Pamukkale Univ Muh Bilim Derg, 27(5), 638-645, 2021

Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi

Pamukkale University Journal of Engineering Sciences

Plant identification with convolutional neural networks and transfer learning

Konvolüsyonel yapay sinir ağları ve öğrenme transferi ile bitki tanıma

Tolgahan KARAHAN [1], *Vasif NABİYEV*[2*]

[1,2]Computer Engineering Department, Engineering Faculty, Karadeniz Technical University, Trabzon, Turkey.
karahantlg@gmail.com, vasif@ktu.edu.tr

Received/Geliş Tarihi: 30.04.2020 Revision/Düzeltme Tarihi: 14.11.2020 doi: 10.5505/pajes.2020.84042
Accepted/Kabul Tarihi: 20.11.2020 Research Article/Araştırma Makalesi

Abstract

Nature is rich with a vast amount of plant and flower species and because of their great diversity; identification of these species requires expertise in the field. Development of an automatic plant identification system can ease this process. In this work, deep Convolutional Neural Networks and Transfer Learning have been utilized in order to develop such an identification system. Images in the database have been collected from other databases and the web and in total it consists of 5,345 flowers and plant images belong to 76 species. 65 of the species are various flower species and 11 of them are other plant species. Data augmentation techniques has been applied in order to increase the number of images in the database and to improve the generalization capacity of the model. For data augmentation, random rotation at four angles, random brightness change in the range of [-0.2, 0.2] and horizontal flip have been applied. Also preprocessing techniques such as center cropping and normalizing have been applied to images before input them to the model. In automatic plant recognition, 0.9971 accuracy achieved on the training set and 0.9897 accuracy achieved on the test set.

Keywords: Yapay zekâ, Derin öğrenme, Konvolüsyonel yapay sinir ağları, Tanımlama, Sınıflandırma.

Öz

Doğa büyük miktarda bitki ve çiçek türü zenginliğine sahiptir ve bu çeşitlilik nedeniyle bu türlerin tanımlanması botanik alanında uzmanlık gerektirmektedir. Bir otomatik bitki tanıma sisteminin geliştirilmesi bu süreci kolaylaştıracaktır. Bu çalışmada, böyle bir sistemi geliştirmek için konvolüsyonel yapay sinir ağlarından ve öğrenme transferinden faydalanılmıştır. Veritabanındaki görüntüler diğer veritabanlarından ve webden toplanmıştır ve toplamda 76 türe ait 5.345 çiçek ve bitkiden oluşmaktadır. Türlerin 65 tanesi çeşitli çiçek türleridir ve 11 tanesi ise diğer bitki çeşitlerindendir. Veritabanındaki görüntü sayısını ve modelin genelleme kapasitesini arttırmak için çeşitli veri çoğaltma yöntemleri uygulanmıştır. Veri çoğaltmak için, 4 açıdan rastgele döndürme, [-0.2, 0.2] aralığında rastgele parlaklık değişimi ve yatay yansıtma işlemleri uygulanmıştır. Aynı zamanda, görüntüleri modele girdi olarak vermeden önce, görüntülere merkezden kesme ve normalizasyon işlemleri uygulanmıştır. Geliştirilen model eğitim verileri için 0.9971, test verileri için ise 0.9897 isabet oranı elde etmiştir.

Anahtar kelimeler: Artificial intelligence, Deep learning, Convolutional neural networks, Identification, Classification.

1 Introduction

Classification of plant and flower species has been subject to various studies. Various methods and models have been developed in order to classify plant or flower species. In recent years, Convolutional Neural Networks have become prominent among other approaches and they have some important advantages over traditional approaches. They have much less parameters than traditional artificial neural networks, so they are more efficient in terms of computation. They are less prone to overfitting problems because of the parameter sharing between different regions of the images [1]. Also, they have the ability to learn more abstract features. For these reasons state of the art image classification models are based on Convolutional Neural Networks.

CNNs have three main types of layers [1]. These layers are convolutional layers, pooling layers, and fully-connected layers. Convolutional layers apply convolution operation to their inputs via kernels and values of these kernels are learned during training process. Pooling layers apply downsampling to their inputs in order to decrease spatial size and fully connected layers use features learned via previous convolutional layers and make classification.

Developments in hardware and using GPU's for parallel implementations make it possible to train deep networks on large scale databases. ImageNet is such a large database and it is commonly used for evaluating the performance of visual recognition algorithms [2]. Deep Convolutional Neural Network models have demonstrated impressive performances on the ImageNet benchmark [3]. As a result they are used in a wide variety of Computer Vision tasks.

Using Convolutional Neural Networks with a technique called Transfer Learning significantly increases classification results. Therefore, the Transfer Learning technique has been used also in this study. Transfer learning can be accomplished by using learned parameters from a different, but related domain. In this way we can adapt learned parameters to our task and make our model to achive better generalization capacity [4]. In this regard, transfer learning can be utilized by adapting large scale networks trained on ImageNet to the task at hand.

Another useful technique to improve generalization capability of Computer Vision models is data augmentation. In data augmentation, samples can be increased by generating synthetic data. New samples can be generated by rotating, flipping, cropping, or distorting the original samples. It can improve model's generalization capability, because patterns

*Corresponding author/Yazışılan Yazar

are introduced to model in different situations [5]. Thus a more robust model is achieved. Data augmentation techniques such as random rotation at four angles, random brightness change in the range of [-0.2, 0.2] and horizontal flip have been used in this study.

In order to develop a plant and flower identification system a dataset has been created. The image database has been split to training and test sets by stratified sampling [6]. Final training set consists of 29932 images and the final test set consists of 7483 images. Thanks to the stratified sampling, each class has been introduced into both training and test sets.

Center cropping and normalization were applied to images before they are provided as input to the model. After 15 epochs, the proposed model has achieved 0.9971 accuracy on the training set and 0.9897 accuracy on the test set.

Also trained model has been used for developing a mobile plant/flower identification application for Android platforms. By using developed mobile application, proposed model has been tested with wild plants.

2 Related work

O'Shea et al. have described Convolutional Neural Networks and their advantages over traditional neural networks. Authors have made explanations about building blocks of CNN's, their overall architecture and operations of each layer [1].

Alex Krizhevsky, et al. proposed their CNN model for ImageNet classification. They have presented the success of CNN models and utilizing GPU's to accelerate Deep Learning [7].

Asifullah Khan et al. have presented a survey on recent CNN architectures. They have described the history of CNN's, their components and the architectural evolution of CNN models. They also mentioned applications of CNN's and recent techniques used with them [8].

C Tan et al. have presented a survey on Deep Transfer Learning. They have described problems that can be overcome with transfer learning and the advantages it provides. They categorized approaches for transfer learning and presented this category [9].

L.Perez and J.Wang have compared various data augmentation techniques and their effects on image classification. They applied traditional and recent data augmentation methods on a subset of ImageNet dataset and presented their results. They also proposed their own method called neural augmentation. In this method a neural network has been used to learn augmentations for best improvements on classifiers [10].

Dyrmann, et al. developed a CNN model for classification of 22 different weed and crop species at early grow stages in order to help site specific weed management. Their image dataset consists of 10,413 images compiled from six different datasets. They also used preprocessing and data augmentation techniques in order to improve classification capacity of the network. They removed all non-green pixels from images via segmentation and by mirroring and rotating the original bineimages they applied data augmentation. They achieved 86.20% accuracy on test set [11].

Guillermo L. Grinblat, et al. replaced a previous machine learning pipeline's feature extraction and classification stages' with CNN model in order to classify white bean, red bean and soybean. They have used vein morphological patterns. In their proposed pipeline, vein morphological patterns extracted via UHMT and central patch extraction applied on binary image outputs of UHMT method. These stages ensured that color and shape information is deleted and only vein morphological patterns are used for classification. Outputs of these stages used as inputs to CNN models and they have two setups named as S1 and S2. They trained several CNN models from models with 2 layers to 6 layers on these images of the vein morphological patterns. They reached the best results at a depth of 5 layers for both setups. For S1 setup, they reached 92.6% mean accuracy and for S2 setup, they reached 96.9% mean accuracy [12].

Kaya, A., et al. analyzed transfer learning methods and different transfer learning scenarios on automatic flower identification. They have designed and implemented five classification models. They have used four different datasets for the evaluation of models and have showed positive impacts of using transfer learning on deep learning models to improve performance [13].

Sue Han Lee, et al used Convolutional Neural Networks to find features that best represent leaf samples. They used a CNN model to learn representations for images of leaves and then they analyzed the features to find most important features by feature visualization techniques. They also quantified the features and find necessary features that represent the leaf data by training a CNN based model on raw leaf data and then applying a Deconvolutional Network to outputs of the CNN based model. They analyzed how CNN characterizes the leaf data and found out that shape of leaves is not a dominant feature, but orders of venation are. They also proposed new features that describe whole leaf structure and local features that focus on leaf venation. They demonstrated that this hybrid global-local feature extraction improves the classification power of plant classification systems [14].

Mario Lasseck presented deep learning techniques for LifeClef 2017. In this work, Lasseck used Deep Convolutional Networks and transfer learning via fine tuning in order to classify 10,000 species and trained several models on different datasets. He also utilized bagging of results from models and data augmentation techniques such as cropping square patches from each image at random positions, horizontal flipping, rotation, random variation of saturation and random variation of lightness applied. His system achieved a mean reciprocal rank (MRR) of 92% and a top-5 accuracy of 96% on the official PlantCLEF test set [15].

Ghazi MM et al. used transfer learning by fine-tuning GoogleLeNet, AlexNet, and VGGNet pre-trained models on LifeCLEF2015 dataset. To decrease chance of overfitting, they also applied data augmentation techniques such as rotation, translation, reflection, and scaling. They fused these different classifiers to improve overall performance. Their combined system achieved 80% accuracy on validation set [16].

Milan Sulc, et al. proposed an automatic recognition system of 10,000 plant species for ExpertLifeCLEF 2018. Their proposed system is based on the Inception-ResNet-v2 and Inception-v4 Convolutional Neural Network architectures and their ensembles. They used preprocessing techniques such as random crop, random left-right flip, brightness and saturation distortion. Instead of using values after the last training epoch they used running averages of the trained network parameters and noticed that it increases the accuracy of their models. Because the class frequencies in the training data follow a long-tailed distribution and hence they have different prior probabilities than the test set they used EM algorithm for

estimation of test set priors by maximization of the likelihood of the test set observations. As a result, they achieved 88.4% accuracy on the full test set [17].

3 Material and methods

3.1 Data sources

The images in the database set are compiled from web searches and the Oxford Flowers Dataset [18]. In total, it consists of 5345 flowers and plants images. Also data augmentation has been applied to the created dataset and thanks to data augmentation its size has been increased to 37415 images [19].

There are 76 different plant species in the image database. 64 of them flower species and 12 of them other plant species. Species belong to Turkey's flora or had brought from abroad to Turkey were chosen as target species.

A second data source that stores information about the species in the image database has been created [19]. Each of the species has corresponding records in this data source. Information regarding species found on online sources and stored as a JSON file [20],[21]. Information about species consists of their name, family name, genus and their origin.

3.2 Data augmentation

After creating the image database, data augmentation techniques have been applied to images. Data augmentation techniques are useful for increasing the generalization ability of the model. For data augmentation four times random rotation, a random brightness change and a horizontal flip have been applied.

For random rotation, four angles have been uniformly sampled in the range of -90 and 90 degrees without replacement. Then each image has been rotated four times according to each of these angles. Reflection has been used as an interpolation method of pixel values after the rotation.

After random rotation, random brightness change has been applied in the range of - 0.2 and 0.2. Output image's pixel values randomly changed in the range of 0.2 times below and above the original pixel values.

Finally, a horizontal flip has been applied to the images. After applying these transformations, 6-fold increase has been achieved on the dataset. Transformations are given in Figure 1.



Figure 1. Original image and transformed images. (Random rotation, random brightness change and horizontal flip have been applied to the original image respectively).

3.3 Preprocessing

Preprocessing operations have been applied to images before providing them to the CNN model as inputs.

In order to extract the most useful features images cropped around the center. After center cropping images have been rescaled to 224 x 224 dimensions.

Finally, normalization has been applied to images in the dataset and pixel values scaled in the range of -1 and 1. In practice, normalization can increase the speed of learning.

3.4 Proposed model

Proposed approach is based on Transfer Learning. Transfer Learning can be utilized for neural networks in two ways: by using a pre-trained network as a feature extractor or by removing the final layer of a pre-trained network and adding classifier for the specified task. The second way can be thought of as two networks are stacked. All or some of the parameters of the pre-trained network can be used or can be trained with appended classifier's parameters. This approach called fine-tuning [22].

In this work fine-tuning approach has been used to transfer learning from ImageNet domain to our task. As a pre-trained network MobileNetV2 has been used, which had been trained on ImageNet database.

MobileNet is an efficient, lightweight CNN architecture that targeted mobile and embedded platforms which have limited computation resources [23]. In this architecture convolutional layers replaced by so-called depthwise separable convolutions in order to perform this operation in a more efficient way.

Convolutional layers are backbone of CNN models. First convolutional layer takes an image as an input and make convolution operation on the image by applying a convolution filter. Numbers of convolutional filters are a hyperparameter and they also determine the depth of the output. Convolutional layers don't break spatial structure of input image and each applied filter produces a two-dimensional output. By stacking together these two-dimensional outputs, feature maps are obtained. In next convolutional layers, inputs and outputs are only feature maps and depth of these feature maps depend on number of filters in each convolutional layer.

Figure 2 shows operations of a Convolutional Neural Network [24].

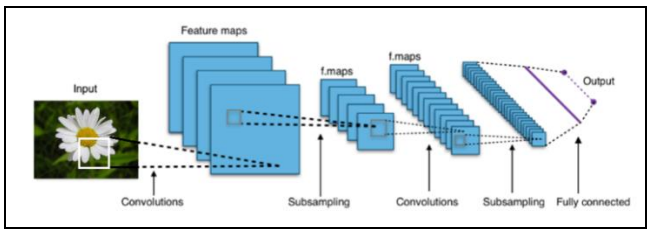


Figure 2. Operation of a standard convolutional layer.

Depthwise separable convolution is consisted of two operations: depthwise convolution and pointwise convolution. Depthwise convolution applies a single filter to each input channel in contrary to standard convolutional layers in which a filter is applied to all channels at once. Then pointwise convolution applies 1x1 convolution to outputs of depthwise convolution operation and combines them [23]. Operation of a depthwise separable convolution block is given in Figure 3 [25].

Output of standard convolutional layer assuming stride one and padding is computed as below [23]:

$$G_{k,l,n} = \sum_{i,j,m} K_{i,j,m,n} * F_{k+i-1,l+j-1,m} \quad (1)$$

Standard convolution operation requires $K \times K \times C_i \times C_o$ parameters where K is spatial width and height of the squared

kernel, C_i is number of input channels and C_o is number of output channels. The total computation cost is $K \times K \times C_i \times C_o \times F \times F$, where F is the spatial width and height of a square feature map.

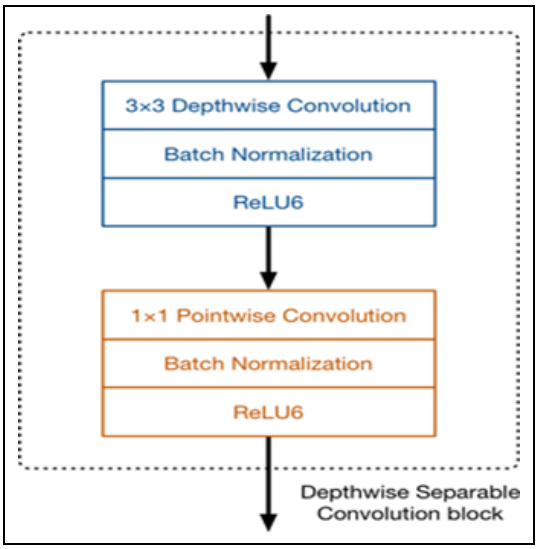


Figure 3. Operation of a depthwise Separable Convolution.

In depthwise separable convolution block, depthwise convolution is used to apply a single filter per input channel and a 1x1 convolution called pointwise convolution is then used to create a linear combination of the output of the depthwise convolution layer. Also batch normalization and ReLU nonlinearities are used for both layers. Depthwise convolution is computed as below [23]:

$$G_{k,l,m} = \sum_{i,j} K_{i,j,m} * F_{k+i-1,l+j-1,m} \quad (2)$$

Depthwise convolution has $K \times K \times C_i \times F \times F$ computational cost. Pointwise convolution also has $C_i \times C_o \times F \times F$, so total cost of the depthwise separable block is $K \times K \times C_i \times F \times F + C_i \times C_o \times F \times F$. Comparing with standard operation of convolutional layer, it leads to reduction in computation of:

$$\frac{K \cdot K \cdot C_i \cdot F \cdot F + C_i \cdot C_o \cdot F \cdot F}{K \cdot K \cdot C_i \cdot C_o \cdot F \cdot F} = \frac{1}{C_i} + \frac{1}{K^2} \quad (3)$$

MobileNetV2 has some architectural changes over MobileNetV1. Besides it still uses depthwise separable and pointwise convolutions, its main building block has been changed. In addition to two layers in the previous dephtwise separable block, a new 1x1 convolution layer has been introduced before the dephtwise convolution layer [23]. Thus, the first 1x1 convolution layer increases the number of channels of the feature map before inputed to the depthwise convolution layer and second 1x1 convolution layer decreases the number of channels of the output feature map of dephtwise convolution layer. Also a residual connection that behaves like a short-cut has been introduced to the network.

Short-cut connections ease the vanishing gradient problem of deep networks and improve the ability of gradients to propagate across network [26]. Main building block of the MobileNetV2, called bottleneck residual block is given in Figure 4 [25].

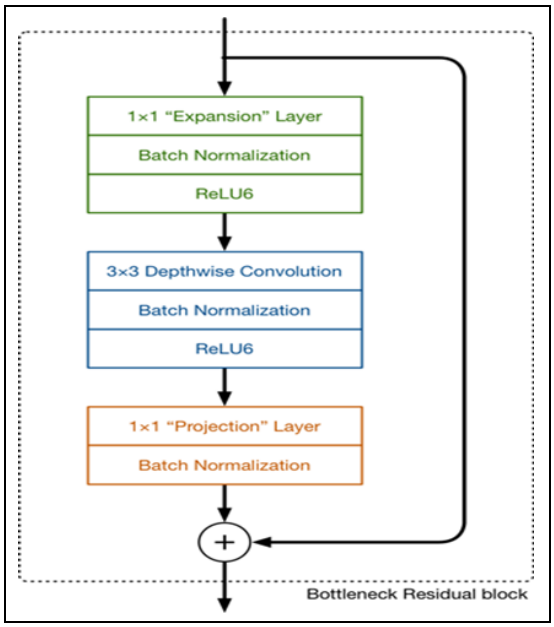


Figure 4. Main building block of MobileNetV2.

Compared with MobileNetV1's dephtwise separable blocks, bottleneck residual blocks has one additional 1x1 convolution layer; however the new architecture provides the ability to use smaller input and output dimensions and a more efficient

Each layer is repeated n times in the architecture. e denotes expansion factor, c denotes number of output channels and s denotes used stride.

For fine-tuning, final layer of the MobileNetV2 architecture has been removed and instead of it, a one layer neural network classifier that comprised of 76 neurons has been used.

In the training phase parameters of the first layers of MobileNet were kept constant and only its last 20 layers and added classifier's parameters were trained.

Images have been provided as input to the network in batches. Each batch has been comprised of 32 images.

Categorical cross-entropy function has been used as a loss function for evaluating the training process and Adamax [27] optimization method has been used as a gradient descent optimizer. Initial learning rate has been selected as 0.02.

4 Experimental results

The model has been developed with Python programming language and trained on Google Colab for utilizing GPU's. Keras, OpenCV and various other libraries used. We also developed a mobile plant identification application based on the proposed model for Android platform [19].

Before the training and test phases, the samples in the database have been shuffled and then split into training and test sets by stratified sampling. The training set has consisted of 29932 images and the test set has consisted of 7483 images. The ratio of images in the test set to all images in the database is 1/5.

In the training process, 0.9971 accuracy rate achieved on the training set after 15 epochs. In each epoch test set also evaluated. Plots of accuracy and loss for the training set are given in Figure 5 and Figure 6 respectively.

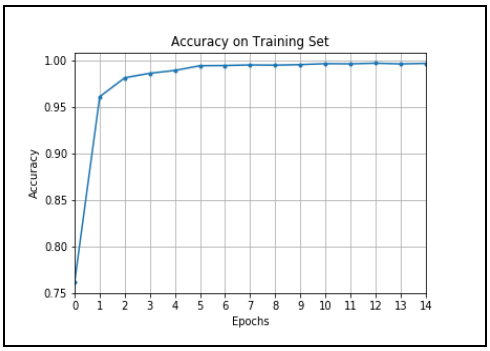


Figure 5. Accuracy respect to each epoch on the training set.

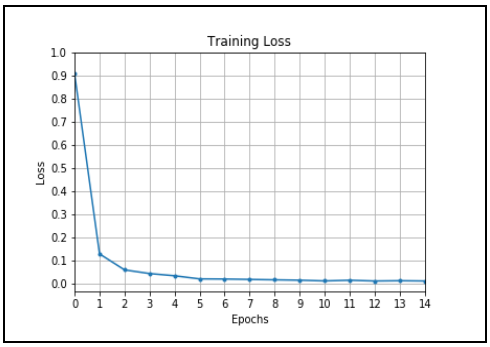


Figure 6. Loss respect to each epoch on the traning set.

The model has been tested on test set after each epoch and as a result 0.9897 accuracy rate achieved on the test set. Plots of accuracy and loss for the test set are given in Figure 7 and Figure 8.

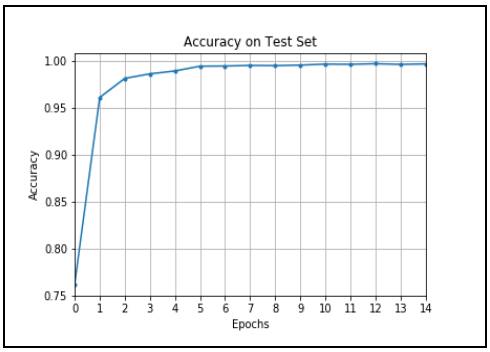


Figure 7. Accuracy respect to each epoch on test set.

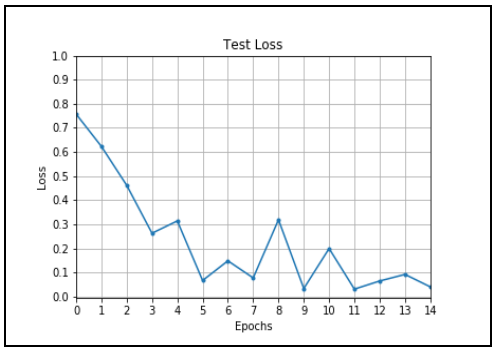


Figure 8. Loss respect to each epoch on test set.

After training, some examples have been drawn from the test set randomly and used for predictions to assess the model visually. Some samples from the test set are given in Figure 9. They have been predicted correctly.

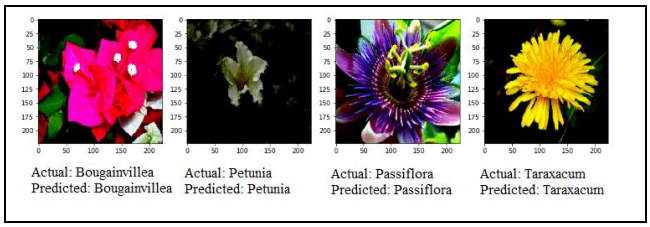


Figure 9. Some additional, visual tests on test set.

Additionally, precision, recall and f1-score metrics have been evaluated for each class. Terms positive and negative refer to predictions of the classifier and terms true and false refer to whether predicted class is correct or not [28]. Their calculations are as follows:

$$precision = \frac{true\ positives}{true\ positives + false\ positives}$$

$$recall = \frac{true\ positives}{true\ positives + false\ negatives}$$

$$f1\ score = \frac{2 * precision * recall}{precision + recall}$$

Comparison with other studies is given in Table 1. Intraclass precision, recall and f1 score metrics are provided in appendix A.

Table 1. Comparison with other studies.

Authors	Number of Classes	Dataset Size	Test Accuracy
Dyrmann et al.	22	10.413	86.2%
Lasseck M.	10000	1.7M	96%
Šulc M et al.	10000	1.75M	88.4%
Ghazi MM, et al.	1000	113.204	80%
Proposed method	76	37.415	98%

Also a mobile plant identification application that based on the proposed model has been developed for Android platform. The application has been developed with Java programming language and the trained model has been integrated to it. It allows users using images from their gallery or taking images by their camera and the application make prediction about the specie in the image. Some information such as name, family, genus and origin is showed to user based on prediction. The application has been tested on wildflowers and plants. A sample is given in Figure 10.



Figure 10. A sample from our mobile application that tested on the wild.

5 Conclusions

In this paper, a plant identification model that based on Convolutional Networks and Transfer Learning has been proposed. It achieved 0.9971 accuracy on the training set and 0.9897 accuracy rate on the test set.

As a conclusion, CNN models by using transfer learning and a mobile application for plant identification have been developed. We hope that this work will be useful in plant and flower identification area.

6 Acknowledgments

The authors would like to thank the Editor and anonymous reviewers for their valuable comments and suggestions to improve the quality of the manuscript.

7 Author contribution statements

In the scope of this study, the Tolgahan KARAHAN, in the literature review, design, data collection, specifying technology stack, implementation, spelling and checking the article in terms of content; Vasif NABİYEV, in the formation of the idea, literature review, review of stages of the study, content suggestions, examining the results, and advising were contributed.

8 Ethics committee approval and conflict of interest statement

There is no need to obtain permission from the ethics committee for the article prepared. There is no conflict of interest with any person/instution in the article prepared.

9 References

[1] O'Shea K et al. "An Introduction to Convolutional Neural Networks". https://arxiv.org/abs/1511.08458 (31.03.2021).

[2] ImageNet. "ImageNet Large Scale Visual Recognition Challenge". http://image-net.org/challenges/LSVRC/ (01.05.2020).

[3] Zeiler MD, Fergus R. "Visualizing and understanding convolutional networks". *ECCV 2014*. Zurich, Switzerland, 6-12 September 2014.

[4] Goodfellow I, Bengio Y, Courville A. *Transfer Learning.* Editors: Dietterich T. Deep Learning, 526-531, Cambridge, Massachusetts, USA, MIT Press, 2016.

[5] Shorten C, Khoshgoftaar TM. "A survey on Image Data Augmentation for deep learning". *Journal of Big Data*, 2019. https://doi.org/10.1186/s40537-019-0197-0.

[6] Lohr S. *Sampling: Design and Analysis*. Editors: Blitzstein SK, Faraway JJ, Tanner M, Zidek J. Stratified Sampling, 73-115, Boca Raton, FL, USA, CRC Press, 2012.

[7] Krizhevsky A, Sutskever I, Hinton GE. "Imagenet classification with deep convolutional neural networks". *Neural Information Processing Systems.* Lake Tahoe, Nevada, USA, 3-6 December 2012.

[8] Khan A, Sohail A, Zahoora U, Qureshi AS. "A Survey of the recent architectures of deep convolutional neural networks". *Artficial Intelligence Review*, 2020. https://doi.org/10.1007/s10462-020-09825-6.

[9] Tan C, Sun F, Kong T, Zhang W, Yang C, Liu C. "A survey on deep transfer learning". *International Conference on Artficial Neural Networks*. Rhodes, Greece, 4-7 October 2018.

[10] Wang J, Perez L. "The effectiveness of data augmentation in ımage classification using deep learning". https://arxiv.org/abs/1712.04621 (31.03.2021).

[11] Dyrmann M, Karstoft H, Midtiby HS." Plant species classification using deep convolutional neural network". *Biosystems Engineering,* 151, 72-80, 2016.

[12] Grinblat GL, Uzal LC, Larese MG. "Deep learning for plant identification using vein morphological patterns". *Computers and Electronics in Agriculture,* 27, 418-424, 2016.

[13] Kaya A, Keçeli AS, Çatal Ç, Yalic HY, Huseyin T, Tekinerdoğan B. "Analysis of transfer learning for deep neural network based plant classification models". *Computers and Electronics in Agriculture,* 158, 20-29, 2019.

[14] Lee HS, Chan CS, Mayo S, Remagnino P. "How deep learning extracts and learns leaf features for plant classification". *Pattern Recognition,* 71, 1-13, 2017.

[15] Lasseck M. "Image-Based plant species ıdentification with deep convolutional neural networks". *CLEF*, Dublin, Ireland, 11-14 September 2017.

[16] Ghazi MM, Yanikoğlu B, Aptoula E. "Plant identification using deep neural networks via optimization of transfer learning parameters". *Neurocomputing*, 235, 228-235, 2017.

[17] Šulc M, Picek L, Matas J. "Plant recognition by inception networks with test-time class prior estimation". *CLEF*, Avignon, France, 10-14 September 2018.

[18] University of Oxford. "102 Category Flower Dataset". http://www.robots.ox.ac.uk/~vgg/data/flowers/102/index.html, (15.06.2019).

[19] Github. "Tolga-Karahan/Mobile-Plant-Identification". https://tolga-karahan.github.io/Mobile-Plant-Identification/(13.11.2020).

[20] Atatürk Orman Çiftliği. "Atatürk Orman Çiftliği". http://www.aoc.gov.tr, (08.09.2020).

[21] Yabani Çiçekler. "Yaban Çiçekler". www.yabanicicekler.com, (08.09.2020).

[22] Yosinski J, Clune J, Bengio Y, Lipson H. "How Transferable are Features in Deep Neural Networks?". https://arxiv.org/abs/1411.1792 (31.03.2021).

[23] Howard AG et al. "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications". https://arxiv.org/abs/1704.04861 (31.03.2021).

[24] Wikipedia Commons. "File: Typical cnn.png". https://commons.wikimedia.org/wiki/File:Typical_cnn.png (12.11.2020).

[25] Machine Think. "MobileNet Version 2" https://machinethink.net/blog/mobilenet-v2/ (05.05.2020).

[26] Sandler M, Howard A, Zhu M, Zhmoginov A, Chen L. "Mobilenetv2: Inverted residuals and linear bottlenecks". *CVPR,* Salt Lake City, USA, 19-21 June 2018.

[27] Kingma DP, Ba JL. "Adam: A method for stochastic optimization". *ICLR,* San Diego, USA,7-9 May 2015.

[28] Kellejer JD, Mac Namee B, D'arcy A. *Fundamentals of Machine Learning for Predictive Data Analytics*. Editor: Mellon M. Precision, Recall, and F1 Measure, 413-417, Cambridge, Massachusetts, USA, The MIT Press, 2015.

Appendix A Precision, Recall and F1-Score

Class	Precision	Recall	f1-score
Ranunculus	1.000	1.000	1.000
Campanula Tomentosa	0.968	0.968	0.968
Echinops	1.000	1.000	1.000
Tussilago	1.000	1.000	1.000
Aconitum	1.000	1.000	1.000
Dianthus	1.000	1.000	1.000
Lilium	1.000	0.986	0.993
Gazania	0.990	1.000	0.995
Vriesea	0.986	1.000	0.993
Dieffenbachia	1.000	1.000	1.000
Cirsium	1.000	1.000	1.000
Eschscholzia	1.000	1.000	1.000
Helleborus	1.000	0.988	0.994
Datura	1.000	0.987	0.993
Convolvulus	1.000	0.899	0.947
Clematis	1.000	1.000	1.000
Euphorbia	1.000	1.000	1.000
Lavandula	0.981	0.981	0.981
Nigella	1.000	0.943	0.971
Taraxacum	1.000	1.000	1.000
Iris Pseuacarus	1.000	1.000	1.000
Trollius	1.000	1.000	1.000
Antirrhinum	0.965	1.000	0.982
Gaillardia	1.000	0.982	0.991
Montera	0.955	1.000	0.977
Bougainvillea	1.000	0.994	0.997
Muscari	1.000	1.000	1.000
Dahlia	1.000	1.000	1.000
Lathyrus	1.000	0.957	0.978
Sansevieria	0.987	1.000	0.993
Zantedeschia	1.000	1.000	1.000
Narcissus	0.987	1.000	0.994
Nephrolepsis	1.000	0.961	0.980
Bellis	1.000	1.000	1.000
Cactus	0.989	1.000	0.994
Echinacea	1.000	1.000	1.000
Calendula	1.000	0.988	0.994
Phlox	0.855	0.985	0.915
Gloriosa	1.000	1.000	1.000
Cycas	0.982	0.964	0.973
Lilium	1.000	1.000	1.000
Dracaena	0.961	1.000	0.980
Cyclamen	0.961	1.000	0.980
Ageratun	0.987	1.000	0.994
Tulipa	1.000	0.986	0.993
Dracaena Fragrans Massengena	0.974	0.987	0.981
Adenium	0.925	1.000	0.961
Canna	1.000	0.989	0.995
Colchium	0.981	1.000	0.991
Passiflora	1.000	1.000	1.000
Papaver	1.000	1.000	1.000
Nymphaea	0.996	1.000	0.998
Primula	1.000	0.974	0.987
Gentiana	0.923	1.000	0.960
Hedera	0.957	1.000	0.978
Astrantia	0.985	1.000	0.993
Thymus	1.000	0.859	0.924
Dimorphotheca	1.000	0.988	0.994
Rhododendron	0.966	0.991	0.979
Cynara	0.960	0.990	0.975
Anthurium	1.000	1.000	1.000
Magnolia	1.000	0.988	0.994
Fritillaria	1.000	0.980	0.990
Oenothera	1.000	1.000	1.000
Rosa	1.000	0.977	0.989
Platycodon	0.982	0.949	0.966
Viola	0.984	1.000	0.992

Appendix A Continued.

Class	Precision	Recall	f1-score
Pelargonium	0.965	1.000	0.982
Petunia	1.000	0.994	0.997
Digitalis	1.000	1.000	1.000
Aechmea	1.000	1.000	1.000
Tagetes	1.000	1.000	1.000
Hibiscus	0.988	1.000	0.994
Ocimum	0.961	0.961	0.961
Helianthus	0.988	1.000	0.994
Gladiolus	1.000	0.925	0.961