UPB HES SO @ PlantCLEF 2017: Automatic Plant Image Identification using Transfer Learning via Convolutional Neural Networks

Alexandru Toma, Liviu Daniel Ştefan, Bogdan Ionescu

Multimedia Lab - CAMPUS, University Politehnica of Bucharest, Romania
bionescu@alpha.imag.pub.ro

Abstract. Recent advances in computer vision have made possible the use of neural networks in large scale image retrieval tasks. An example application is the automated plant classification. However, training a network from scratch takes a lot of computational effort and turns out to be very time consuming. In this paper, we investigate a transfer learning approach in the context of the 2017 PlantCLEF task, for automatic plant image classification. The proposed approach is based on the well-known AlexNet Convolutional Neural Network (CNN) model. The network was fine-tuned using the 2017 PlantCLEF Encyclopedia of Life (EOL) training data, which consists of approximately 260,000 plant images belonging to 10,000 species. The learning process was sped up in the upper layers leaving original features almost untouched. Our best proposed official run scored 0,361 in terms of the Mean Reciprocal Rank (MRR) when evaluated on the test dataset.

Keywords. LifeCLEF, plant identification, deep learning, transfer learning, convolutional neural networks.

1   Introduction

Plants are one of the most vital life forms on Earth, having a tremendous contribution to the well-functioning of the ecosystems. Hence plant taxonomy plays a key role in preservation of plant species across the globe. Nevertheless, plant taxonomy is problematic and often results in duplicate identifications, given the difficulty of the classification task and the error of the human operator performing the manual classification. Therefore, the plant identification challenge of the Conference and Labs of the Evaluation Forum (CLEF) [5,6,7,8,9,10] in conjunction with initiatives such as LeafSnap [11] and Pl@ntNet [12] aim to bring together computer vision enthusiasts and botanists to exploit big amounts of raw image queries in a fully automatic way and therefore to ensure a sustainable approach when comes to ecological monitoring studies and environmental conservation.

Convolutional neural networks came to the attention of the research groups when a team led by Geoffrey Hinton and Alex Krizhevsky won the ImageNet Large Scale Visual Recognition Competition (ILSVRC) [13] with record-breaking results [1]. In

order to train the model, they used a subset of the ImageNet database. Therefore, the model was trained on more than one million images and was able to classify images into 1,000 objects categories.

Lately, a trend has been observed that industry together with research groups are using more and more in their work deep convolutional networks architectures, due to the high performance outputs. The same trend was also observed among the research groups participating in the LifeCLEF plant identification challenge with outstanding performance [2,3,4]. Likewise, transfer learning became a common practice among researchers when comes to the use of convolutional neural networks [14,15,16] mainly because fine-tuning a network is much faster and easier than training from scratch. Moreover, the pre-trained network has already learned a rich set of features that can be applied to a wider range of tasks.

PlantCLEF 2016 campaign has brought together 94 research groups. Among these only 8 research groups succeeded in submitting their runs [10]. They had to build robust plant classification systems in order to solve a multi-organ plant classification problem, i.e., identification of 1,000 species of plants corresponding to 7 different organs, along with an open-set recognition problem, i.e., automatic detection of invasive species from unknown classes.

PlantCLEF 2017 campaign, continues the challenge and aims to automatically detect in the Pl@ntNet raw query images, specimens of plants belonging to the provided training data. Another objective for this year is to evaluate the performance of a system built with noisy data against one built using trusted data. Therefore, two main training sets are provided, each being part of the same list of 10,000 plant species: a "trusted" training set based on the online collaborative Encyclopedia of Life (EoL) and a "noisy" training set built using web crawlers [20,21].

In this paper, we present the participation of the UPB HES SO team to the task. Our proposal is a transfer learning approach, adapted to plant image classification. We use a pre-trained model of the AlexNet CNN [1], accelerating the learning process in the upper layers. A number of 4 runs were sent for evaluation, each run based on the same model, but having different set of hyper parameters.

The rest of this paper is organized as follows. Section 2 describes the proposed method based on the fine-tuning of the AlexNet model for plant identification. Section 3 describes the training process, experiments and the results. Conclusions and future challenges are presented in Section 4.

2   Method Description

In order to build our plant identification system we make use of transfer learning. For that we considered a pre-trained deep neural network model which is part of the Matlab Neural Network Toolbox [17]. We employed AlexNet.

AlexNet can be split into two distinct parts, based on the role they accomplish in the neural network. The first part of AlexNet is responsible with feature learning, being comprised of five convolutional layers from which the first, the second and the fifth are followed by max-pooling layers; while the second part of AlexNet encloses

three fully-connected layers with an output layer of 1000 neurons for classification, representing each class in the neural network (see Fig. 1).

For fine-tuning of AlexNet we used the PlantCLEF 2017 Encyclopedia of Life (EOL) training set, having a base learning rate of 0.001 and a batch size of 512 without weight decay. In order to prevent the overfitting, two strategies were used: firstly, we set a threshold so that the training stops if the mean accuracy of the previous 50 iterations is greater than 99%; and secondly we imposed a L2 regularization factor of 0.001. The training process was sped up in the last fully connected layer, by multiplying the base learning rate with a factor of 10.

Also, the last fully connected layer was modified to fit our needs: its output size was increased to 10,000, representing our classes. Therefore, the network will be able to distinguish between 10,000 different species of plants.
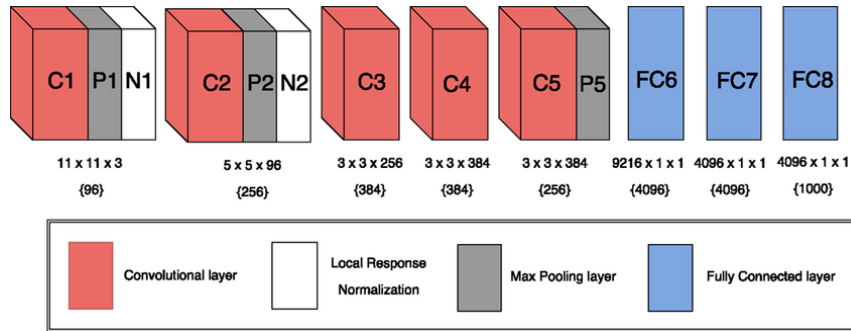

Fig. 1. An illustration of the AlexNet model. The number inside curly braces represents the number of filters with dimensions mentioned above it [19].

3   Experiments and results

This section gives an overview on experiments we have conducted for both validation of the method and for final training itself. We firstly sought for the optimal hyper parameters in order to fine-tune the neural network model by conducting short experiments on a selected group of 1,000 categories of plants. Then, we took another validation step towards final training phase by using all the 10,000 categories of plants together with the optimal set of hyper parameters. Thereafter, we have fine-tuned the network in order to obtain our final runs. These aspects will be depicted in the following sub-sections.

3.1   Preliminary experiments on development data

We have considered the PlantCLEF 2017 Encyclopedia of Life (EOL) training dataset containing about 260,000 images from which some were in grayscale and others in CMYK. We have removed all the grayscale images since most of them were representing sketches of plants, low quality images, or even maps (see Fig. 2). In this way

a total of 65 images from 45 categories were removed from the database. We also found 19 CMYK images which have been reduced to RGB, i.e., removing the 4th dimension, and further used in training. All the images were resized to 227 x 227 pixels due to the AlexNet input layer requirements.

In our experiments we have used the Matlab Neural Network Toolbox [17]. Our models were trained using Stochastic Gradient Descent with momentum 0.9 (SGDM).

Fig. 2. Examples of grayscale images (from left to right: sketch, low quality image, map).

To find the optimal hyper parameters to use for fine-tuning the network, we have selected the first 1,000 categories, in ascending order, from the training dataset. The main reason for choosing only a small sub-set from the training dataset was that in this way we could get a quick estimate of the final results by varying several hyper parameters.

Experiments thus consisted in choosing the first 1,000 categories from the training set, representing 25,094 images from which 80% (per class) were used for training and the rest of 20% (per class) were used in evaluation.

We split our preliminary experiments in two phases. In the first phase we conducted a coarse exploration of data by varying several hyper parameters: weight learn rate along with the bias learn rate multiplying factors in the last fully connected layer, batch size and threshold used in the early stopping strategy. The purpose of this step was to find the best approach which could score the highest accuracy when evaluated on validation data. In the second phase we took the winner from the previous one and started to vary the L2 regularization factor to see what contribution could bring to the performance of the network. Throughout the experiments the base learning rate was constant.

Experiments have shown that weight and bias learn rates from the last fully connected layer as well as the strategies used to prevent the overfitting have a major impact on the performance of the network. Therefore the best approach having a multiplying factor of 10 for weight and bias learn rates and a batch size of 512 has reached the early stopping threshold of 98% in 2,066 iterations, i.e., about 53 epochs, scoring 45.6% accuracy on the validation data (see Table 1).

Then, in the second phase the best approach having the same hyper parameters as the winner from the first one but with a L2 regularization factor of 0.001 has reached the early stopping threshold of 99% in 3,117 iterations, i.e., about 80 epochs, scoring 47% accuracy on the validation data (see Table 2).

We have carried out yet another preliminary experiment before we trained the network for our submission: we validated the optimal hyper parameters on the entire training set from which 80% of images (per class) were used for training and the rest

of 20% of images (per class) were used for validation. The main objective of this experiment was to give us an estimate about the network performance when trained on the entire dataset. We therefore trained the network with the optimal hyper parameters having a base learning rate of 0.001, weight and bias learning rate multiplying factor in the last fully connected layer of 10, batch size of 512 along with an early stopping threshold of 99% and L2 regularization factor of 0.001. The model has reached the threshold in 16,668 iterations, i.e., about 42 epochs, and scored 30.55% accuracy on the validation set.

Table 1. Preliminary results in the first evaluation phase. BLR denotes the base learning rate, WLR and BiLR denotes the weight, respectively the bias learn rate factor in the last fully connected layer.

No.	BLR	WLR Factor	BiLR Factor	Batch Size	Threshold	Accuracy
1	0.001	100	100	256	95%	35.59%
2	0.001	100	100	512	95%	39.96%
3	0.001	100	100	512	98%	41.38%
4	0.001	100	100	512	98%	40.56%
5	0.001	10	10	512	98%	45.67%
6	0.001	1	1	512	98%	45.18%

Table 2. Preliminary results in the second evaluation phase (note that the 3rd experiment has not reached the threshold of 99%, being stopped at 100 epochs, reaching only 97.85%).

No.	L2 Regularization	Number of Epochs	Accuracy
1	0.0005	80	45.28%
2	0.001	80	47.03%
3	0.005	100	44.46%

3.2 Results on the testing data

Following the previous experiments, we have used the above mentioned hyper parameters in order to train four models which have been submitted for evaluation. Each of these runs is detailed below as follows:

— UPB HES-SO Run 1: This is our primary submission hence is the starting point for the other runs. In this run we fine-tuned the AlexNet neural networks model using the optimal hyper parameters from the validation step without weight decay:

- Base learning rate   0.001
- Weight learn rate factor   10
- Bias learn rate factor   10

- Batch size   512
- Threshold   94%
- L2 regularization factor   0.001

This model has reached the early stopping threshold in 51,943 iterations, i.e., about 104 epochs, and scored 0.326 in terms of the Mean Reciprocal Rank (MRR) when evaluated on the test dataset. It took the 11th place amongst the models which were trained only with trusted dataset [20].

— UPB HES-SO Run 2: This model was trained for about 18 epochs using Run 1 as a starting point and therefore having the same hyper parameters but learning rate factor in the last fully connected layer was set to 20. No updates on base learning rate were done during training. It achieved a score of 0.305 in terms of MRR when evaluated on the test dataset and took the 12th place amongst the models which were trained only with trusted dataset [20].

— UPB HES-SO Run 3: In this run we wanted to see to what extent the network performance can be improved by decreasing the learning rate factor in the last fully connected layer and further by adding a learning rate schedule. We used Run 1 as a starting point with the learning rate factor in the last fully connected layer set to 5. We have trained the network for about 14 epochs without weight decay obtaining 96.88% accuracy on the training set. After that we dropped the learning rate by a factor of 0.5 each epoch until the 18th epoch having 99.22% accuracy on the training set. This model achieved a score of 0.361 in terms of MRR when evaluated on the test dataset, proving that adding a learning rate schedule within training stage could have a positive impact on the performance of the network. It took the 9th place amongst the models which were trained only with trusted dataset [20].

— UPB HES-SO Run 4: The setup was similar to Run 3 but having the learning rate factor in the last fully connected layer set to 2. We have trained the network for about 13 epochs without weight decay obtaining 97.85% accuracy on the training set. After that we dropped the learning rate by a factor of 0.5 each epoch until the 18th epoch having 99.02% accuracy on the training set. However, when evaluated on the test dataset we achieved the same score of 0.361 as for Run 3 and this model took the 10th place amongst the models which were trained only with trusted dataset [20]. In this case lowering the learning rate factor in the last fully connected layer did not bring any improvement on the performance but still this hyper parameter has a key role in transfer learning and its value cannot be neglected.

We have considered the PlantCLEF 2017 EOL training dataset, mainly because we wanted to validate our method aiming for the highest performance and we thought that this couldn't be achieved on the noisy dataset alone.

All the models were trained using a NVIDIA Quadro M4000 GPU. Training for the first run took about 7 days to complete, while for the other runs took about 27 hours, the latter being derived from the first run.

The results for these four runs are depicted within Table 3.

Table 3. Results on the testing data. The official score is in terms of MRR and the official rank represents the place obtained amongst the models which were trained only with trusted dataset.

Run	Official score	Official rank
UPB HES-SO Run1	0.326	11
UPB HES-SO Run2	0.305	12
UPB HES-SO Run3	0.361	9
UPB HES-SO Run4	0.361	10

4   Conclusions and future work

In this paper, we presented the participation of UPB HES SO team to the 2017 PlantCLEF challenge. Choosing the optimal hyper parameters when fine-tuning a pre-trained neural network model is a sensitive topic. Although transfer learning might be very convenient when comes to deep convolutional neural networks, it could lead to poor system performance if hyper parameters are wrongly selected. Another important aspect is the choice of the pre-trained model which should satisfy both the requirements of the task and the hardware limitations.

   We explored several hyper parameters in context of transfer learning to find to what extent each one would impact the performance of the network. Thereby we found that weight and bias learn rates from the last fully connected layer as well as the strategies used to prevent the overfitting influence the network's ability to adapt to new tasks. Besides early stopping threshold and L2 regularization factor, a learn rate schedule has to be considered in order to improve the network performance. Thus our system scored 0,326 MRR when trained without weight decay, respectively 0.361 MRR when the base learning rate was dropped to half every epoch.

   As previously mentioned, the chosen pre-trained model is a particularly important aspect. Therefore we believe that our method could be improved if we choose for instance one of the VGG Neural Network models [18] and also if we take earlier into account a learn rate schedule.

   We deeply understand the tremendous impact of plant taxonomy for environmental conservation. We intend to continue our research activity over the plant classification problem, complementing our approach with more advanced techniques such as data augmentation or even make use of Probabilistic Neural Networks (PNN). In this way, we aim to provide a sustainable approach when comes to ecological monitoring studies and environmental conservation.

References

1. Krizhevsky, A., Sutskever, I., Hinton, G. E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, pp. 1097-1105. (2012)
2. Hang, S. T., Tatsuma, A., Aono, M.: Bluefield (KDE TUT) at LifeCLEF 2016 Plant Identification Task. In: CLEF (Working Notes). (2016)
3. Mostafa, M. G., Berrin, Y., Erchan, A.: Open-set Plant Identification Using an Ensemble of Deep Convolutional Neural Networks. In: CLEF (Working Notes). (2016)
4. Sulc, M., Mishkin, D., Matas, J.: Very Deep Residual Networks with MaxOut for Plant Identification in the Wild. In: CLEF (Working Notes). (2016)
5. Goeau, H., Bonnet, P., Joly, A., Boujemaa, N., et al.: The CLEF 2011 plant images classification task. In: CLEF (Notebook Papers/Labs/Workshop). (2011)
6. Goeau, H., Bonnet, P., Joly, A., Boujemaa, N., et al.: The ImageCLEF 2012 plant identification task. In: CLEF (Online Working Notes/Labs/Workshop). (2012)
7. Goeau, H., Bonnet, P., Joly, A., Bakic, V., et al.: The ImageCLEF 2013 plant identification task. In: CLEF (Working Notes). (2013)
8. Goeau, H., Joly, A., Bonnet, P., et al.: LifeCLEF plant identification task 2014. In: CLEF (Working Notes). (2014)
9. Goeau, H., Bonnet, P., Joly, A.: LifeCLEF plant identification task 2015. In: CLEF (Working Notes). (2015)
10. Goeau, H., Bonnet, P., Joly, A.: Plant identification in an open-world (LifeCLEF 2016). In: CLEF working notes 2016. (2016)
11. Neeraj Kumar, Peter N. Belhumeur, Arijit Biswas, et al: Leafsnap: A Computer Vision System for Automatic Plant Species Identification. In: Proceedings of the 12th European Conference on Computer Vision (ECCV). (2012)
12. Goeau, H., Bonnet, P., Joly, A.: Pl@ntNet mobile app. In: Proceedings of the 21st ACM international conference on Multimedia, pp. 423-424. (2013)
13. Russakovsky, O., Deng, J., Su, H., et al.: ImageNet Large Scale Visual Recognition Challenge. In: International Journal of Computer Vision (IJCV). Vol 115, Issue 3 , pp. 211–252. (2015)
14. Toth, B. P, Osvath, M., Papp, D., Szucs, G.: Deep Learning and SVM Classification for Plant Recognition in Content-Based Large Scale Image Retrieval. In: CLEF (Working Notes). (2016)
15. Lee, S. H., Chang, Y. L., Chan, C. S., Remagnino, P.: Plant Identification System based on a Convolutional Neural Network for the LifeCLEF 2016 Plant Classification Task. In: CLEF (Working Notes). (2016)
16. McCool, C., Ge, Z., Corke, P.: Feature Learning via Mixtures of DCNNs for Fine-Grained Plant Classification. in: CLEF (Working Notes). (2016)
17. Matlab Neural Network Toolbox,
    URL: https://www.mathworks.com/products/neural-network.html
18. Simonyan, K., Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition. In: CVPR. (2014)
19. Srinivas, S., Sarvadevabhatla, R. K., Mopuri, K. R. et al.: A Taxonomy of Deep Convolutional Neural Nets for Computer Vision. In: Vision Systems Theory, Tools and Applications, a section of the journal Frontiers in Robotics and AI. (2016)
20. Goeau, H., Bonnet, P., Joly, A.: Plant identification based on noisy web data: the amazing performance of deep learning (LifeCLEF 2017). In: CLEF working notes 2017. (2017)

21. Joly, A., Goeau, H., Glotin, H., et al.: LifeCLEF 2017 Lab Overview: multimedia species identification challenges, CLEF 2017 Proceedings, Springer Lecture Notes in Computer Science (LNCS). (2017)