE Transactions on Software Engineering.

[36] C. Wang, S. Mahadevan, Heterogeneous domain adaptation using manifold alignment, in: IJCAI proceedings-international joint conference on artificial intelligence, Vol. 22, 2011, p. 1541.

[37] D. Cook, K. D. Feuz, N. C. Krishnan, Transfer learning for activity recognition: A survey, Knowledge and information systems 36 (3) (2013) 537–556.

[38] S. J. Pan, Q. Yang, et al., A survey on transfer learning, IEEE Transactions on knowledge and data engineering 22 (10) (2010) 1345–1359.

[39] K. Weiss, T. M. Khoshgoftaar, D. Wang, A survey of transfer learning, Journal of Big Data 3 (1) (2016) 9.

[40] O. Day, T. M. Khoshgoftaar, A survey on heterogeneous transfer learning, Journal of Big Data 4 (1) (2017) 29.

[41] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in neural information processing systems, 2012, pp. 1097–1105.

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

[43] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556.

[44] H.-C. Shin, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, R. M. Summers, Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics

and transfer learning, IEEE transactions on medical imaging 35 (5) (2016) 1285–1298.

[45] C.-K. Shie, C.-H. Chuang, C.-N. Chou, M.-H. Wu, E. Y. Chang, Transfer representation learning for medical image analysis, in: Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE, IEEE, 2015, pp. 711–714.

[46] T. Burks, S. Shearer, F. Payne, Classification of weed species using color texture features and discriminant analysis, Transactions of the ASAE 43 (2) (2000) 441.

[47] C. A. Priya, T. Balasaravanan, A. S. Thanamani, An efficient leaf recognition algorithm for plant classification using support vector machine, in: Pattern Recognition, Informatics and Medical Engineering (PRIME), 2012 International Conference on, IEEE, 2012, pp. 428–432.

[48] A. Karpathy, The unreasonable effectiveness of recurrent neural networks, Andrej Karpathy blog.

[49] S. G. Wu, F. S. Bao, E. Y. Xu, Y.-X. Wang, Y.-F. Chang, Q.-L. Xiang, A leaf recognition algorithm for plant classification using probabilistic neural network, in: Signal Processing and Information Technology, 2007 IEEE International Symposium on, IEEE, 2007, pp. 11–16.

[50] O. Söderkvist, Computer vision classification of leaves from swedish trees (2001).

[51] P. F. Silva, A. R. Marcal, R. M. A. da Silva, Evaluation of features for leaf discrimination, in: International Conference Image Analysis and Recognition, Springer, 2013, pp. 197–204.

[52] S. P. Mohanty, D. P. Hughes, M. Salathé, Using deep learning for image-based plant disease detection, Frontiers in plant science 7 (2016) 1419.

[53] W. Claes, R. Per, H. Martin, C. Magnus, R. Björn, A. Wesslén, Experimentation in software engineering: an introduction, Online Available: http://books. google. com/books.

[54] M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L. B. da Silva Santos, P. E. Bourne, et al., The fair guiding principles for scientific data management and stewardship, Scientific data 3.