

# Improving CNN classifiers by estimating test-time priors

Milan Sulc and Jiri Matas

Faculty of Electrical Engineering, Czech Technical University in Prague

sulcmila,matas@fel.cvut.cz

## Abstract

*The problem of different training and test set class priors is addressed in the context of CNN classifiers. We compare two approaches to the estimation of the unknown test priors: an existing Maximum Likelihood Estimation (MLE) method and a proposed Maximum a Posteriori (MAP) approach introducing a Dirichlet hyper-prior on the class prior probabilities. Experimental results show a significant improvement in the fine-grained classification tasks using known evaluation-time priors, increasing top-1 accuracy by 4.0% on the FGVC iNaturalist 2018 validation set and by 3.9% on the FGVCx Fungi 2018 validation set. Estimation of the unknown test set priors noticeably increases the accuracy on the PlantCLEF dataset, allowing a single CNN model to achieve state-of-the-art results and to outperform the competition-winning ensemble of 12 CNNs. The proposed MAP estimation increases the prediction accuracy by 2.8% on PlantCLEF 2017 and by 1.8% on FGVCx Fungi, where the MLE method decreases accuracy.*

## 1. Introduction

A common assumption of many machine learning algorithms is that the training set is independently sampled from the same data distribution as the test data [1, 8, 9]. In practice, this assumption is often violated - training samples may be obtained from diverse sources where classes appear with frequencies differing from the test-time. For instance, for the task of fine-grained recognition of plant species from images, training examples can be downloaded from an online encyclopedia. However, the number of photographs of a species in the encyclopedia may not correspond to the frequency a species is queried in a plant identification service.

Problems related to the differences between training- and test-set domains are studied in the field of domain adaptation [3, 14]. In transductive learning [6], the source and target tasks are the same, but the data representations or distributions are different. We are interested in the special case of calibrating classifier predictions when statistical properties of observations from the same class stay the same (i.e.



Figure 1. Examples from the fine-grained datasets FGVCx Fungi 2018 (top) and PlantCLEF 2017 (bottom).

appearance does not change), and the only assumed difference is in the class priors  $p(c_k)$ .

Methods [4, 16] for adjusting classifier outputs to new and unknown a-priori probabilities have been published years ago, yet the problem of changed class priors is commonly not addressed in computer vision tasks where the situation arises. An exception is the work of Royer et Lampert [15], who consider the case of sequential adaptation at prediction time (i.e. sample after sample) and take a classical Bayesian approach, using a symmetric Dirichlet distribution to form a posterior (mean) predictive estimate.

This paper focuses mainly on the case when a whole dataset is available at test time. Adopting the Maximum Likelihood Estimation (MLE) approach [4, 16], we propose an alternative solver for the MLE optimization, and we formulate a more stable Maximum a Posteriori (MAP) estimation approach with a Dirichlet hyperprior.

Section 2 provides a formulation of the problem: a probabilistic interpretation of CNN classifier outputs in Section 2.1, compensation for the change in a-priori class probabilities in Section 2.2 and estimation of the new a-priori probabilities using the frameworks of Maximum Likelihood in Section 2.3 and Maximum a Posteriori in Section 2.4.

Experiments in Section 3 show that state-of-the-art CNNs on fine-grained image classification tasks noticeably benefit from the adaptation to new class prior probabilities, and that the Dirichlet hyper-prior introduced to the proposed MAP approach improves the results over the ML estimate on most datasets. While our experiments focus on Neural Networks, the proposed framework is applicable to all classifiers with probabilistic (posterior) outputs.

## 2. Problem formulation and methodology

### 2.1. Probabilistic interpretation of CNN outputs

Let us assume that a classifier with parameters  $\theta^*$  is trained to provide an estimate of posterior probabilities of classes  $c_1, \dots, c_K \in C$  given an image observation  $\mathbf{x}_i$ :

$$f_{\text{CNN}}(c_k|\mathbf{x}_i, \theta^*) \approx p(c_k|\mathbf{x}_i), \quad (1)$$

This is a common interpretation of the process of training a deep network by minimizing the cross-entropy loss  $L_{\text{CE}}$  over samples  $\mathbf{x}_i$  with known class-membership labels  $c_{ik}$ :

$$\begin{aligned} \theta^* &= \arg \min_{\theta} L_{\text{CE}} = \arg \min_{\theta} - \sum_{i=1}^N \sum_{k=1}^K c_{ik} \log f(c_k|\mathbf{x}_i, \theta) \\ &= \arg \max_{\theta} \sum_{i=1}^N \log f(c_{y_i}|\mathbf{x}_i, \theta) = \arg \max_{\theta} \prod_{i=1}^N f(c_{y_i}|\mathbf{x}_i, \theta) \end{aligned} \quad (2)$$

where  $c_{ik}$  is a one-hot encoding of class label  $y_i$ :

$$c_{ik} = \begin{cases} 1 & \text{if } k = y_i \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

### 2.2. New a-priori class distribution

When the prior class probabilities  $p_e(c_k)$  in our validation/test<sup>1</sup> set differ from the training set, the posterior  $p_e(c_k|\mathbf{x}_i)$  changes too. The probability density function  $p(\mathbf{x}_i|c_k)$ , describing the statistical properties of observations  $\mathbf{x}_i$  of class  $c_k$ , remains unchanged:

$$p(\mathbf{x}_i|c_k) = \frac{p(c_k|\mathbf{x}_i)p(\mathbf{x}_i)}{p(c_k)} = p_e(\mathbf{x}_i|c_k) = \frac{p_e(c_k|\mathbf{x}_i)p_e(\mathbf{x}_i)}{p_e(c_k)} \quad (4)$$

Since  $\sum_{k=1}^K p_e(c_k|\mathbf{x}_i) = 1$ , we can get rid of the unknown probabilities  $p(\mathbf{x}_i)$ ,  $p_e(\mathbf{x}_i)$  of fixed sample  $\mathbf{x}_i$ :

$$p_e(c_k|\mathbf{x}_i) = p(c_k|\mathbf{x}_i) \frac{p_e(c_k)p(\mathbf{x}_i)}{p(c_k)p_e(\mathbf{x}_i)} \propto p(c_k|\mathbf{x}_i) \frac{p_e(c_k)}{p(c_k)} \quad (5)$$

The class priors  $p(c_k)$  can be empirically quantified as the number of images labeled as  $c_k$  in the training set. The test-time priors  $p_e(c_k)$  are, however, often unknown at test time.

<sup>1</sup>We use index  $e$  to denote all evaluation-time distributions.

### 2.3. ML estimate of new a-priori probabilities

Saerens et al. [16] proposed to approach the estimation of unknown test-time a-priori probabilities by iteratively maximizing the likelihood of the test observations

$$\begin{aligned} L(\mathbf{x}_1, \dots, \mathbf{x}_N) &= \prod_{i=1}^N p_e(\mathbf{x}_i) = \prod_{i=1}^N \left[ \sum_{k=1}^K p_e(\mathbf{x}_i, c_k) \right] = \\ &= \prod_{i=1}^N \left[ \sum_{k=1}^K p(\mathbf{x}_i|c_k)p_e(c_k) \right] \end{aligned} \quad (6)$$

They derive a simple EM algorithm comprising of the following steps:

$$p_e^{(s)}(c_k|\mathbf{x}_i) = \frac{p(c_k|\mathbf{x}_i) \frac{p_e^{(s)}(c_k)}{p(c_k)}}{\sum_{j=1}^K p(c_j|\mathbf{x}_i) \frac{p_e^{(s)}(c_j)}{p(c_j)}} \quad (7)$$

$$p_e^{(s+1)}(c_k) = \frac{1}{N} \sum_{i=1}^N p_e^{(s)}(c_k|\mathbf{x}_i) \quad (8)$$

where Eq. 7 is the Expectation-step, Eq. 8 is the Maximization-step, and  $p_e^0(c_k)$  may be initialized, for example, by the training set relative frequency  $\approx p(c_k)$ .

Du Plessis and Sugiyama [4] proved that this procedure is equivalent to fixed-point-iteration minimization of the KL divergence between the test observation density  $p_e(\mathbf{x})$  and a linear combination of the class-wise predictions  $q_e(\mathbf{x}) = \sum_{k=1}^K P_k p(\mathbf{x}|c_k)$ , where  $P_k$  are the estimates of  $p_e(c_k)$ .

$$\begin{aligned} \text{KL}(q_e||p_e) &= \int p_e(\mathbf{x}) \log \frac{p_e(\mathbf{x})}{q_e(\mathbf{x})} d\mathbf{x} = \\ &= \int p_e(\mathbf{x}) \log p_e(\mathbf{x}) d\mathbf{x} - \int p_e(\mathbf{x}) \log \sum_{k=1}^K P_k p(\mathbf{x}|c_k) d\mathbf{x} \end{aligned} \quad (9)$$

Note that estimating the priors  $\mathbf{P}^{\text{MLE}} = (P_1, \dots, P_K)$  by minimization of the KL divergence on the test set  $(\mathbf{x}_1, \dots, \mathbf{x}_N)$  can be rewritten as maximization of the log-likelihood  $\ell(\mathbf{x}_1, \dots, \mathbf{x}_N) = \log L(\mathbf{x}_1, \dots, \mathbf{x}_N)$  of the observed data given the prior probability estimates  $P_k \approx p_e(c_k)$ :

$$\begin{aligned} \arg \min_{\mathbf{P}} \text{KL}(q_e||p_e) &= \arg \max_{\mathbf{P}} \frac{1}{N} \sum_{i=1}^N \underbrace{\log \sum_{k=1}^K P_k p(\mathbf{x}_i|c_k)}_{\ell} \\ &= \arg \max_{\mathbf{P}} \sum_{i=1}^N \log \sum_{k=1}^K P_k \underbrace{\frac{p(c_k|\mathbf{x}_i)p(\mathbf{x}_i)}{p(c_k)}}_{\alpha_{ik}} = \mathbf{P}^{\text{MLE}} \\ \text{s.t. } \sum_{k=1}^K P_k &= 1; \forall k : P_k \geq 0 \end{aligned} \quad (10)$$

As shown in [4], using the EM algorithm from Eq. 7, 8 may not result in the unique optimal value, as the mapping of the fixed-point iteration is not a contraction mapping.

We therefore experiment also with direct optimization of the objective from Eq. 10 using the projected gradient descent algorithm [2]. At each step  $s$ , we update the variables as follows:

$$P_k^{(s+1)} = \pi \left( P_k^{(s)} + \lambda \frac{\partial \ell(\mathbf{x}_1, \dots, \mathbf{x}_N)}{\partial P_k} \right), \quad (11)$$

where  $\lambda$  is the learning rate,  $\pi$  represents projection onto the unit simplex, and the partial derivatives are:

$$\frac{\partial \ell(\mathbf{x}_1, \dots, \mathbf{x}_N)}{\partial P_k} = \sum_{i=1}^N \frac{a_{ik}}{\sum_{j=1}^K P_j a_{ij}} \quad (12)$$

To compute the Euclidean projection  $\pi$  onto the unit simplex, we use the efficient algorithm from [5, 18].

## 2.4. MAP estimate of new a-priori probabilities

Having a prior knowledge on the categorical distribution,  $p(\mathbf{P})$ , the maximum a-posteriori (MAP) estimate of the class prior probabilities is:

$$\begin{aligned} \mathbf{P}^{\text{MAP}} &= \arg \max_{\mathbf{P}} p(\mathbf{P} | (\mathbf{x}_1, \dots, \mathbf{x}_N)) \\ &= \arg \max_{\mathbf{P}} p(\mathbf{P}) \prod_{i=1}^N p(\mathbf{x}_i | \mathbf{P}) \\ &= \arg \max_{\mathbf{P}} \log p(\mathbf{P}) + \sum_{i=1}^N \log p(\mathbf{x}_i | \mathbf{P}) \\ &\text{s.t. } \sum_{k=1}^K P_k = 1; \forall k : P_k \geq 0 \end{aligned} \quad (13)$$

Note that the second term is the log-likelihood from the previous section,  $\ell(\mathbf{x}_1, \dots, \mathbf{x}_N) = \sum_{i=1}^N \log p(\mathbf{x}_i | \mathbf{P})$ .

Let us model the prior knowledge about the categorical distribution by the symmetric Dirichlet distribution:

$$p(\mathbf{P}) = \frac{1}{B(\alpha)} \prod_{k=1}^K P_k^{\alpha-1} \quad (14)$$

parameterized by  $\alpha > 0$ , where the normalization factor for the symmetric case is  $B(\alpha) = \frac{\Gamma(\alpha)^K}{\Gamma(\alpha K)}$ .

Choosing an  $\alpha \geq 1$  favours dense distributions, and thus avoids setting the categorical priors too close to zero. Zero priors may suppress even highly confident predictions. Moreover, the Dirichlet distribution with  $\alpha \geq 1$  is a log-concave distribution, allowing optimization with the

projected gradient descent optimizer from Section 2.3 by adding the following gradient components:

$$\frac{\partial \log p(\mathbf{P})}{\partial P_k} = \frac{\partial (\alpha - 1) \log(P_k)}{\partial P_k} = \frac{\alpha - 1}{P_k} \quad (15)$$

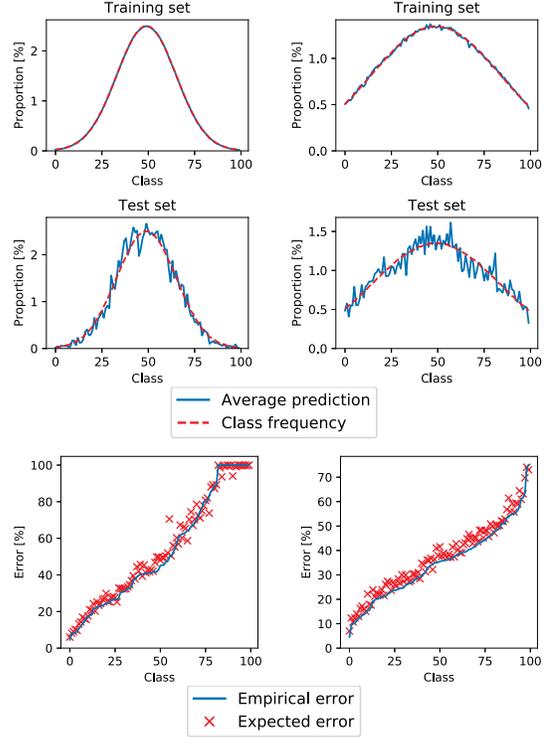


Figure 2. Top and middle row: Comparison of class frequency and CNN output marginalization over all images in the train- and test-sets sampled from CIFAR-100. Bottom row: test set empirical error  $\epsilon_k^{\text{emp}}$  and the expected error  $\epsilon_k$ , sorted by  $\epsilon_k^{\text{emp}}$ .

## 3. Experiments

The following fine-grained classification datasets are used for experiments in this Section:

**CIFAR-100** [11] is a popular dataset for smaller-scale classification experiments. It contains small resolution (32x32) color images of 100 classes. The full dataset contains 500 training samples and 100 test samples for each class. We sample a number of its unbalanced subsets for our experiments in this Section.

**PlantCLEF 2017** [7] was a plant species recognition challenge. The provided training images for 10,000 plant species consisted from an EOL<sup>2</sup> "trusted" training set, a significantly larger "noisy" training set (obtained from Google and Bing image search results, including mislabeled or irrelevant images), and the previous years (2015-2016) images depicting only a subset of the species. We use the

<sup>2</sup>downloaded from the Encyclopedia of Life, <http://www.eol.org/>

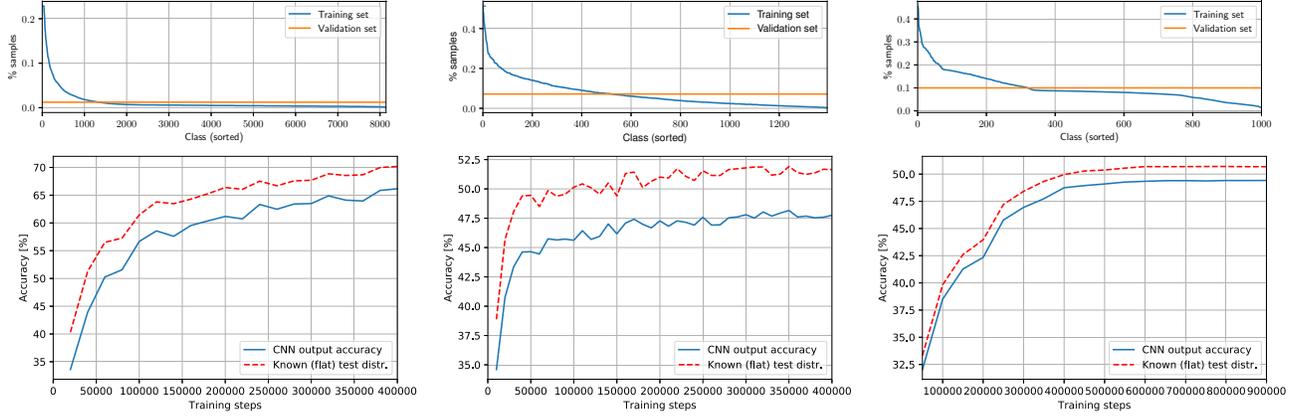


Figure 3. Training and validation set distributions (top) and accuracy before and after correcting predictions with the known/uniform val. set distribution (bottom) for FGVC iNaturalist 2018 (left), FGVCx Fungi 2018 (middle) and Webvision 2017 (right).

training data in two ways: Either training on all the sets together - further denoted as *PlantCLEF-All*, or excluding the "noisy" set - further denoted as *PlantCLEF-Trusted*. The test set from PlantCLEF 2017 is used for evaluation. All data is publicly available<sup>3</sup>. PlantCLEF presents an example of a real-world fine-grained classification task, where the number of available images per class is highly unbalanced.

**FGVC iNaturalist 2018** is a large scale species classification competition, organized with the FGVC5 workshop at CVPR 2018. The provided dataset covers 8,142 species of plants, animals and fungi. The training set is highly unbalanced and contains almost 440K images. A balanced validation set of 24K images is provided.

**FGVCx Fungi 2018** is another species classification competition, focused only on fungi, also organized with the FGVC5 workshop at CVPR 2018. The dataset covers nearly 1,400 fungi species. The training set contains almost 86K images, and is highly unbalanced. The validation set is balanced, with 4,182 images in total.

**Webvision 1.0** [13] (also Webvision 2017) is a large dataset designed to facilitate learning visual representation from noisy web data. It contains more than 2.4 million of images crawled from Flickr and Google Images and covers the same 1,000 classes as the ILSVRC 2012 dataset. The number of images per category ranges from hundreds to more than 10 thousand, depending on the number of queries generated from the synset for each category and on the availability of images on the Flickr and Google.

### 3.1. Validation of class posterior estimates

Let us check if the marginalization of predictions on training and validation data estimates the class priors well:

$$p(c_k) = \frac{1}{N} \sum_{i=1}^N p(c_k | \mathbf{x}_i) \approx \frac{1}{N} \sum_{i=1}^N f_{\text{CNN}}(c_k | \mathbf{x}_i) \approx \frac{N_k}{N}, \quad (16)$$

<sup>3</sup><http://imageclef.org/lifeclef/2017/plant>, [imageclef.org/lifeclef/2016/plant](http://imageclef.org/lifeclef/2016/plant)

where  $N_k = \sum_{i=1}^N c_{ik}$  is the number of images of class  $c_k$ . We randomly picked subsets of CIFAR-100 that follow chosen non-uniform distributions. A 32-layer Residual Network<sup>4</sup> [10] was trained on the training-subsets. The comparison of empirical class frequencies and the estimates obtained by marginalization (i.e. averaging CNN predictions) is displayed in Figure 2 (top). The training set class distributions are estimated almost perfectly. The estimates on the test set are more noisy, but approximate the class frequencies well.

Figure 2 (bottom) compares the expected error  $\epsilon_k$  and the empirical error  $\epsilon_k^{\text{emp}}$  for each class  $c_k$ :

$$\epsilon_k = \frac{1}{N_k} \sum_{i: y_i=k} 1 - p(c_k | \mathbf{x}_i), \quad (17)$$

$$\epsilon_k^{\text{emp}} = \frac{1}{N_k} \sum_{i: y_i=k} \llbracket k \neq \arg \max_{c_j} f_{\text{CNN}}(c_j | \mathbf{x}_i) \rrbracket, \quad (18)$$

### 3.2. Adjusting predictions when test-time priors are known

To experiment with known test-time prior probabilities  $p_e(c_k)$ , we use the training and validation sets from the FGVC iNaturalist<sup>5</sup> and the FGVCx Fungi<sup>6</sup> Classification Competitions 2018. In both challenges the validation sets are balanced, i.e. the class prior distribution is uniform. A state-of-the-art Convolutional Neural Network, Inception-v4 [17], was fine-tuned for each task. The predictions were corrected as defined by Eq.5.

A similar case is the Webvision 2017 dataset, where the training set is highly unbalanced and the validation set is balanced. In the classification/baseline experiments of Li et al. [13], the change of class prior probabilities is not taken into

<sup>4</sup><http://github.com/tensorflow/models/tree/master/research/resnet>

<sup>5</sup><http://sites.google.com/view/fgvc5/competitions/inaturalist>

<sup>6</sup><http://sites.google.com/view/fgvc5/competitions/fgvcx/fungi>

Train. distribution												
Acc.[%]	48.15	55.70	60.88	64.01	65.62	<b>67.29</b>	36.68	47.72	54.00	56.57	60.37	61.66
– after ML	49.71	56.94	61.64	64.58	65.62	67.11	38.67	49.05	55.18	57.05	60.59	61.74
– after MAP, $\alpha = 3$	49.75	56.94	61.65	<b>64.59</b>	65.64	67.18	38.75	49.20	55.19	<b>57.10</b>	60.58	<b>61.76</b>
– after MAP, $\alpha = 10$	<b>50.07</b>	<b>56.97</b>	<b>61.68</b>	64.55	<b>65.70</b>	67.23	<b>39.12</b>	<b>49.34</b>	<b>55.22</b>	<b>57.10</b>	<b>60.69</b>	<b>61.76</b>
Acc.[%] known $p_e(c_k)$	51.20	57.61	62.23	64.73	65.92	67.44	40.62	50.07	55.86	57.49	60.92	62.11

Table 1. Accuracy of CNN classifiers trained on unbalanced CIFAR-100 subsets (top) and evaluated on the full CIFAR-100 test set, adjusted by estimated class priors using the MLE and MAP estimates. Predictions adjusted by an oracle knowing the class priors (bottom).

consideration. Similarly to [13] we train an AlexNet network from scratch. (Note that our model did not converge to the same accuracy, probably due to difference in implementation and hyper-parameters.)

Figure 3 displays the training and evaluation distribution and the improvement in accuracy achieved by correcting the predictions with the known priors. The improvement in top-1 accuracy is **4.0%** and **3.9%** after 400K training steps (and up to **7.4%** and **4.9%** during fine-tuning) for the FGVC iNaturalist and FGVCx Fungi classification challenges respectively and **1.3%** for the Webvision 2017 dataset.

### 3.3. Adjusting predictions when the whole test set with unknown priors is available at test-time

The PlantCLEF 2017 test set is an example of a test environment where no knowledge about the class distribution was available. The training set is highly unbalanced, the test set does not follow the training set statistics and it does not contain examples from all classes.

We used an Inception-V4 model pre-trained on all available training data (*PlantCLEF-All*). Results in Table 2 show that the top-1 accuracy increases by **3.4%** when estimating the test set priors using the EM algorithm [16]. To compare with the results of the 2017 challenge, we combine the predictions per specimen observation (the test set contained several images per specimen, linked by ObservationID meta-data) and compute the observation-identification accuracy. After the test set prior-estimation our single CNN outperforms the winning submission of PlantCLEF 2017 composed of 12 CNNs (ResNet-152, ResNeXt-101 and GoogLeNet architectures).

Networks trained on the selected subsets of CIFAR-100 from Section 3.1 were evaluated on the full (balanced) CIFAR-100 test set with different adjustments of predictions: none, ML estimate, MAP estimate, and oracle-provided test-time priors. The results are compared in Table 1. As expected, the ground truth priors always lead to the best results. With only one exception, estimating the test-time priors always increases accuracy. The MAP estimate consistently achieves higher test-time accuracy, although, as illustrated in Figure 4, the likelihood of its estimate is lower than of the ML estimates. This demonstrates the importance of adding prior assumptions on the estimated class

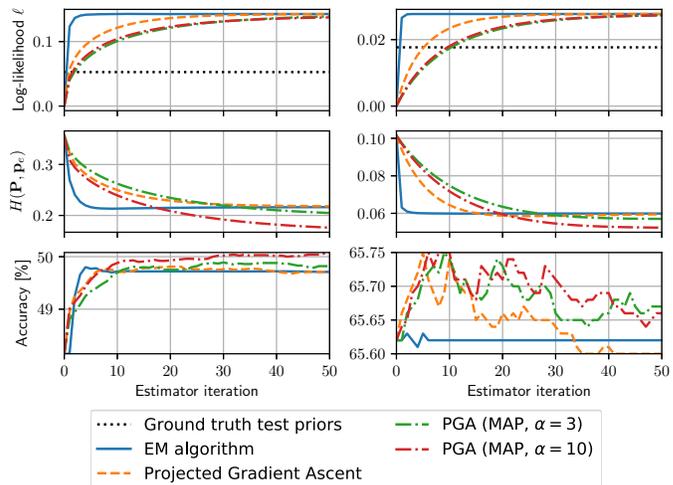


Figure 4. Iterative estimation of test-time priors on the full CIFAR-100 test set from CNNs trained on unbalanced CIFAR-100 subsets.

prior probabilities. The EM algorithm for ML estimation, however, converges noticeably faster.

Figure 5 summarizes the estimation of class priors on the fine-grained datasets PlantCLEF, FGVCx Fungi and Webvision. MAP estimation has a positive effect on the FGVCx Fungi dataset, where it increases accuracy by 1.8%, while ML estimate leads to a decrease in accuracy. All estimation methods decrease the accuracy on Webvision, MAP has the lowest decrease. The poor performance on Webvision may be related to the high number of outliers in the training set - Li et al. [13] suggest that only 66% of the images can be considered inliers. This may affect the reliability of the CNN posterior estimate. The accuracy on PlantCLEF increases by 2.8% after MAP estimation and by 3.4% after ML estimation. Note that on PlantCLEF, many classes are not present in the test set and therefore the optimization is actually disadvantaged by the Dirichlet hyperprior preventing the class priors from converging to zero.

### 3.4. Adjusting posterior probabilities on-line with new test samples

In practical tasks, test samples are often evaluated sequentially rather than all at once. We evaluated how the test-time class prior estimation on the PlantCLEF 2017 dataset

Model	Accuracy	Acc. after EM	Acc. per observation, (our method after EM)	Acc. per observation, $p_e(c_k)$ known
Inception V4	83.3%	86.7%	<b>90.8%</b>	93.7%
Ensemble of 12 CNNs [12] (PlantCLEF2017 winner)	–	–	88.5%	–

Table 2. Improvement in accuracy after applying the iterative test set prior estimation in the PlantCLEF 2017 plant identification challenge.

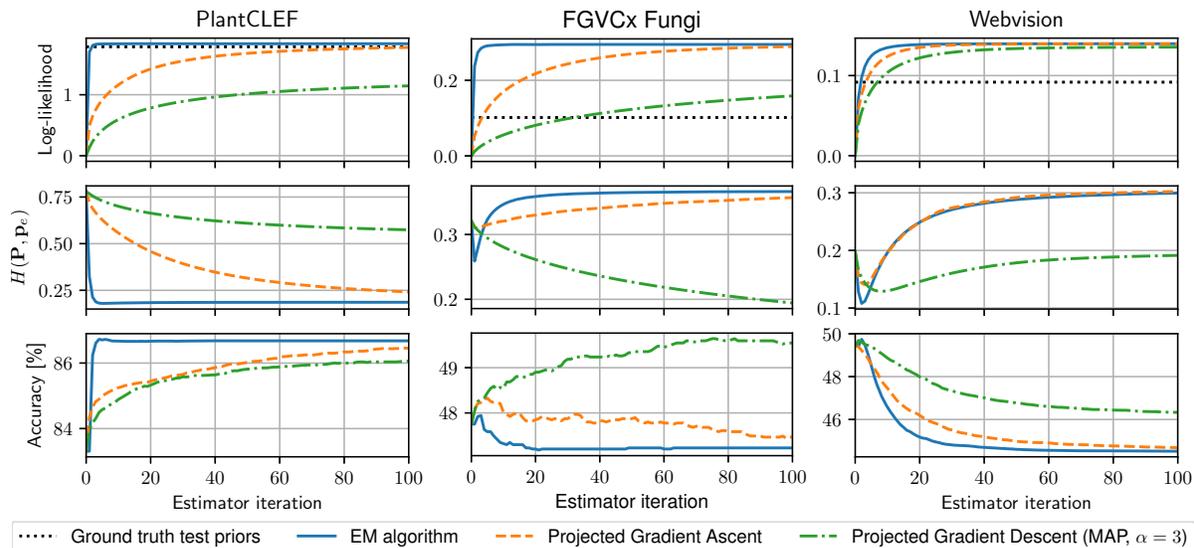


Figure 5. Iterative estimation of test-time priors on fine-grained datasets: PlantCLEF (Inception-v4), FGVCx Fungi (Inception-v4), and Webvision 1.0 (AlexNet). Top row: The log-likelihood surrogate  $\ell$ . Middle row: Hellinger distance between the prior estimate and ground truth class frequencies. Bottom row: Accuracy.

affects the results on-line, i.e. when the priors are estimated from the already seen examples, see Figure 6. After about 1,000 test samples, the predictions adjusted by class priors iteratively estimated by the EM algorithm gain a noticeable margin against plain CNN predictions.

## 4. Conclusions

The paper highlighted the importance of not ignoring the commonly found difference between the class priors in the training and test sets in computer vision. We compared two approaches: the existing MLE [?] and the proposed MAP approach, applying the Dirichlet prior on the categorical distributions.

Experimental results show a significant improvement on the FGVC iNaturalist 2018 and FGVCx Fungi 2018 classification tasks using the known evaluation-time priors, increasing the top-1 accuracy by 4.0% and 3.9% respectively. Iterative EM estimation of test-time priors on the PlantCLEF 2017 dataset increases the image classification accuracy by 3.4%, allowing a single CNN model to achieve state-of-the-art results and outperform the competition-winning ensemble of 12 CNNs. Adding the Dirichlet prior, preventing the class prior estimates from getting too close to zero, brings a slightly lower 2.8% increase in accuracy

on the PlantCLEF dataset (where many classes are actually missing in the test set), but improves the results and stability in most cases, including the FGVCx Fungi dataset, where it increased accuracy by 1.8% while the ML estimate would lead to a decrease. The estimation of new priors didn't help only on Webvision dataset - this may be related to the high amount ( $\approx 34\%$ ) of outliers in the dataset.

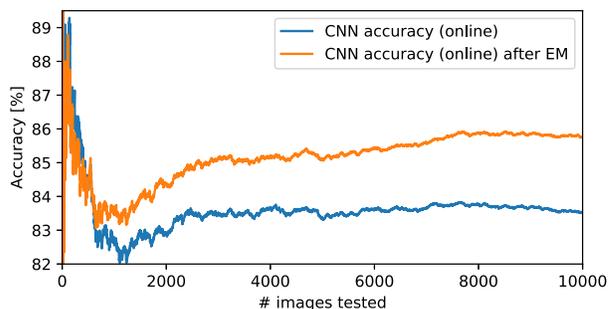


Figure 6. On-line test-prior estimation for PlantCLEF 2017.

## Acknowledgements

This research was supported by the Technology Agency of the Czech Republic project TE01020415 – V3C Visual Computing Competence Center, by Toyota Motor Europe and by the CTU student grant SGS17/185/OHK3/3T/13.

## References

- [1] C. M. Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, 2006.
- [2] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.
- [3] G. Csurka. Domain adaptation for visual applications: A comprehensive survey. *arXiv preprint arXiv:1702.05374*, 2017.
- [4] M. C. Du Plessis and M. Sugiyama. Semi-supervised learning of class balance under class-prior change by distribution matching. *Neural Networks*, 50:110–119, 2014.
- [5] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra. Efficient projections onto the  $l_1$ -ball for learning in high dimensions. In *Proceedings of the 25th international conference on Machine learning*, pages 272–279. ACM, 2008.
- [6] Y. Fu, T. M. Hospedales, T. Xiang, and S. Gong. Transductive multi-view zero-shot learning. *IEEE transactions on pattern analysis and machine intelligence*, 37(11):2332–2345, 2015.
- [7] H. Goeau, P. Bonnet, and A. Joly. Plant identification based on noisy web data: the amazing performance of deep learning (lifeclef 2017). *CEUR Workshop Proceedings*, 2017.
- [8] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.
- [9] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.
- [10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [11] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. 2009.
- [12] M. Lasseck. Image-based plant species identification with deep convolutional neural networks. *Working Notes of CLEF*, 2017, 2017.
- [13] W. Li, L. Wang, W. Li, E. Agustsson, and L. Van Gool. Webvision database: Visual learning and understanding from web data. *arXiv preprint arXiv:1708.02862*, 2017.
- [14] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.
- [15] A. Royer and C. H. Lampert. Classifier adaptation at prediction time. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1401–1409, 2015.
- [16] M. Saerens, P. Latinne, and C. Decaestecker. Adjusting the outputs of a classifier to new a priori probabilities: a simple procedure. *Neural computation*, 14(1):21–41, 2002.
- [17] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, volume 4, page 12, 2017.
- [18] W. Wang and M. A. Carreira-Perpinán. Projection onto the probability simplex: An efficient algorithm with a simple proof, and an application. *arXiv preprint arXiv:1309.1541*, 2013.