Learning with Noisy and Trusted Labels for Fine-Grained Plant Recognition

Milan Šulc and Jiří Matas

Center for Machine Perception, Dept. of Cybernetics, Faculty of Electrical Eng.,
Czech Technical University in Prague, Czech Republic
{sulcmila,matas}@cmp.felk.cvut.cz

Abstract. The paper describes the deep learning approach to automatic visual recognition of 10 000 plant species submitted to the PlantCLEF 2017 challenge. We evaluate modifications and extensions of the state-of-the-art Inception-ResNet-v2 CNN architecture, including maxout, bootstrapping for training with noisy labels, and filtering the data with noisy labels using a classifier pre-trained on the trusted dataset. The final pipeline consists of a set of CNNs trained with different modifications on different subsets of the provided training data. With the proposed approach, we were ranked as the third best team in the LifeCLEF 2017 challenge.

1 Introduction

The plant identification challenge PlantCLEF 2017 [1] is a part of the LifeCLEF activity [2] organized within CLEF 2017 – The Conference and Labs of the Evaluation Forum. The task of the challenge is automatic plant identification using computer vision. A similar task has been the subject of previous challenges [3,4], yet PlantCLEF 2017 aims at a significantly larger scale: recognizing plants from 10 000 species.

Two sets of training data, with different properties and sources but both covering the same 10 000 plant species, were provided by the organizers:

1. A set based on the online collaborative Encyclopedia Of Life (EoL) containing 256 287 images and corresponding xml files with meta-information. An important field in the meta-information is the "Observation ID", which is an identifier connecting images of the same specimen (object of observation). This dataset is considered "trusted", i.e. the ground truth labels should allbe assigned correctly.
2. A noisy training set built using web crawlers, or more precisely, obtained by google and bing image search. It thus contains images not related to the given plant species. This set is provided in the form of a list of more than 1442k image URLs. We obtained nearly 1405k images from the list, the remaining images failed to download.

The evaluation is performed on a test set containing 25 170 images of 13 471 observations (specimen).

The rest of the paper is structured as follows: the deep learning approach and all proposed modifications are described in Section 2. Preliminary experiments are described and their evaluation is discussed in Section 3. Post-processing steps are described in Section 4. The runfiles submitted to PlantCLEF are listed in 5. Conclusions are drawn in Section 6.

2 The Proposed Methods

In recent years, Deep Convolutional Neural Networks (CNNs) have become the core of state-of-the-art solutions of many computer vision tasks, especially those related to recognition and detection of objects. This is also the case for plant recognition, where in previous PlantCLEF challenges 2015 [4] and 2016 [5,3] the deep learning submissions [6,7,8,9,10,11,12] outperformed combinations of hand-crafted methods significantly.

2.1 Inception-ResNet-v2

The submitted model is based on the state-of-the-art convolutional neural network architecture, the Inception-ResNet-v2 model [13] which introduced residual Inception modules, i.e.inception modules with residual connections. Both the paper [13] and our preliminary experiments show that this network architecture leads to superior results compared with other state-of-the-art CNN architectures. The publicly available[1] Tensorflow model pretrained on ImageNet was used for initial values of network parameters. The main hyperparameters were set as follows:

Optimizer	RMSProp with momentum 0.9 and decay 0.9.
Weight decay	0.00004.
Learning rate	Starting LR 0.01, decay factor 0.94, exponential decay, ending LR 0.0001.
Batch size	32.

2.2 MaxOut

We experimented with adding maxout to the end of the network, which was helpful in our submission to PlantCLEF 2016: an additional fully-connected (FC) layer was added on top of the network, before the classification FC layer. The activation function in the added layer is maxout [14], maximum over slices of the layer:

$$h_i(x) = \max_{j \in [1,k]} z_{ij}, \tag{1}$$

[1] https://github.com/tensorflow/models/blob/master/slim/README.md#pre-trained-models

where $z_{ij} = \mathbf{x}^{\mathrm{T}}\mathbf{W}_{..ij} + b_{ij}$ is a standard FC layer with parameters $W \in \mathbb{R}^{d\times m\times k}$, $b \in {}^{m\times k}$.

One can understand maxout as a piecewise linear approximation to a convex function, specified by the weights of the previous layer. This is illustrated in Figure 1.

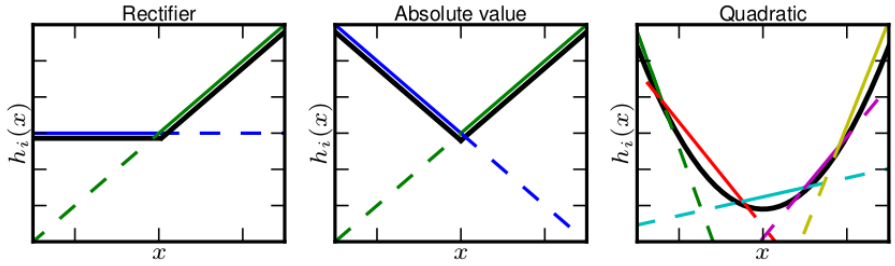

Fig. 1: Illustration of maxout from Goodfellow et al. [14].

We added a FC layer with 4096 units. The maxout activation operates over $k = 4$ linear pieces of the FC layer, i.e. $m = 1024$. Dropout with a keep probability of 80% is applied before the FC layers. The final layer is a 10000-way softmax classifier corresponding to the number of plant species needed to be recognized.

We observed is that the additional FC layer has to be batch normalized [15]. Without normalization, the architecture becomes unstable with the default setting of hyperparameters, leading to unexpected drop in accuracy.

2.3 Bootstrapping

In order to improve learning from noisy labels, Reed et. al. [16] proposed a simple consistency objective, which does not require an explicit information about the noise distribution.

Intuitively, the new objective(s) takes into account the current predictions of the network, lowering the damage done by incorrect labels. Reed al. propose two variants of the objective, denoted as Bootstrapping for consistency in multi-class prediction:

– **soft bootstrapping** uses the probabilities q_k estimated by the network (softmax):

$$L_{\mathrm{soft}}(\mathbf{q}, \mathbf{t}) = \sum_{k=1}^{N} [\beta t_k + (1-\beta) q_k] \log q_k \tag{2}$$

Reed et al. [16] point out that the objective is equivalent to softmax regression with minimum entropy regularization, which was previously studied in [17]; encouraging high confidence in predicting labels.

– **hard bootstrapping** uses the strongest prediction $z_k = \begin{cases} 1 \text{ if } k = \text{argmax} q_i \\ 0 \text{ otherwise} \end{cases}$

$$L_{\text{hard}}(\mathbf{q}, \mathbf{t}) = \sum_{k=1}^{N} [\beta t_k + (1 - \beta) z_k] \log q_k \tag{3}$$

The experiments of [16] show that the two objectives improve learning in the case of label noise, achieving the best accuracy with hard bootstrapping. We decided to follow the result of [16] and use hard booststrapping with $\beta = 0.8$ in our experiments. The search for the optimal value of β was ommited for computational reasons and limited time for the competition, yet the dependence between the amount of label noise and the optimal setting of hyperparameter β is an interesting topic of future work.

3 Experiments

We used a subset of the test data from the previous year's PlantCLEF 2016 challenge to thoroughly evaluate the proposed methods. We only used 2583 images from the previous year dataset, for which we found species-correspondences in the 2017 task. This small validation set covers only a small subset of the classes, but should be sufficient for an approximate evaluation of the method.

The sections below describe the experiments and corresponding design choices:

3.1 Fine-tuning vs. Training from Scratch

The first issue tested was whether the network should be trained from scratch, or fine-tuned from an ImageNet-pretrained model. We compared the two scenarios by training only on the "trusted" dataset. As illustrated in Figure 2, training from scratch converges very slow. After 150k iterations (mini-batches) fine-tuning leads to 65.1% accuracy, while training from scratch only gets to 44.5%. For illustration 150k training iterations take ca. 65 hours on an NVIDIA Titan X GPU. Therefore we decided for fine-tuning.

3.2 Training on Trusted and Noisy Data

We fine-tuned the system with different settings described in Section 2 on the "trusted" (EOL) data only, as well as on the combination of both "trusted" and "noisy" data (EOL+WEB). The soft- and hard- bootstrapping were used for training with "noisy" data. Figure 3 shows that after 200k iterations, the networks trained only on the "trusted" data performed slightly better. The two best performing networks trained on the "trusted" (EOL) dataset will be used in the follow-up experiments.

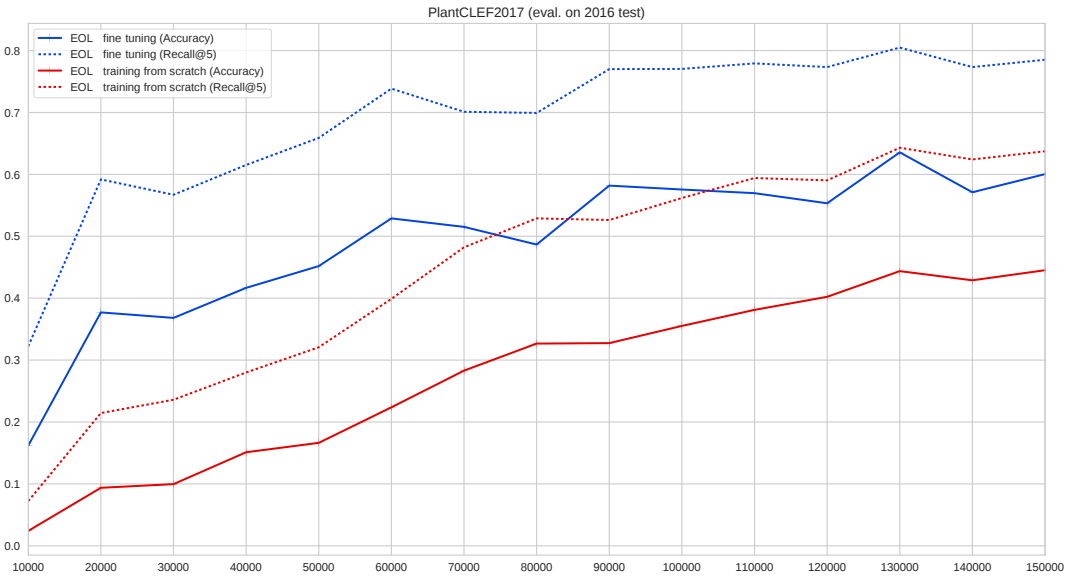

Fig. 2: Accuracy (solid) and recal@5 (dotted) when fine-tuning (red) and training from scratch (blue).

3.3 Filtering the Noisy Data and Further fine-tuning

In order to filter out wrongly labeled examples from the "noisy" part of the training set, we used the network pretrained on the "trusted set" (from Section 3.2) to predict the labels from images. Only images, where the network prediction was equal to the label were kept in the "filtered noisy" dataset. This reduced the size of the "noisy" set from ca 1405k images to ca 425k images.

Let us denote the two networks fine-tuned on the "trusted" (EOL) dataset in Section 3.2 as follows:

- **Net #1:** Fine-tuned on "trusted" (EOL) set without maxout for 200k iterations.
- **Net #2:** Fine-tuned on "trusted" (EOL) set with maxout for 200k iterations.

Further fine-tuning was performed from these models pre-trained (fine-tuned) on the "trusted" set. In order to perform bagging from several networks, we divide the data into 3 disjoint folds. Then each setting is used to further fine-tune three networks, each on different 2 of the 3 folds. Each network is further fine-tuned for 50k iterations.

- **Net #3,#4,#5:** Fine-tuned from #1 for 50k iterations on the "trusted" dataset.
- **Net #6,#7,#8:** Fine-tuned from #2 for 50k iterations on the "trusted" dataset, with maxout.

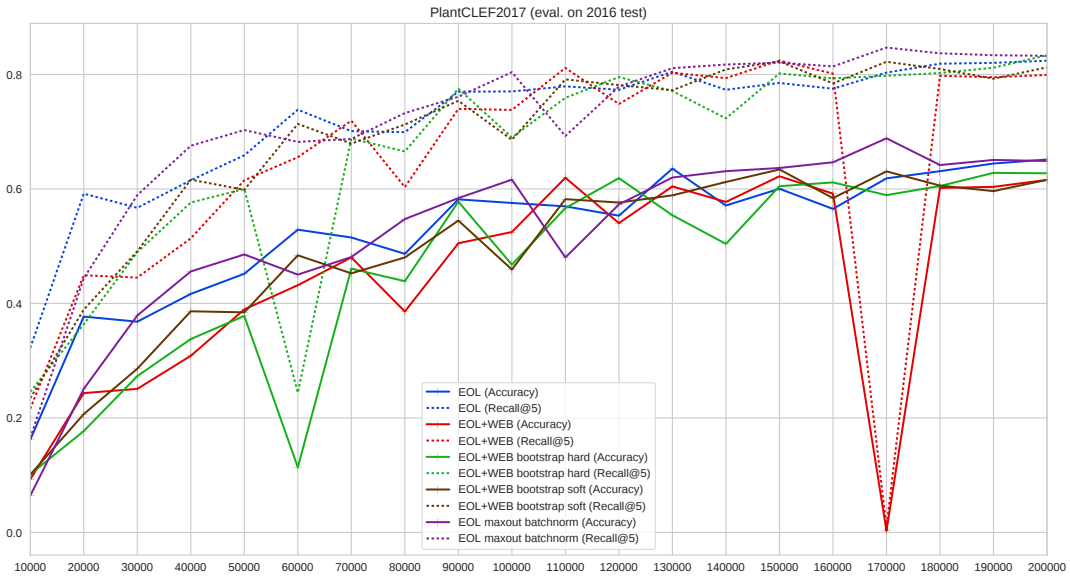

Fig. 3: Accuracy (solid) and recal@5 (dotted) for different settings.

- **Net #9,#10,#11:** Fine-tuned from #1 for 50k iterations on the "trusted" and filteret noisy data.
- **Net #12,#13,#14:** Fine-tuned from #1 for 50k iterations on the "trusted" and filteret noisy data, with hard bootstrapping.
- **Net #15,#16,#17:** Fine-tuned from #2 for 50k iterations on the "trusted" and filteret noisy data, with maxout.

Figure 4 shows the validation of the further fine-tuning. Although there are certain differences, all the networks (listed below) are quite precise, yet do not individually bring much improvement compared to the networks from Section 3.2. The strength here is in combination of the differently fine-tuned networks. the red dashed line in 4 shows the final accuracy (after 50k it. of fine-tuning) of their combination.

4 Post Processing on the Test Set

4.1 Averaging predictions per observation

As shown by the previous year's challenge winner [12] and confirmed by the experiments described in this report, averaging the predictions over images of the same observation (specimen) increases accuracy significantly. Therefore we also average scores per observations in all submitted runfiles.

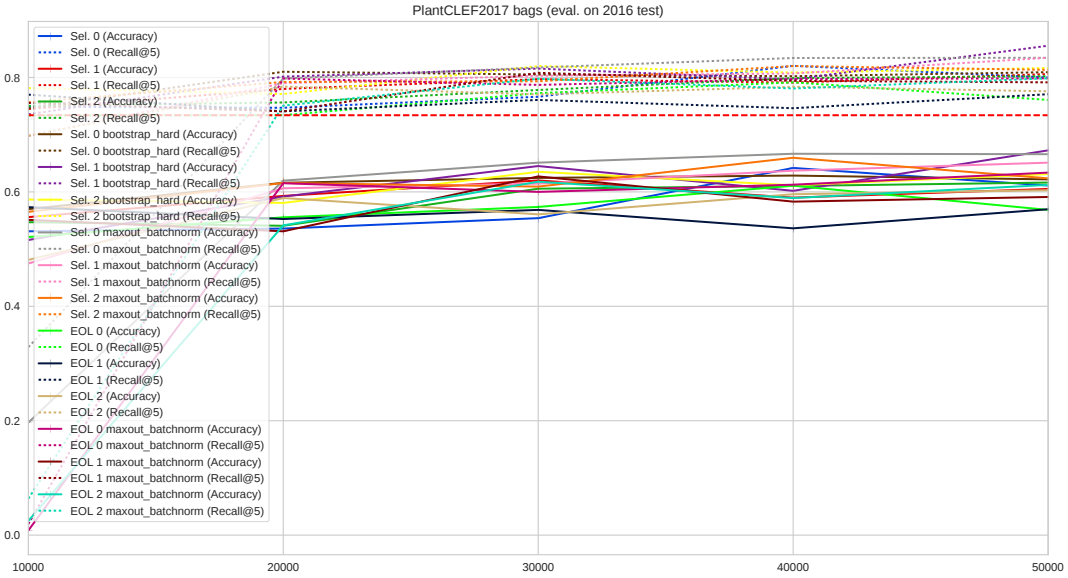

Fig. 4: Accuracy (solid) and recal@5 (dotted) for further fine-tuning using different settings.

4.2 Adjusting Test Set Prediction Distribution

Given the fact that we are evaluating the whole test set of images, we decided to experiment with adjusting the prediction distribution over the test set. Some plant species are certainly much rarer to observe than other. We assumed that the species in the test set might not follow the same distribution as the species in the training set. We computed the prior $p(K)$ for each class K among the observations in the "trusted" dataset, and estimated the prior $p_t(K)$ of on the test set. Let $q(K|X)$ be the prediction confidence for class K, given input image X. The final prediction taking into account the possible shift in the distributions was:

$$q^*(K|X) = q(K|X)\sqrt{\frac{p(K)}{p_t(K)}}, \tag{4}$$

where the square root is used to make the adjustment less severe.

5 Description of the Submitted Runfiles

In PlantCLEF 2017, each participant is allowed to submit up to four runfiles with the results. We submitted the following run files:

- *CMP Run 1* combines all 17 networks by summimg their results.
- *CMP Run 2* uses the prediction distribution adjustment from Section 4.2 on top of the results from the first runfile.

– *CMP Run 3* combines only networks trained on the "trusted" data.
– *CMP Run 4* again adds the prediction distribution adjustment on top of results from the third runfile.

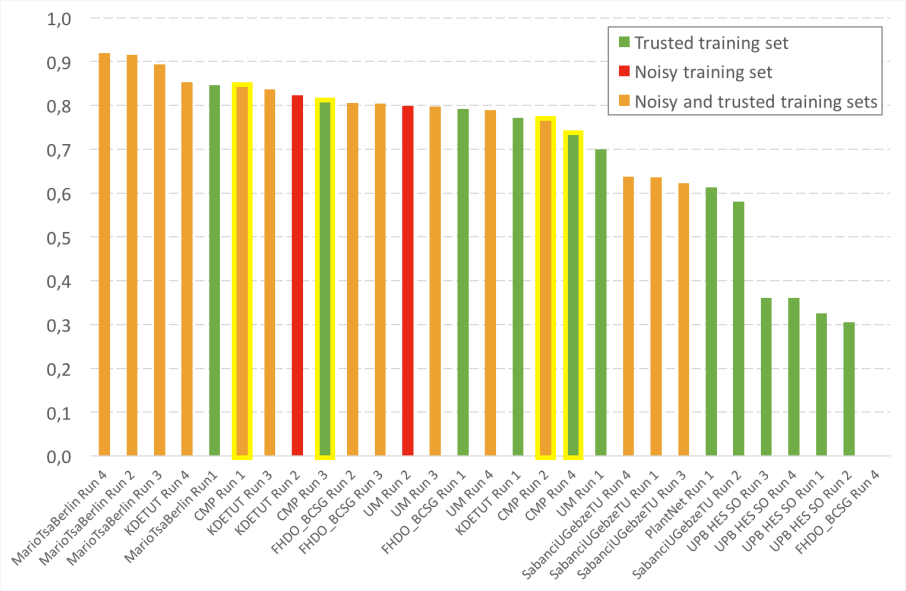

Fig. 5: Results of the PlantCLEF 2017 [1] challenge.

6 Conclusions

The difficulties of the challenge lie in the high number of classes, high intra-class variations, small inter-class variations, and learning from noisy data downloaded by web crawlers.

To overcome these difficulties, we employed a state-of-the-art deep learning architecture and compared a number of approaches to increase the accuracy of very fine-grained classification when learning from noisy data. The results of the challenge are depicted in Figure 5. Based on our evaluation, the following steps increase the classification accuracy:

– Maxout [14] with batch normalisation [15] of the added FC layer.
– Filtering the noisy data using a model trained on a trusted database.
– Bagging of several networks fine-tuned under different conditions.

Adjusting the species distribution on the test set, on the other hand, has decreased the recognition accuracy noticeably.

Acknowledgements

Milan Šulc was supported by Electrolux Student Support Programme and by CTU student grant SGS17/185/OHK3/3T/13, Jiří Matas was supported by The Czech Science Foundation Project GACR P103/12/G084.

References

1. Hervé Goëau, Pierre Bonnet, and Alexis Joly. Plant identification based on noisy web data: the amazing performance of deep learning (lifeclef 2017). In *CLEF working notes 2017*, 2017.
2. Alexis Joly, Hervé Goëau, Hervé Glotin, Concetto Spampinato, Pierre Bonnet, Willem-Pier Vellinga, Jean-Christophe Lombardo, Robert Planqué, Simone Palazzo, and Henning Müller. Lifeclef 2017 lab overview: multimedia species identification challenges. In *Proceedings of CLEF 2017*, 2017.
3. Alexis Joly, Hervé Goëau, Hervé Glotin, Concetto Spampinato, Pierre Bonnet, Willem-Pier Vellinga, Julien Champ, Robert Planqué, Simone Palazzo, and Henning Müller. Lifeclef 2016: multimedia life species identification challenges. In *Proceedings of CLEF 2016*, 2016.
4. Hervé Goëau, Pierre Bonnet, and Alexis Joly. Lifeclef plant identification task 2015. In *Working Notes of CLEF 2015 - Conference and Labs of the Evaluation forum, Toulouse, France, September 8-11, 2015*. CEUR-WS, 2015.
5. Hervé Goëau, Pierre Bonnet, and Alexis Joly. Plant identification in an open-world (lifeclef 2016). In *CLEF working notes 2016*, 2016.
6. Sungbin Choi. Plant identification with deep convolutional neural network: Snumedinfo at lifeclef plant identification task 2015. In *Working Notes of CLEF 2015 - Conference and Labs of the Evaluation forum, Toulouse, France, September 8-11, 2015*. CEUR-WS, 2015.
7. ZongYuan Ge, Chris McCool, Condrad Sanderson, and Peter Corke. Content specific feature learning for fine-grained plant classification. In *Working Notes of CLEF 2015 - Conference and Labs of the Evaluation forum, Toulouse, France, September 8-11, 2015*. CEUR-WS, 2015.
8. Julien Champ, Titouan Lorieul, Maximilien Servajean, and Alexis Joly. A comparative study of fine-grained classification methods in the context of the lifeclef plant identification challenge 2015. In *Working Notes of CLEF 2015 - Conference and Labs of the Evaluation forum, Toulouse, France, September 8-11, 2015*. CEUR-WS, 2015.
9. Angie K. Reyes, Juan C. Caicedo, and Jorge E. Camargo. Fine-tuning deep convolutional networks for plant recognition. In *Working Notes of CLEF 2015 - Conference and Labs of the Evaluation forum, Toulouse, France, September 8-11, 2015*. CEUR-WS, 2015.
10. Milan Šulc, Dmytro Mishkin, and Jirı Matas. Very deep residual networks with maxout for plant identification in the wild. In *Working notes of CLEF 2016 conference*, 2016.
11. Mostafa Mehdipour Ghazi, Berrin Yanikoglu, and Erchan Aptoula. Open-set plant identification using an ensemble of deep convolutional neural networks. *Working notes of CLEF*, 2016.
12. Siang Thye Hang, Atsushi Tatsuma, and Masaki Aono. Bluefield (kde tut) at lifeclef 2016 plant identification task. In *Working notes of CLEF 2016 conference*, 2016.

13. Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. *arXiv preprint arXiv:1602.07261*, 2016.
14. Ian J Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron Courville, and Yoshua Bengio. Maxout networks. *arXiv preprint arXiv:1302.4389*, 2013.
15. Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.
16. Scott Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. Training deep neural networks on noisy labels with bootstrapping. *arXiv preprint arXiv:1412.6596*, 2014.
17. Yves Grandvalet and Yoshua Bengio. Entropy regularization. *Semi-supervised learning*, pages 151–168, 2006.