Improving Model Performance for Plant Image Classification With Filtered Noisy Images
FHDO Biomedical Computer Science Group (BCSG)

Andreas R. Ludwig[1], Helga Piorek[1], Andreas H. Kelch[1], David Rex[1], Sven Koitka[1,2], and Christoph M. Friedrich[1]

[1] University of Applied Sciences and Arts Dortmund (FHDO)
Department of Computer Science
Emil-Figge-Strasse 42,
44227 Dortmund, Germany
http://www.inf.fh-dortmund.de,
andreasralf.ludwig003@stud.fh-dortmund.de, h.piorek@stud.fh-dortmund.de,
andreas.kelch001@stud.fh-dortmund.de, darex@live.de,
sven.koitka@fh-dortmund.de, christoph.friedrich@fh-dortmund.de
[2] TU Dortmund University Department of Computer Science
Otto-Hahn-Str. 14,
44227 Dortmund, Germany

Abstract. The training of convolutional neural networks for image recognition usually requires large image datasets to produce favorable results. Those large datasets can be acquired by web crawlers that accumulate images based on keywords. Due to the nature of data in the web, these image sets display a broad variation of qualities across the contained items. In this work, a filtering approach for noisy datasets is proposed, utilizing a smaller trusted dataset. Hereby a convolutional neural network is trained on the trusted dataset and then used to construct a filtered subset from the noisy datasets. The methods described in this paper were applied to plant image classification and the created models have been submitted to the PlantCLEF 2017 competition.

Keywords:
convolutional neural networks, plant image classification, plantCLEF, oversampling, transfer learning, filtering noisy datasets

1 Introduction

The LifeCLEF plant identification task (PlantCLEF) [5,7] is an annual competition and part of the LifeCLEF evaluation campaign.
This year's PlantCLEF task dataset was comprised of 10,000 classes but contained only up to around 1,250 samples per class. Some classes were only represented by a handful of samples. There was a second, noisy dataset, which was derived from results of Bing and Google search queries. These results are partly representing images other than plants or have been labeled incorrectly.

Up to now, image classifiers have rarely been trained with datasets with more than 1,000 classes. In most of these cases, such datasets were only used to test the effectiveness and results of large distributed learning systems. In [11] a deep autoencoder that was trained on an ImageNet-10K [15] dataset comprising 10,000 classes reached a top-1 accuracy of 19.2 %. It was also tested on the ImageNet-21K dataset comprising 21,841 classes, where it reached a top-1 accurary of 15.8 % . These results were later improved by [4], where a top-1 accuracy of 29.8 % on ImageNet-21K was reached with Alexnet [9]. With Nearest Class Mean classification a top-1 accuracy of 23.9 % has been achieved by [12] on the ImageNet-10K dataset. By using fisher vectors [13] a top-1 accuracy of 19.1 % was achieved by [16] on the same dataset.

This paper aims to show that modern architectures for convolutional neural networks, such as InceptionV4 or InceptionResNetV2 [17], can achieve state-of-the-art results on the given dataset. Furthermore it was assessed if the training results can be improved by training a subset of the noisy dataset. The subset was created by filtering the noisy dataset with an already trained neural network.

As seen in [2,3,14] transfer-learning can improve the training results of neural networks. Therefore the models in this work were trained by an approach similar to the two phases fine-tuning approach described in [3]. At the beginning, the weights of the output layer were randomly initialized and then trained with a small learning rate for a few epochs. Afterward, the entire network was trained. The methodologies of this paper are further described in Section 2. The training of the neural networks as well as the overall results of this paper are described in Section 3. The conclusion can be found in Section 4.

2 Methodology

2.1 Dataset Split

In order to be able to evaluate the quality of the created classifiers, the data of the Encyclopedia of Life (EOL) dataset were randomly split into a training and a validation set. The training set consisted of 90 % of the EOL data and the validation set consisted of the remaining 10 %. In this way, the pretrained models' performance could be estimated during development. For the final submission, the models were trained on the complete EOL dataset in order to take advantage of the full training set.

2.2 Regular Training of the Model

The general approach to the task at hand was to utilize pretrained models provided on the Tensorflow Slim Git page[3] as a starting point for the training. Since the utilized models have been trained to recognize one thousand different classes, many of which are irrelevant for the PlantCLEF competition, these

[3] Maintainer: Nathan Silberman and Sergio Guadarrama; [last access: 29.06.2017] https://github.com/tensorflow/models/tree/master/slim

models were fine-tuned using the training set. As a result of the fine-tuning, there were models produced containing only the lower level filters of the pretrained model and output layers primed to the 10,000 classes of the PlantCLEF competition. Subsequent to the fine-tuning process, the models were trained further utilizing the training set and filtered web datasets. The general workflow is displayed in Figure 1.

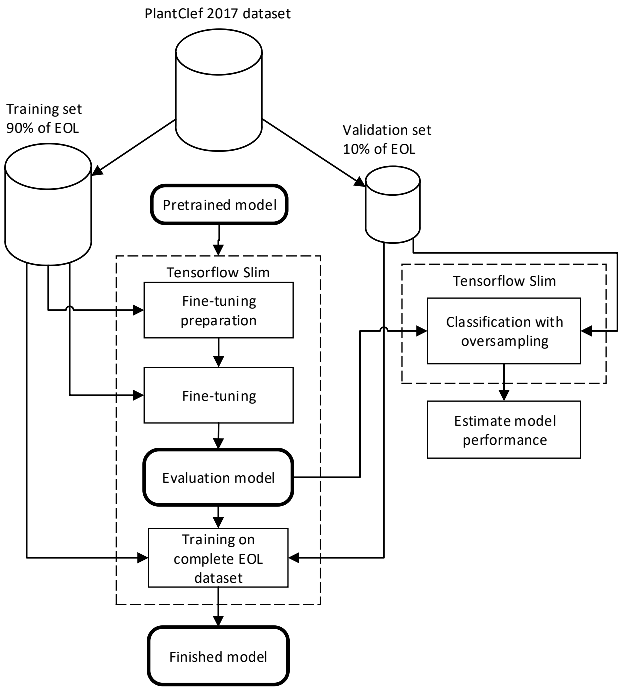

Fig. 1. The diagram shows the general workflow for Run 1. The EOL dataset was split in a training and a validation set. A pretrained InceptionResNetV2 checkpoint was fine-tuned on the training set and evaluated using the validation set. The training was finished by training the network with the entire EOL dataset.

2.3 Utilized CNN Architectures

The pretrained models on the Tensorflow Slim Git page were trained on the ILSVRC-2012-CLS dataset [15]. The model checkpoints were published with the accuracy that could be reached on the corresponding test set. Since the InceptionResNetV2 and InceptionV4 [17] architectures are evidently able to produce the best accuracies, 80.4 % top-1 for InceptionResNetV2 and 80.2 % top-1 for

InceptionV4. The training was conducted with a fine-tuning procedure similar to the two-step fine-tuning approach described in [3]. The standard TF-slim data augmentation[4] functions were used for training.

2.4 Filtered Web Datasets

After visually examining the web dataset it became apparent, that apart from plant images in varying qualities, there were also a large number of images that displayed unrelated items such as postage stamps or music instruments.

Due to the noisy nature of the web dataset, the data were filtered to reduce the possibility that the classification results of the existing models are weakened. For this purpose, the whole web dataset was classified using a model that showed a good estimate performance. This model has been trained on the whole EOL set and was submitted as Run 1. Ultimately all the pictures that could be classified correctly within top-5 predictions were compiled to a web-top-5 dataset containing 556,584 samples. One could assume that using top-5 web subset for further training improves the accuracy of the trained model solely with regard to the classes that have already been learned sufficiently. It would be interesting to investigate, if filtering different proportions of the predicted classes (e.g. top-100) could improve the accuracy. However due to limitations of processing time only the filtering approach described above was pursued. Figure 2 visualizes the procedure used for filtering the web dataset.

The plots in Figure 3 show the number of occurrences of distinct classes within the training dataset and the filtered dataset sorted in descending order. The number of occurrences for every class was logarithmized. The filtered subset is missing any samples for 696 classes, which is a general problem of the presented filtering approach. Training data for classes that are already well trained can be selected from the noisy dataset, but all the images of classes that can not be classified within a margin will be removed along with the noise. The objective of further research would be to find a top-x which maximizes noise filtering without dropping too many classes.

2.5 Oversampling

To improve the results of the classification, multiple crops of an image were used to calculate an average classification per sample, as described in [6,8,17]. For each sample, ten crops were created, one at each corner, one centered and a mirrored version of these five crops. The final prediction was estimated as an average of the ten samples of every image.

3 Results

The estimated performance metrics were calculated using models that had not yet been trained with the complete EOL dataset. These models were used to

[4] Maintainer: Nathan Silberman and Sergio Guadarrama; [last access: 29.06.2017] https://github.com/tensorflow/models/tree/master/slim

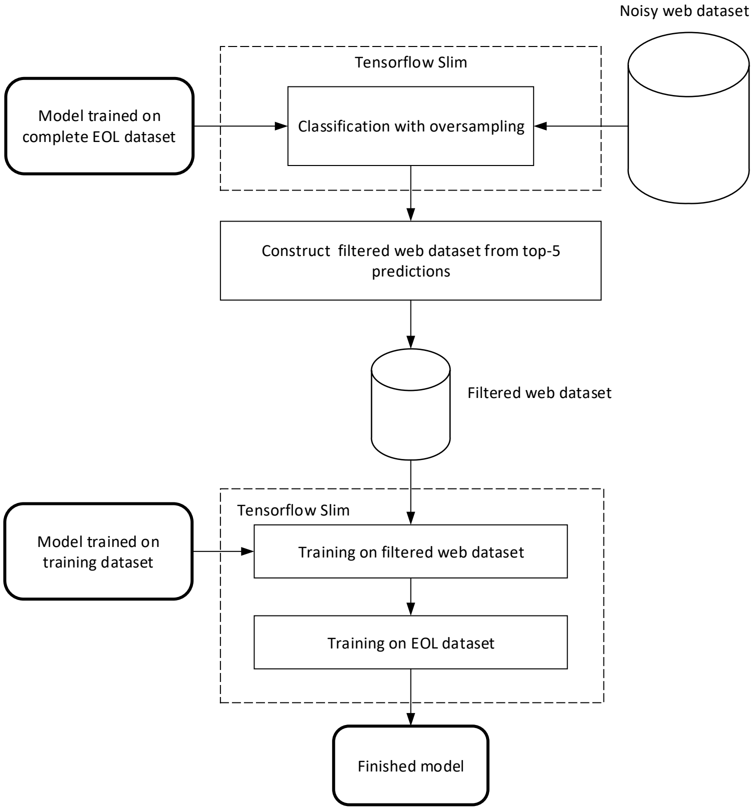
Noisy web dataset
Tensorflow Slim
Model trained on complete EOL dataset
Classification with oversampling
Construct filtered web dataset from top-5 predictions
Filtered web dataset
Tensorflow Slim
Model trained on training dataset
Training on filtered web dataset
Training on EOL dataset
Finished model

Fig. 2. The diagram shows the filtering procedure. At first the noisy dataset is classified using the finished Run 1 model. Afterwards the images that could be classified correctly within top-5 predictions were accumulated to a filtered web subset. This web subset was then used for further training

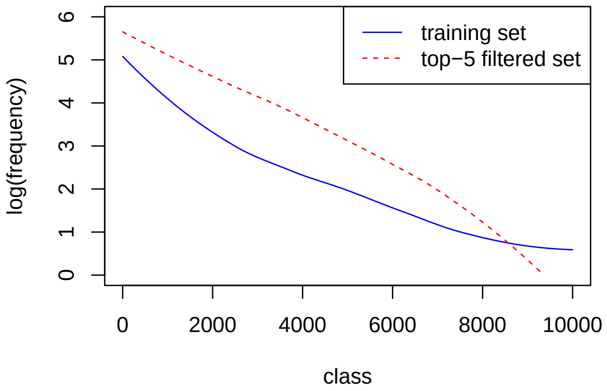

Fig. 3. Number of samples per class in training dataset and the top-5 filtered dataset sorted by the number of samples per class. The number of samples per class is logarithmized

receive an estimate of the performance through classification on the validation set. It is important to notice that in the EOL dataset, every image belongs to a unique observation. In contrast, the official test set contains a large number of observations with multiple images, allowing the model to predict the label of an image based on an average of multiple images. The official MRR score was calculated based on observations, whereas the presented validation MRR scores were estimated on images solely. Table 1 shows the training error of the finished models and the ensemble. The model performance and accuracy of the first three runs are presented in Table 2. Run 4 was excluded from both Tables as it is considered to be broken.

3.1 FHDO BCSG Run 1 - InceptionResNetV2 Trained on EOL Dataset

The first submission was trained on the aforementioned pretrained model. For the fine-tuning training step, the logits and auxiliary logits were randomly initialized instead of being copied from the utilized checkpoint. These two layers had been trained using a relatively small learning rate of 0.0045 for five epochs. The two layers were chosen based on the assumption that the lower level layers have already been suitably prepared for classification tasks while the topmost layers needed to be primed onto the 10,000 different classes of the PlantCLEF competition.

Subsequently, all the layers of this model were trained on the training set for fifty epochs with the parameters shown in Table 3. The model that was trained

Table 1. Classification results on the training datasets using the finished models

Datasets	**Run ID**	**MRR**	**Top-1**	**Top-5**
Training (90 %)	Run 1	0.898	0.856	**0.949**
	Run 2	0.894	0.854	0.941
	Run 3	**0.900**	**0.860**	0.946
Validation (10 %)	Run 1	0.509	0.418	0.611
	Run 2	**0.515**	**0.426**	**0.615**
	Run 3	0.513	0.423	0.614
EOL	Run 1	**0.846**	**0.800**	**0.901**
	Run 2	0.835	0.789	0.889
	Run 3	0.841	0.795	0.895

Table 2. Official scores on the test set and estimated performance and accuracies on training and validation set of Run 1-3

Run ID	Estimated performance						Official score
	Training			Validation			MRR
	MRR	Top-1	Top-5	MRR	Top-1	Top-5	
Run 1 EOL	0.889	0.845	**0.942**	0.471	0.382	0.570	0.792
Run 2 EOL + Filtered	0.887	0.845	0.939	**0.503**	**0.421**	**0.594**	**0.806**
Run 3 Ensemble	**0.890**	**0.848**	0.941	0.493	0.406	0.588	0.804

Table 3. Training parameters for Run 1 and 2

	Run 1	**Run 2**
Mini-batch size	32	32
Steps	50 epochs (334,700 steps)	30 epochs (200,820 steps)
Learning rate	0.045	0.045
Optimizer	rmsprop	rmsprop
Learning rate decay type	exponential	exponential
Learning rate decay factor	0.47	0.47
Number of epochs per decay	2	2
Weight decay	0.00004	0.00004

for fifty epochs yielded an MRR of 0.4661 on the validation set and was therefore chosen as the model for the first submitted run. To finish the training the net was trained for another five epochs using the complete EOL dataset. Table 4 shows the effect of oversampling on the estimated MRR scores for Run 1.

Table 4. Run 1: The results of classification on the validation set with one central crop in comparison to oversampling

Crop method	**Estimated performance**		
	MRR	Top-1	Top-5
Central Crop	0.466	0.375	0.567
Oversampling	**0.471**	**0.382**	**0.570**

3.2 FHDO_BCSG Run 2 - InceptionResNetV2 Trained on EOL and Web-Top-5 Datasets

The second submission was trained analogously to the first submission. As mentioned before, a pretrained model was used and only the logits and auxiliary logits layer were trained for five epochs with a small learning rate of 0.0045. Eventually, all layers of the model were trained for thirty epochs using the parameters shown in Table 3.

In order to examine the effect of the web filtering approach, a web subset was created using the completely trained Run 1 model. Images were added to this set if the annotated class of the noisy training set was in the top-5 predictions of the model trained for Run 1. With this, a web subset finally consisting of 556,584 images was assembled. Following the filter process, the Run 2 model was trained on the web-top-5 dataset for five epochs and afterwards trained for another five epochs on the training dataset. The parameters for this training were chosen analogously to the training of Run 1 with a starting learning rate of 0.000275 and five epochs. The learning rate was adopted from the last training epoch of the previous training step. To finish up the training for this model, the net was trained for another five epochs on the complete EOL dataset analogue to the procedure used for Run 1. Before executing the final training step, the MRR scores were estimated on the validation set, leading to a result of 0.503. This value could have been biased though, due to the web-top-5 subset being compiled with a network that was trained on the complete EOL dataset. Table 5 shows the effects of oversampling on the estimated MRR scores of Run 2. Oversampling improved the MRR scores slightly at the cost of higher computing demands.

In order to investigate the effectiveness of the filtering approach, the training procedure of Run 2 was modified and two more models were trained. The starting point for both models was an InceptionResNetV2 trained on the training set for thirty epochs. The training for the first evaluation model was conducted once again with 556,584 images from the web dataset. Instead of being selected by the filtering approach, they were chosen randomly. The training for the second evaluation model was conducted with the complete web dataset. The first model was trained for five epochs (86,966 steps) and the other model with an equivalent number of steps (86,966 steps). The training parameters were chosen analogously to the training of the Run 2 model. Afterwards the models were trained on the training dataset for another five epochs. The results on the validation set showed that using the filtered data for training improved the estimated MRR scores compared to using a randomly drafted dataset or the whole noisy dataset. The results of this analysis are displayed in Table 6.

Table 5. Run 2: The results of classification with one central crop in comparison to classifaction with oversampling on the validation set

Crop method	**Estimated performance**		
	MRR	Top-1	Top-5
Central Crop	0.492	0.411	0.582
Oversampling	**0.503**	**0.421**	**0.594**

Table 6. The results of training with a filtered dataset in comparison to training with a randomly drafted dataset or the whole web dataset. The scores were calculated on the validation set

Run variations	**MRR**	**Top-1**	**Top-5**	**Training steps**
Run 2	**0.503**	**0.421**	0.594	86,966
Random images	0.487	0.397	0.588	86,966
Whole web dataset	0.495	0.402	**0.598**	86,966

3.3 FHDO_BCSG Run 3 - Ensemble of Run 1 and 2

Composing an ensemble run from multiple prediction models can improve the overall accuracy in comparison to a single prediction model [10]. In order to create an ensemble, the mean values of all the class predictions of the different models for every image were calculated and then assembled into a new set of predictions as shown in [10]. The reasoning behind this is that one model might be better at predicting certain classes while being worse than the other models on other classes and vice versa. Presumably, the ensemble would be better if it was composed from a number of accurate and diverse runs.

Due to the fact, that on the validation set, Run 2 was producing the higher MRR value of 0.503 compared to the value of 0.471 of Run 1, the models were weighted for the ensemble. In this way the presumably better model would have a higher influence on the final prediction. Run 1 contributed 1/3 and Run 2 contributed 2/3 to Run 3.

3.4 FHDO_BCSG Run 4

Run 4 was based on an InceptionV4 architecture but did only achieve modest estimated MRR scores due to misconfigurations.

4 Conclusion

The utilized filtering approach improved the predictions of the resulting model on the validation set and the official score. The filtering approach increased computation costs. Recent studies suggest [1], that neural networks can only recognize samples of unknown classes to a certain extent. Since there were only few samples available for some classes and since some classes were very similar to one another, there is a chance that samples belong to a known class other than the one they were labeled with.

The use of oversampling leads to a minimal increase in estimated MRR scores, as shown in [17]. Since multiple crops increase the processing time during classification, the oversampling method is not suited for every scenario.

Acknowledgement

The authors gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X Pascal GPU, which supported this research.

References

1. Abhijit Bendale and Terrance E. Boult: Towards Open Set Deep Networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016). pp. 1563 – 1572 (2016)
2. Agrawal, P., Girshick, R., Malik, J.: Analyzing the Performance of Multilayer Neural Networks for Object Recognition. In: Proceedings of the 13th European Conference Computer Vision (ECCV 2014), Zurich, Switzerland, September 6-12, 2014, Part VII. pp. 329–344. Springer International Publishing (2014), http://dx.doi.org/10.1007/978-3-319-10584-0_22
3. Branson, S., Horn, G.V., Belongie, S.J., Perona, P.: Bird species categorization using pose normalized deep convolutional nets. In: CoRR. vol. abs/1406.2952 (2014), http://arxiv.org/abs/1406.2952
4. Chilimbi, T., Suzue, Y., Apacible, J., Kalyanaraman, K.: Project Adam: Building an Efficient and Scalable Deep Learning Training System. In: Proceedings of the 11th USENIX Symposium on Operating Systems Design and Implementation (OSDI 2014). pp. 571–582. USENIX Association, Broomfield, CO (2014)

5. Goëau, H., Bonnet, P., Joly, A.: Plant identification based on noisy web data: the amazing performance of deep learning (LifeCLEF 2017). In: Working Notes of CLEF 2017 - Conference and Labs of the Evaluation Forum, Dublin, Ireland, 11-14 September, 2017. CEUR-WS Proceedings Notes (2017)
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016) (2016)
7. Joly, A., Goëau, H., Glotin, H., Spampinato, C., Bonnet, P., Vellinga, W.P., Lombardo, J.C., Planqué, R., Palazzo, S., Müller, H.: LifeCLEF 2017 Lab Overview: multimedia species identification challenges. In: Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the 8th International Conference of the CLEF Association, CLEF 2017, Dublin, Ireland, September 11-14, 2017. Lecture Notes of Computer Science (LNCS), vol. 10456 (2017)
8. Koitka, S., Friedrich, C.M.: Optimized Convolutional Neural Network Ensembles for Medical Subfigure Classification. In: Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the 8th International Conference of the CLEF Association, CLEF 2017, Dublin, Ireland, September 11-14, 2017. Lecture Notes of Computer Science (LNCS), vol. 10456. Springer Verlag (2017)
9. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet Classification with Deep Convolutional Neural Networks. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 25, pp. 1097–1105. Curran Associates, Inc. (2012)
10. Kuncheva, L.: Combining Pattern Classifiers: Methods and Algorithms. Wiley, 2 edn. (2014)
11. Le, Q., Ranzato, M., Monga, R., Devin, M., Chen, K., Corrado, G., Dean, J., Ng, A.: Building high-level features using large scale unsupervised learning. In: Langford, J., Pineau, J. (eds.) Proceedings of the 29th International Conference on Machine Learning (ICML 2012). pp. 81–88. New York, USA (July 2012)
12. Mensink, T., Verbeek, J., Perronnin, F., Csurka, G.: Distance-Based Image Classification: Generalizing to New Classes at Near-Zero Cost. IEEE Transactions on Pattern Analysis and Machine Intelligence 35(11), 2624–2637 (Nov 2013)
13. Perronnin, F., Akata, Z., Harchaoui, Z., Schmid, C.: Towards Good Practice in Large-Scale Learning for Image Classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2012). pp. 3482–3489. IEEE (Jun 2012)
14. Reyes, A.K., Caicedo, J.C., Camargo, J.E.: Fine-tuning deep convolutional networks for plant recognition. In: Cappellato, L., Ferro, N., Jones, G.J.F., SanJuan, E. (eds.) CLEF (Working Notes). CEUR Workshop Proceedings, vol. 1391. CEUR-WS.org (2015)
15. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV) 115(3), 211–252 (2015)
16. Sánchez, J., Perronnin, F., Mensink, T., Verbeek, J.: Image Classification with the Fisher Vector: Theory and Practice. International Journal of Computer Vision 105(3), 222–245 (2013), http://dx.doi.org/10.1007/s11263-013-0636-x
17. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.A.: Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. In: International Conference on Learning Representations 2016 Workshop (ICLR 2016) (2016)