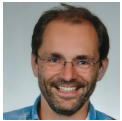
Fine-grained Recognition of Plants and Fungi from Images

Milan Šulc[1]



Jiří Matas[1]

Lukáš Picek[2]

[1]Department of Cybernetics
Faculty of Electrical Engineering
Czech Technical University in Prague

[2]Department of Cybernetics
Faculty of Applied Sciences
University of West Bohemia

May 17, 2021 @ Computer Vision Methods

Plant/Fungi Species Recognition

Many species identification tasks exist, depending on:

- the observation (view):
  1. from an image (or images) of a defined organ
  2. from image(s) of a larger part of the specimen
  3. from unspecified image(s) of the specimen
- acquisition conditions (controlled background, viewpoint, occlusion vs. unconstrained)
- number of *species* and granularity of the decision (typically fine-grained).

We formulate each of the considered tasks as a single-label classification problem on a set $\mathcal{C} = \{1, 2, \ldots, K\}$ of K classes (species). For fine-grained recognition $K \gg 1$.

Fine-grained Species Recognition Is a Difficult Task

The observation depends on many factors:

- Genotype
- Age
- Season
- Local environment (Climate, Altitude, Illumination)
- Clutter (other plants in the foreground or background)
- Acquisition conditions
- Device

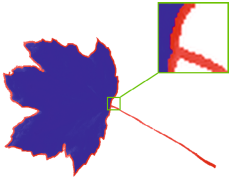
The classes (species) often have:

- Small inter-class differences
- High intra-class variability

Texture Recognition Approach to Plant Recognition

Recognition of bark (bark texture covers all image area):

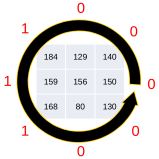
Recognition of segmented leaves:



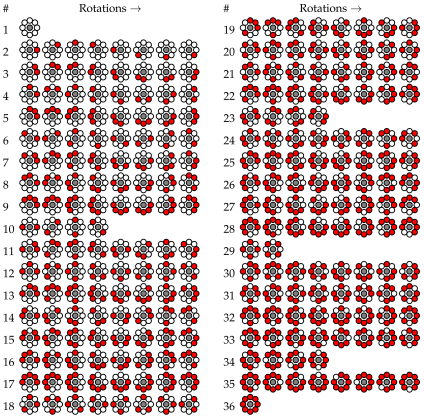
Fast Features Invariant to Rotation and Scale of Texture (Ffirst)

Ffirst extend previously published[1,2] texture descriptors.

- Completed Local Binary Patterns (sign- and magnitude-LBP)



- Rotation-invariant representation: "Histogram Fourier features"
- Scale space for multi-scale description and scale "invariance"
- Linear SVMs + explicit feature maps



[1] Milan Šulc and Jiří Matas. "Kernel-mapped histograms of multi-scale LBPs for tree bark recognition". In: *IVCNZ 2013*.

[2] Milan Šulc and Jiří Matas. "Tree Identification from Images". Master Thesis. Czech Technical University in Prague, 2014.

Texture Recognition Approach: Results

- Best results on the Austrian Federal Forests bark dataset
- Best results on all 6 experimented leaf datasets, achieving >99% accuracy on most datasets, including 99.5% accuracy on the MEW dataset of 153 tree species.[3]
- Competitive results[4] on standard texture recognition datasets:
  > 99% accuracy on the Brodatz32, UIUCTex, UMD, CUReT and KTH-TIPS datasets, 87.9% and 76.6% acc. on the more difficult KTH-TIPS2a and KTI-TIPS2b datasets.[5]
- Fast descriptor: 200x200 px image takes about 0.05 sec. on a laptop (without a GPU).
- We noticed a significant color bias on several standard texture recognition datasets and proposed improvements to global color descriptors.

[3] Later outperformed by deep learning approach Šulc and Matas (Plant Methods, 2017), achieving > 99.9% accuracy on the MEW dataset.

[4] Milan Šulc and Jiří Matas. "Fast Features Invariant to Rotation and Scale of Texture". In: *Computer Vision - ECCV 2014 Workshops* (Zurich, Switzerland). Vol. 8926. LNCS. Springer, 2015, pp. 47–62. ISBN: 978-3-319-16180-8. DOI: 10.1007/978-3-319-16181-5_4.

[5] The more recent CNN-based approach of Cimpoi et al. (IJCV, 2016) further improves the classification scores on most texture recognition datasets.

Deep Learning Approach to Species Identification "in the Wild"

Deep Convolutional Neural Network (CNN) for the tasks of plant and fungi recognition "in the wild":

- Large-scale databases available from computer vision challenges (LifeCLEF 2016-2019, FGVCx Flowers 2018, FGVCx Fungi 2018).
- **Prior shift** is a common phenomenon: Distribution of species on training data often differs from test data.

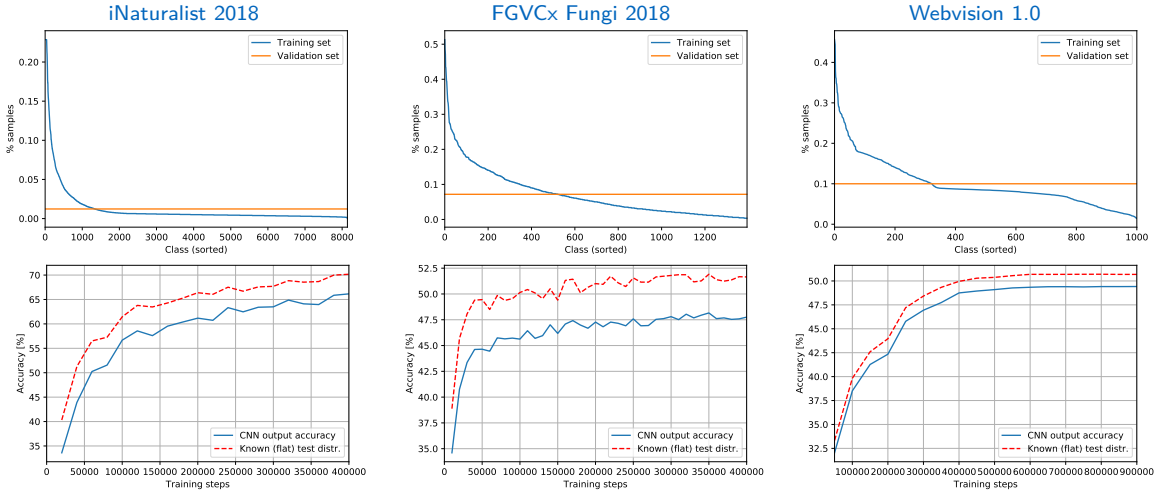
Prior Shift

A CNN classifier $\boldsymbol{f}_{\mathsf{CNN}} : \mathbb{R}^{W \times H \times 3} \mapsto \Delta^{K-1}$ is typically trained by minimizing the cross entropy loss $L_{\mathsf{CE}} = -\sum_{i=1}^{N} \log f_{\mathsf{CNN}}(y_i|\boldsymbol{x}_i;\boldsymbol{\theta}))$ on a training set $\mathcal{T} = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_N, y_N)\}$ to estimate posterior probabilities: $f_{\mathsf{CNN}}(k|\boldsymbol{x};\boldsymbol{\theta}) \approx p(k|\boldsymbol{x})$.

Assume $p(\boldsymbol{x}|k)$ remains unchanged, but class priors $p_Y^e(k)$ on the new data differ from the training set $p_Y(k)$. Then $p^e(k|\boldsymbol{x})$ differs from $p(k|\boldsymbol{x}) \approx f_{\mathsf{CNN}}(k|\boldsymbol{x})$:

$$p^e(k|\boldsymbol{x}) = p(k|\boldsymbol{x}) \frac{p_Y^e(k) p_X(\boldsymbol{x})}{p_Y(k) p_X^e(\boldsymbol{x})} \propto p(k|\boldsymbol{x}) \frac{p_Y^e(k)}{p_Y(k)}$$

When New Priors Are Known



Maximum Likelihood Estimation of New Priors

Saerens et al.[6] proposed the following EM procedure to maximize the likelihood $L(\mathcal{E}) = \prod_{\boldsymbol{x}\in\mathcal{E}} p_X^e(\boldsymbol{x}) = \prod_{\boldsymbol{x}\in\mathcal{E}} \sum_{k=1}^{K} p^e(\boldsymbol{x}, k) = \prod_{\boldsymbol{x}\in\mathcal{E}} \sum_{k=1}^{K} p(\boldsymbol{x}|k) p_Y^e(k)$:

$$p^{(s)}(k|\boldsymbol{x}) = \frac{p(k|\boldsymbol{x}) \dfrac{\hat{p}_k^{(s)}}{p_Y(k)}}{\sum_{j=1}^{K} p(j|\boldsymbol{x}) \dfrac{\hat{p}_j^{(s)}}{p_Y(j)}} \tag{1}$$

$$\hat{p}_k^{(s+1)} = \frac{1}{N^e} \sum_{\boldsymbol{x}\in\mathcal{E}} p^{(s)}(k|\boldsymbol{x}) \tag{2}$$

[6] Marco Saerens et al. "Adjusting the outputs of a classifier to new a priori probabilities: a simple procedure". In: *Neural computation* 14.1 (2002), pp. 21–41.

Maximum Likelihood Estimation of New Priors

The EM algorithm may not result[7] in the unique optimal value.

We therefore also experiment with optimization of the MLE objective

$$\hat{\boldsymbol{p}}^{\mathsf{MLE}} = \arg\max_{\boldsymbol{p}} \sum_{\boldsymbol{x}\in\mathcal{E}} \log \sum_{k=1}^{K} p_k \underbrace{\frac{p(k|\boldsymbol{x})}{p_Y(k)}}_{a_{ik}} \cancel{p_X(\boldsymbol{x})} \text{ s.t. } \sum_{k=1}^{K} p_k = 1;\ \forall k : p_k \geq 0$$

by projected gradient ascent:

$$\hat{p}_k^{(s+1)} = \pi\left(\hat{p}_k^{(s)} + \lambda \frac{\partial \ell(\mathcal{E})}{\partial \hat{p}_k}\right), \text{ where } \frac{\partial \ell(\mathcal{E})}{\partial \hat{p}_k} = \sum_{\boldsymbol{x}\in\mathcal{E}} \frac{a_{ik}}{\sum_{j=1}^{K} \hat{p}_j a_{ij}}$$

[7] Marthinus Christoffel Du Plessis and Masashi Sugiyama. "Semi-supervised learning of class balance under class-prior change by distribution matching". In: *Neural Networks* 50 (2014), pp. 110–119.

Maximum A Posteriori Estimation of New Priors

Assuming a hyper-prior $p(\boldsymbol{p})$ on the categorical priors:

$$\hat{\boldsymbol{p}}^{\text{MAP}} = \arg\max_{\boldsymbol{p}} p(\boldsymbol{p}|\mathcal{E}) = \arg\max_{\boldsymbol{p}} p(\boldsymbol{p}) \prod_{\boldsymbol{x}\in\mathcal{E}} p(\boldsymbol{x}|\boldsymbol{p}) =$$

$$= \arg\max_{\boldsymbol{p}} \left[\log p(\boldsymbol{p}) + \sum_{\boldsymbol{x}\in\mathcal{E}} \log p(\boldsymbol{x}|\boldsymbol{p})\right] \text{ s.t. } \sum_{k=1}^{K} p_k = 1;\ \forall k : p_k \geq 0$$

We use symmetric Dirichlet distribution: $p(\boldsymbol{p}) = \frac{1}{B(\alpha)} \prod_{k=1}^{K} p_k^{\alpha-1}$

favouring dense distributions, i.e. parameterized by $\alpha > 1$.

Log-concave, just adds gradient components: $\frac{\partial \log p(\hat{\boldsymbol{p}})}{\partial \hat{p}_k} = \frac{\alpha - 1}{\hat{p}_k}$

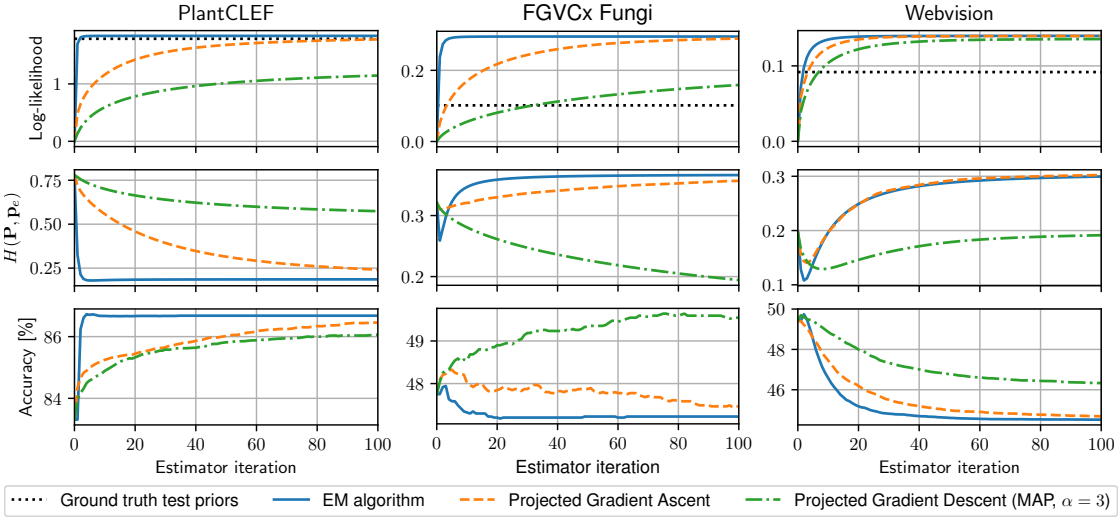
Adjusting Predictions by Estimated Priors

CNN trained on unbalanced CIFAR-100 subsets and evaluated[8] on full test set:

Train. distribution												
Acc. (%)	48.15	55.70	60.88	64.01	65.62	**67.29**	36.68	47.72	54.00	56.57	60.37	61.66
after MLE	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	**64.59**	65.64	67.18	38.75	49.20	55.19	**57.10**	60.58	**61.76**
after MAP, $\alpha = 10$	**50.07**	**56.97**	**61.68**	64.55	**65.70**	67.23	**39.12**	**49.34**	**55.22**	**57.10**	**60.69**	**61.76**
with known $p^e_Y(k)$	51.20	57.61	62.23	64.73	65.92	67.44	40.62	50.07	55.86	57.49	60.92	62.11

[8] Milan Šulc and Jiří Matas. "Improving CNN classifiers by estimating test-time priors". In: *Proceedings of the IEEE International Conference on Computer Vision Workshops*. 2019.

Adjusting Predictions by Estimated Priors



LifeCLEF Plant (Herbs, Trees, Ferns) Identification Challenges

PlantCLEF 2016 →→

- 1 000 plant species, 113K training images

PlantCLEF 2017

- 10,000 plant species
- "trusted" train. set: 256K images (EoL.org)
- "noisy" train. set: 1.4M images (web search)

ExpertLifeCLEF 2018

- same 10,000 species as in 2017
- comparison against human experts

PlantCLEF 2019

- 10,000 species from "data-defficient" regions
- "trusted" train. set: 72K images (EoL.org)
- "noisy" train. set: 375K images (web search)

Meta-data available:

- "types of view": *Leaf*, *Leaf Scan*, *Flower*, *Fruit*, *Stem*, *Branch*, *Entire*
- "Observation ID" available

LifeCLEF Plant (Herbs, Trees, Ferns) Identification Challenges

PlantCLEF 2016

- 1 000 plant species, 113K training images

PlantCLEF 2017 →→

- 10,000 plant species
- "trusted" train. set: 256K images (EoL.org)
- "noisy" train. set: 1.4M images (web search)

ExpertLifeCLEF 2018 →→

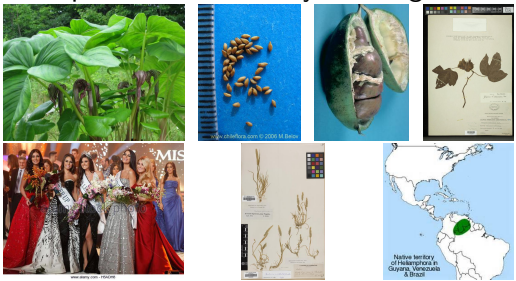
- same 10,000 species as in 2017
- comparison against human experts

PlantCLEF 2019

- 10,000 species from "data-defficient" regions
- "trusted" train. set: 72K images (EoL.org)
- "noisy" train. set: 375K images (web search)

Examples from the "trusted" training set:

Examples from the "noisy"training set:



LifeCLEF Plant (Herbs, Trees, Ferns) Identification Challenges

PlantCLEF 2016

- 1 000 plant species, 113K training images

PlantCLEF 2017

- 10,000 plant species
- "trusted" train. set: 256K images (EoL.org)
- "noisy" train. set: 1.4M images (web search)

ExpertLifeCLEF 2018

- same 10,000 species as in 2017
- comparison against human experts

PlantCLEF 2019 →→

- 10,000 species from "data-defficient" regions
- "trusted" train. set: 72K images (EoL.org)
- "noisy" train. set: 375K images (web search)
- Species from the Guiana shield and the Amazon rain forest
- 20% of species: less than 10 images

Examples from the "trusted" training set:

Examples of wrongly annotated images from the "noisy" training set:

FGVC 2018 Competitions

FGVCx Fungi:

- 1394 fungi species
- 85 578 train. + 4 182 val. images
- 9 758 competition test images

FGVCx Flower classification challenge:

- 997 species of flowering plants
- 669 304 train. images, no val. set
- 12 961 competition test images

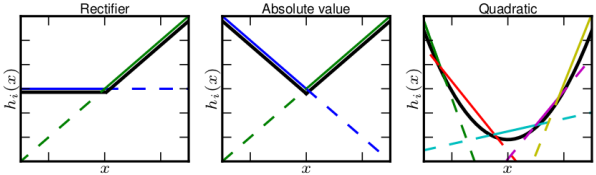
iNaturalist 2018:

- 8 142 species (plants, fungi, animals, ...)
- 437 513 train. + 24 426 val. images
- 149 394 competition test images

Species Recognition Competitions: Methods

- Fine-tuning state-of-the-art CNN classifiers[9,10] from ImageNet checkpoints
- Combining predictions from an ensemble of CNNs (sum or mode)
- Combining predictions from images with the same ObservationID (if available)
- Filtering noisy data
- Image augmentation (during training and at test time)
- Running averages (exponential decay) of trainable variables:

  $68.4\% \rightarrow 80.2\%$ acc. (!) with Inception-ResNet-v2
- Adjusting predictions to prior shift
- Maxout[11]: $\forall i \in \{1, \ldots, m\}: \quad h_i(x) = \max_{j \in \{1,\ldots,s\}} z_{ij}, z_{ij} = \boldsymbol{x}^\mathsf{T} \boldsymbol{W}_{.ij} + b_{ij}$



[9] Kaiming He et al. "Deep residual learning for image recognition". In: *CVPR*. 2016.

[10] Christian Szegedy et al. "Inception-v4, inception-resnet and the impact of residual connections on learning.". In: *AAAI*. 2017.

Species Recognition Competitions: What Didn't Work

- Handling "types of view" separately decreased precision in preliminary experiments
- "Bootstrapping"[12] consistency objectives for training on noisy labels

$$L_{\text{soft}}(\boldsymbol{q}, \boldsymbol{t}) = \sum_{k=1}^{K} [\beta t_k + (1-\beta) q_k] \log q_k, \qquad L_{\text{hard}}(\boldsymbol{q}, \boldsymbol{t}) = \sum_{k=1}^{K} [\beta t_k + (1-\beta) z_k] \log q_k$$

  - *soft bootstrapping* uses the softmax predictions q_k,
  - *hard bootstrapping* uses only the strongest prediction $z_k = \begin{cases} 1 \text{ if } k = \arg\max_i q_i \\ 0 \text{ otherwise} \end{cases}$

[12] Scott Reed et al. "Training deep neural networks on noisy labels with bootstrapping". In: *arXiv preprint arXiv:1412.6596* (2014).

LifeCLEF Competitions: Results

	Place	Our score	Winner score	Metric
PlantCLEF 2016	3rd	70.1% mAP	74.2% mAP	Mean average precision
+ ObservationID $\sum$		78.8% mAP		
PlantCLEF 2017	3rd	84.3% MRR	92.0% MRR	Mean reciprocal rank
+ correct prior shift		86.7% MRR		$\text{MRR} = \frac{1}{\lvert Q \rvert} \sum_{i=1}^{\lvert Q \rvert} \frac{1}{\text{rank}_i}$
PlantCLEF 2018	**1st**	88.4% acc.	$=$	Top-1 accuracy
PlantCLEF 2019	(post-challenge)	31.9% acc.	24.7%	Top-1 accuracy

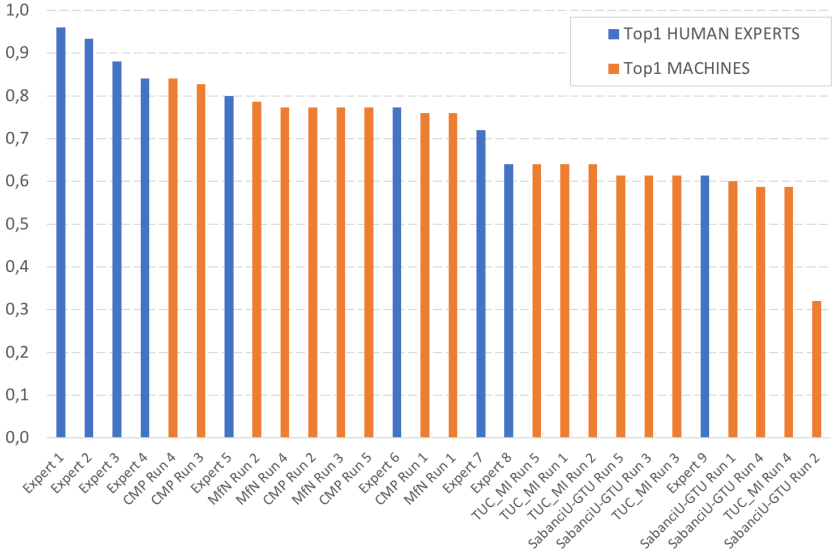
FGVC 2018 Competitions: Official Results (top 10)

FGVCx Fungi	Recall@3 Err.(%)	
Team Name	Private	Public
CMP (ours)	**21.197**	**20.772**
digitalspecialists	23.188	23.471
Val An	25.091	25.213
DL Analytics	28.341	26.853
Invincibles	28.751	28.493
Tian Xi	32.235	31.636
Igor Krashenyi	32.616	34.164
wakaka	42.219	41.339
George Yu	47.621	47.113
Xinshao	67.837	67.509

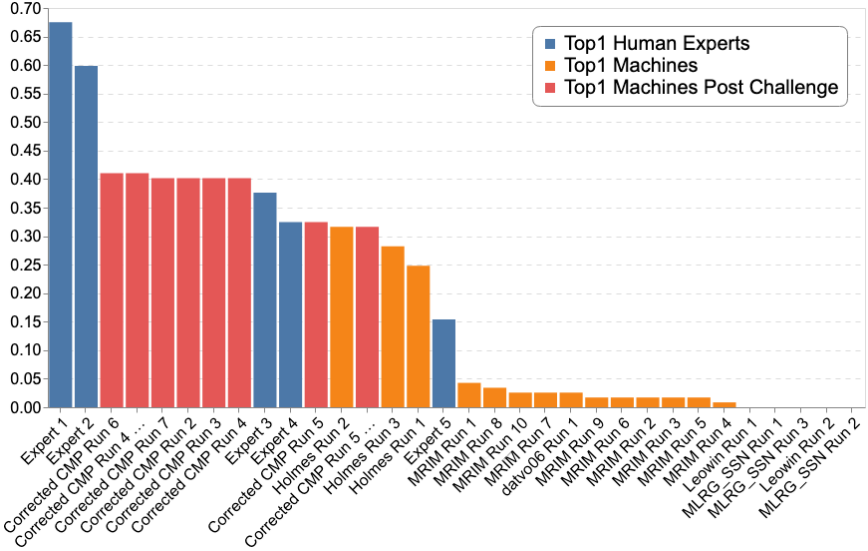
iNaturalist	Recall@3 Err.(%)	
Team Name	Private	Public
DLUT VLG	12.858	13.068
DL Analytics	13.981	14.214
fadivugibs	14.618	14.914
CMP (ours)	**16.076**	**16.360**
flSHpAM	16.892	17.149
traveler	16.988	17.235
yen	17.201	17.412
Shuang	18.357	18.549
Mr.M	20.092	20.291
Dequan Wang	20.814	21.157

FGVCx Flowers	Top 1 Err.(%)	
Team Name	Private	Public
CMP (ours)	**7.599**	**6.828**
fadivugibs	8.177	7.638
DLUT VLG	8.242	7.677
yen	8.396	7.716
xiaoxiao	9.579	8.641
NDer MJU	11.636	10.976
thesouthfrog	15.211	13.618
nimahai	16.368	15.258
Miroslav Štola	20.187	19.637
lmao	20.342	20.177

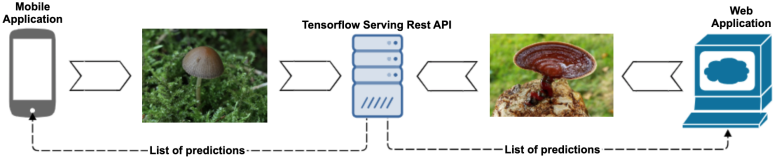
ExpertLifeCLEF 2018: Experts vs. Machines

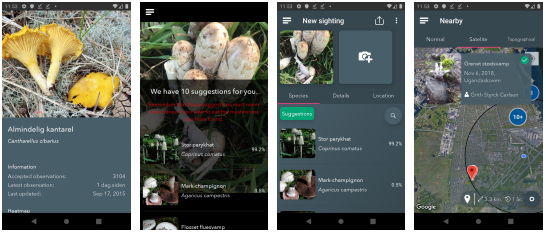


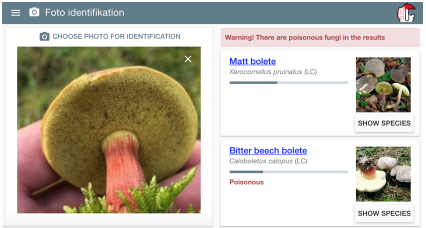
PlantCLEF 2019: Experts vs. Machines on Data Deficient Species



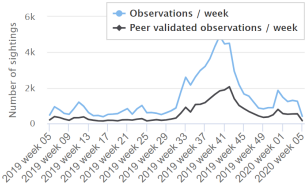
Fungi Recognition Service for the Atlas of Danish Fungi







Oct.-Dec. 2018: 11K new sightings with images
Oct.-Dec. 2019: 21K new sightings with images
Oct.-Dec. 2020: 28K new sightings with images



Danish Fungi 2020 Dataset

- fungi observations from 1874 till the end of 2020
- precise annotation with taxonomy-accurate class labels
- highly unbalanced long-tailed class distribution
- rich observation meta-data (time, location, substrate, habitat, .. for $> 99\%$ images
- zero overlap with ImageNet

DF20

- 1,604 classes
- 248,466 training images
- 27,608 (public) test images

DF20 – Mini

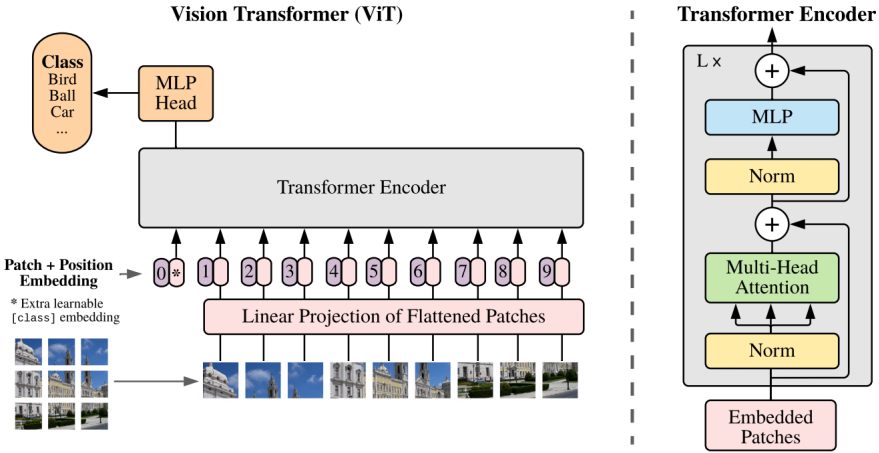
- 182 classes
- 32,753 training images
- 3,640 (public) test images

Danish Fungi 2020: Baseline Experiments

			Top1 [%]	**Top3** [%]	$\mathbf{F}_1^m$		**Top1** [%]	**Top3** [%]	$\mathbf{F}_1^m$
DF20–Mini	EfficientNet-B0	224×224	65.66	83.35	0.531	384×384	70.22	85.69	0.596
	EfficientNet-B3		66.90	83.49	0.537		72.09	87.17	0.624
	SE-ResNeXt-101		69.48	85.58	0.593		72.34	87.53	0.631
	ViT-Base/16		69.37	**86.54**	0.589		74.84	88.74	0.655
	ViT-Large/16		**70.71**	86.51	**0.599**		**75.96**	**89.37**	**0.664**
DF20	EfficientNet-B0		70.38	85.18	0.613		75.27	88.65	0.670
	ViT-Base/16		73.45	87.15	0.658		79.40	90.93	0.724
	ViT-Large/16		**75.32**	**88.12**	**0.679**		**81.25**	**91.93**	**0.747**

Table 1: Classification results of selected CNN and ViT architectures on DF20 and DF20 - Mini dataset for two input resolutions.

Best on Danish Fungi 2020: Vision Transformers[13]



[13] Alexey Dosovitskiy et al. "An image is worth 16x16 words: Transformers for image recognition at scale". In: *arXiv preprint arXiv:2010.11929* (2020).

Danish Fungi 2020: Baseline for Utilizing Metadata

Assumption: $P(I|S) = P(I|S, D)$.

Then the class posterior given the image I and metadata D is:

$$P(S|I, D) = P(S|I)\frac{P(S|D)}{P(S)}\frac{P(I)}{P(I|D)} \propto P(S|I)\frac{p(S|D)}{p(S)},$$

To use several metadata, we assume:

$$P(S|D_1, D_2) \propto \frac{P(S|D_1)P(S|D_2)}{P(S)}.$$

Danish Fungi 2020: Utilizing Metadata

	H	M	S	Top1 [%]	Top3 [%]	F_1^m
Danish Fungi 2020	×	×	×	73.45	87.15	0.658
	✔	×	×	+2.00	+1.42	+0.036
	×	✔	×	+1.37	+1.23	+0.024
	×	×	✔	+0.98	+0.96	+0.016
	×	✔	✔	+2.30	+2.10	+0.039
	✔	×	✔	+2.92	+2.41	+0.051
	✔	✔	×	+3.16	+2.50	+0.056
	✔	✔	✔	**+3.58**	**+3.05**	**+0.062**

Table 2: Performance gains from *Fungus* observation metadata: H - Habitat, S - Substrate, M - Month, and their combinations, on DF20. ViT-Base/16 with image size 224×224.

Recent Work

The Danish Fungi 2020 dataset and benchmark

- published on arXiv[14]
- submitted to a major computer vision conference (in review)
- coming soon: public benchmark with **DNA sequenced private test set**

Classifier calibration and prior shift adaptation

- recently submitted a patent application with Toyota
- submitted to a major computer vision conference (in review)

[14] Lukáš Picek et al. "Danish Fungi 2020 – Not Just Another Image Recognition Dataset". In: *arXiv preprint arXiv:2103.10107* (2021).

Estimating new priors using Confusion Matrix

A standard procedure [13,15] for prior estimation is based on the classifier confusion matrix in the format $C_{d|y}$, containing probabilities $p(D = i|Y = k)$ of deciding for label i when the true class is k.

$$p(D = i) = \sum_{k=1}^{K} p(D = i|Y = k)p(Y = k)$$

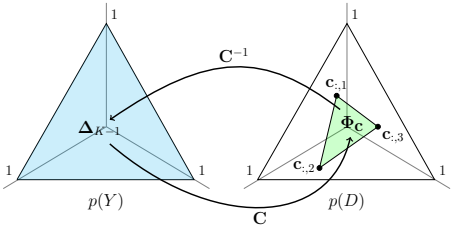
$$p(D) = C_{d|y}p(Y)$$

The new priors can then be estimated as follows:

$$\hat{p}_{\mathcal{E}}(Y) = \hat{C}_{d|y}^{-1}\hat{p}_{\mathcal{E}}(D)$$

Problem of the simple Confusion Matrix method

The confusion matrix defines a convex set Φ_C of feasible values $p(D)$ within the probability simplex:



Inconsistent estimates of the Confusion Matrix and $p(D)$ can result in a vector outside of the probability simplex, i.e. the estimate can contain negative values. E.g.:

$$\hat{C}_{d|y} = \begin{bmatrix} 0.8 & 0.2 \\ 0.2 & 0.8 \end{bmatrix}, \ \hat{p}\mathcal{E}(D) = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \ \rightarrow \ \hat{p}(Y) = \begin{bmatrix} 4/3 \\ -1/3 \end{bmatrix}$$

The proposed MLE based on Confusion Matrix

We assume a multinomial distribution of the observed classifier decision:

$$L(\boldsymbol{Q}) = p(\boldsymbol{n}|\boldsymbol{Q}) = \frac{(n_1 + \ldots + n_K)!}{n_1! \cdot \ldots \cdot n_K!} \cdot q_1^{n_1} \cdot \ldots \cdot q_K^{n_K}$$

The log-likehood $\ell(\boldsymbol{P}) = \log p(n|\boldsymbol{P}) = \sum_{k=1}^{K} n_k \log(\boldsymbol{c}_{k,:} \cdot \boldsymbol{P}) + \theta_{\boldsymbol{n}}$ is then maximized with condition on solutions on the probabilistic simplex:

$$\hat{P} = \arg\max_{\boldsymbol{P}} \ell(\boldsymbol{P}) = \arg\max_{\boldsymbol{P}} \sum_{k=1}^{K} n_k \log \boldsymbol{c}_{k,:}\boldsymbol{P}$$

$$\text{s.t.:} \quad \sum_{k=1}^{K} P_k = 1; \quad \forall k : P_k \geq 0$$

Adding a hyper-prior is convenient

$$\hat{P}_{MAP} = \arg\max_{\boldsymbol{P}} p(\boldsymbol{P}|\boldsymbol{n}) = \arg\max_{\boldsymbol{P}} p(\boldsymbol{P})p(\boldsymbol{n}|\boldsymbol{P})$$
$$= \arg\max_{\boldsymbol{P}} \log p(\boldsymbol{P}) + \arg\max_{\boldsymbol{P}} \log p(\boldsymbol{n}|\boldsymbol{P})$$

s.t. $\quad \forall k : P_k \geq 0; \quad \sum_{k=1}^{K} P_k = 1$

Results

Dataset		BCTS calibrated	NA	CM	CM^L	SCM	SCM^L	Oracle
CIFAR100*	LT→UNI	✗	$31.66^{\pm 1.27}$	$22.02^{\pm 2.56}$	$\mathbf{33.00^{\pm 1.56}}$	$26.64^{\pm 3.85}$	$\mathbf{33.50^{\pm 1.38}}$	$34.06^{\pm 1.35}$
	LT→UNI	✓	$31.71^{\pm 1.29}$	$18.56^{\pm 3.70}$	$\mathbf{31.90^{\pm 1.49}}$	$24.52^{\pm 5.49}$	$\mathbf{33.78^{\pm 1.30}}$	$34.54^{\pm 1.32}$
	UNI→LT	✗	$63.83^{\pm 0.82}$	$\mathbf{68.06^{\pm 0.92}}$	$68.04^{\pm 0.76}$	$68.10^{\pm 0.81}$	$\mathbf{68.23^{\pm 0.71}}$	$69.13^{\pm 0.58}$
	UNI→LT	✓	$63.83^{\pm 0.82}$	$\mathbf{69.08^{\pm 0.94}}$	$69.07^{\pm 1.00}$	$69.31^{\pm 0.94}$	$\mathbf{69.38^{\pm 0.78}}$	$70.65^{\pm 0.75}$
Places365	LT→UNI	✗	$25.14^{\pm 0.14}$	$17.79^{\pm 0.28}$	$\mathbf{27.77^{\pm 0.45}}$	$19.78^{\pm 2.21}$	$\mathbf{28.64^{\pm 0.22}}$	$32.99^{\pm 0.46}$
	LT→UNI	✓	$25.16^{\pm 0.14}$	$15.52^{\pm 0.66}$	$\mathbf{27.11^{\pm 0.32}}$	$17.86^{\pm 1.62}$	$\mathbf{27.32^{\pm 0.18}}$	$33.51^{\pm 0.25}$
	UNI→LT	✗	$58.17^{\pm 1.01}$	$81.16^{\pm 0.61}$	$\mathbf{81.67^{\pm 0.66}}$	$82.04^{\pm 0.15}$	$\mathbf{82.09^{\pm 0.64}}$	$88.14^{\pm 0.27}$
	UNI→LT	✓	$58.17^{\pm 1.01}$	$81.16^{\pm 0.59}$	$\mathbf{81.70^{\pm 0.61}}$	$82.04^{\pm 0.15}$	$\mathbf{82.42^{\pm 0.90}}$	$88.15^{\pm 0.30}$
ImageNet	LT→UNI	✗	$34.30^{\pm 0.19}$	$19.48^{\pm 0.74}$	$\mathbf{33.57^{\pm 0.33}}$	$23.94^{\pm 2.04}$	$\mathbf{35.78^{\pm 0.17}}$	$37.34^{\pm 0.15}$
	LT→UNI	✓	$34.31^{\pm 0.19}$	$16.78^{\pm 0.64}$	$\mathbf{31.94^{\pm 0.51}}$	$20.26^{\pm 4.08}$	$\mathbf{35.55^{\pm 0.23}}$	$37.45^{\pm 0.19}$

Table 3: *"Improve Estimates from Confusion Matrix."* Accuracy (± std. dev.) after adaptation with new prior estimate based on confusion matrix (CM) inversion and our proposed method (CM^L). SCM denotes soft confusion matrix, NA denotes no adaptation, Oracle is adaptation with ground truth priors. Results on CIFAR are averaged from 10 experiments, results on Places and ImageNet are averaged from 5 experiments.

Results

Dataset		BCTS calibrated	NA	MLE EM	MLE CM^L	MLE SCM^L	MAP MAP	MAP CM^M	MAP SCM^M	Oracle
CIFAR100*	LT→UNI	✗	$31.66^{\pm 1.27}$	$32.81^{\pm 1.41}$	$33.00^{\pm 1.56}$	$\mathbf{33.50^{\pm 1.38}}$	$32.87^{\pm 1.40}$	$33.88^{\pm 1.40}$	$\mathbf{33.92^{\pm 1.36}}$	$34.06^{\pm 1.35}$
	LT→UNI	✓	$31.71^{\pm 1.29}$	$26.18^{\pm 1.61}$	$31.90^{\pm 1.49}$	$\mathbf{33.78^{\pm 1.30}}$	$27.85^{\pm 1.54}$	$33.96^{\pm 1.46}$	$\mathbf{34.11^{\pm 1.62}}$	$34.54^{\pm 1.32}$
	UNI→LT	✗	$63.83^{\pm 0.82}$	$67.23^{\pm 0.88}$	$68.04^{\pm 0.76}$	$\mathbf{68.23^{\pm 0.71}}$	$66.89^{\pm 0.92}$	$67.11^{\pm 0.91}$	$\mathbf{67.13^{\pm 0.79}}$	$69.13^{\pm 0.58}$
	UNI→LT	✓	$63.83^{\pm 0.82}$	$69.17^{\pm 0.91}$	$69.07^{\pm 1.00}$	$\mathbf{69.38^{\pm 0.78}}$	$68.30^{\pm 0.77}$	$\mathbf{68.42^{\pm 0.85}}$	$68.26^{\pm 0.72}$	$70.65^{\pm 0.75}$
Places365	LT→UNI	✗	$25.14^{\pm 0.14}$	$28.02^{\pm 0.92}$	$27.77^{\pm 0.45}$	$\mathbf{28.64^{\pm 0.22}}$	$28.86^{\pm 0.68}$	$30.27^{\pm 0.41}$	$\mathbf{30.88^{\pm 0.36}}$	$32.99^{\pm 0.46}$
	LT→UNI	✓	$25.16^{\pm 0.14}$	$26.25^{\pm 1.02}$	$27.11^{\pm 0.32}$	$\mathbf{27.32^{\pm 0.18}}$	$28.29^{\pm 0.08}$	$29.37^{\pm 0.30}$	$\mathbf{29.39^{\pm 0.14}}$	$33.51^{\pm 0.25}$
	UNI→LT	✗	$58.17^{\pm 1.01}$	$\mathbf{82.63^{\pm 0.31}}$	$81.67^{\pm 0.66}$	$82.09^{\pm 0.64}$	$\mathbf{78.81^{\pm 0.47}}$	$76.97^{\pm 0.45}$	$75.54^{\pm 0.43}$	$88.14^{\pm 0.27}$
	UNI→LT	✓	$58.17^{\pm 1.01}$	$\mathbf{82.70^{\pm 0.34}}$	$81.70^{\pm 0.61}$	$82.42^{\pm 0.90}$	$\mathbf{77.05^{\pm 0.44}}$	$76.12^{\pm 0.59}$	$73.29^{\pm 0.46}$	$88.15^{\pm 0.30}$
ImageNet	LT→UNI	✗	$34.30^{\pm 0.19}$	$34.63^{\pm 0.29}$	$33.57^{\pm 0.33}$	$\mathbf{35.78^{\pm 0.17}}$	$35.10^{\pm 0.21}$	$36.51^{\pm 0.17}$	$\mathbf{36.90^{\pm 0.16}}$	$37.34^{\pm 0.15}$
	LT→UNI	✓	$34.31^{\pm 0.19}$	$24.39^{\pm 2.31}$	$31.94^{\pm 0.51}$	$\mathbf{35.55^{\pm 0.23}}$	$32.08^{\pm 2.11}$	$35.39^{\pm 0.09}$	$\mathbf{36.80^{\pm 0.16}}$	$37.45^{\pm 0.19}$

Table 4: *"How to estimate new priors?"* Accuracy (± std. dev.) after adaptation to new priors estimated with different Maximum Likelihood and Maximum A Posteriori estimates. NA denotes no adaptation, Oracle is adaptation with ground truth priors. Results on CIFAR are averaged from 10 experiments, results on Places and ImageNet are averaged from 5 experiments.

Summary

- *Fast Features Invariant to Rotation and Scale of Texture*
  - accuracy $> 99\%$ on most leaf datasets
  - competitive results in texture classification: $> 99\%$ acc. on standard texture datasets
  - computationally efficient: 200x200 px image takes about 0.05s
- Deep learning approach for "in-the-wild" species classification:
  - competition-winning contributions to fine-grained recognition challenges
  - reached human-expert level of accuracy
  - applied as an online recognition service for a Atlas of Danish Fungi, increasing the involvement of users in biodiversity data collection
- Adjusting to prior shift is important.
  - estimating new priors using MLE compared with a proposed MAP estimation, increasing the reliability and accuracy in several tasks
  - an analogous approach can be used to utilize metadata in the decision process
- The Danish Fungi 2020 dataset for fine-grained classification:
  - accurate class labels, zero overlap with ImageNet, and rich meta-data

Thank you.

Discussion.