Czech Technical University in Prague
Faculty of Electrical Engineering

DOCTORAL THESIS

May 2020

Milan Šulc

Czech Technical University in Prague
Faculty of Electrical Engineering
Department of Cybernetics

Fine-grained Recognition of Plants and Fungi from Images

Doctoral Thesis

Milan Šulc

Prague, May 2020

Ph.D. Programme: Electrical Engineering and Information Technology
Branch of Study: Artificial Intelligence and Biocybernetics

Supervisor: prof. Ing. Jiří Matas, Ph.D.

Abstract

The thesis contributes to fine-grained recognition of plant and fungi species from images, ranging from scans and photos of leaves and bark taken in controlled conditions to unconstrained observations of plants and fungi “in the wild” with complex background and clutter in the scene. The constrained tasks of bark and leaf identification are approached as a texture recognition problem. For more complex species recognition tasks with large scale datasets available, we take a deep learning approach. In many instances of the species recognition problem, test-time categorical priors differ from the training set. We address the problems of adjusting outputs of probabilistic classifiers to new priors and estimating the new priors. In particular, we note that training a neural network by cross entropy minimization leads to a model whose outputs should be an estimate of the posterior probabilities. We experimentally validate related statistical properties of the outputs of Convolutional Neural Network (CNN) classifiers. For estimation of test-time categorical priors, a Maximum Likelihood estimation approach is compared with a proposed Maximum a Posteriori estimation, adding a hyper-prior favouring dense prior distributions. We show that adding such hyper-prior increases the reliability of the estimate and increases the classification accuracy in several fine-grained classification tasks.

The proposed texture recognition method, Fast Features Invariant to Rotation and Scale of Texture (Ffirst), achieved excellent results in leaf and bark classification, as well as in standard texture classification. The deep learning approach presented in the thesis has scored first in several species recognition competitions on “in the wild” plant and fungi identification, where the views of the observed specimen vary significantly and the difficulty is increased by occlusions and background clutter. The results confirm the benefits of practices such as combining predictions from an ensemble of models, filtering potentially noisy data, data augmentation, and using the moving averages of the trained variables. An experimental comparison with human experts in plant identification shows that the best ensembles of deep CNNs reach the human expert accuracy in image-based plant identification. The competition-winning model for fungi recognition is applied in a citizen-science project and assists the collection of fungi observations, valuable for several research fields including mycology and biodiversity research.

Abstrakt

Tato práce se zabývá rozpoznáváním druhů rostlin a hub z obrazu, od rozpoznávání skenů a fotografií listů a kůry v kontrolovaných podmínkách až po neomezená pozorování rostlin a hub “ve volné přírodě” s komplikovaným pozadím a změtí různých objektů ve scéně. Rozpoznávání kůry a listů jsme řešili pomocí rozpoznávání textury. Ke složitějším úlohám rozpoznávání druhů rostlin a hub s velkým množstvím trénovacích dat jsme použili hluboké učení neuronových sítí. V úlohách rozpoznávání druhů se apriorní pravděpodobnosti tříd na trénovací a testovací sadě často liší. Věnujeme se problémům přizpůsobení výstupů pravděpodobnostních klasifikátorů novým apriorním pravděpodobnostem a odhadu těchto pravděpodobností. Poukazujeme, že učení neuronové sítě minimalizací křížové entropie vytváří model, který by měl odhadovat aposteriorní pravděpodobnosti. Experimentálně ověřujeme některé statistické vlastnosti takových modelů. Pro odhad nových apriorních pravděpodobností porovnáváme metodu maximální věrohodnosti (MLE) a navržený přístup metodou Maximum a Posteriori (mAP), v níž přidáváme hyper-prior upřednostňující pravděpodobnostní rozdělení bližší rovnoměrnému rozdělení. Ukazujeme, že takový hyper-prior zvyšuje spolehlivost odhadu a přesnost klasifikace na několika klasifikačních úlohách. Navržená metoda pro rozpoznávání textury, Fast Features Invariant to Rotation and Scale of Texture (Ffirst), dosáhla vynikajících výsledků v klasifikaci listů a kůry, jakož i ve standardním rozpoznávání textur. Hluboké konvoluční sítě prezentované v této práci se umístily na prvním místě v několika mezinárodních soutěžích v automatickém rozpoznávání druhů rostlin a hub “ve volné přírodě”, s různorodými pohledy na sledované jedince, často s překryvem a komplikovaným pozadím. Výsledky potvrzují výhodnost postupů jako jsou kombinace predikcí souboru několika modelů, filtrování potenciálně chybně anotovaných dat, rozšiřování (augmentace) dat, nebo používání plovoucího průměru trénovaných proměnných. Experimentální porovnání s lidskými experty v rozpoznávání rostlin ukazuje, že nejlepší soubory hlubokých neuronových sítí dosahují v rozpoznávání rostlin z obrazu přesnosti lidských expertů. Model, který vyhrál soutěž v automatickém rozpoznávání hub, byl aplikován do projektu občanské vědy (citizen-science), kde asistuje při sběru dat o pozorování hub, důležitých pro řadu oborů jako např. mykologie či výzkum biodiverzity.

Acknowledgement

I am very grateful to Jiří Matas, my advisor, for his patient guidance through my doctoral studies, for sharing his knowledge and skills, and for teaching me the art of research. I cannot thank him enough for his support, for all discussions full of novel ideas, and for his enthusiasm to always push the state of the art.

I would like to thank all my colleagues and friends from the Visual Recognition Group and the Center for Machine Perception for creating a supportive and pleasant research environment. I am grateful to the co-authors of my publications listed in Appendix A. Special thanks to Lukáš Picek for his contributions to our winning submissions to several plant and fungi recognition competitions and for staying up for our night calls before the deadlines.

I would like to thank Mirko Navara, Jana Nosková, Tomáš Hodaň and Milan Pšenička for their valuable feedback on the thesis manuscript.

I am also grateful for having the chance to work on other interesting computer vision problems at the time of my PhD studies during my internship at Xerox Research Centre Europe, during my Google internship in the Mobile Vision team, and during my work on the Czech Technical University projects with Electrolux and Toyota.

Finally, I want to thank my family for their endless support, which allowed me to pursue my dreams.

I gratefully acknowledge that the research was supported by the CTU student grants SGS13/142/OHK3/2T/13, SGS15/155/OHK3/2T/13 and SGS17/185/OHK3/3T/13, by the Electrolux Student Support Programme and by Toyota Motor Europe. Access to computing and storage facilities owned by parties and projects contributing to the National Grid Infrastructure MetaCentrum, provided under the programme "Projects of Large Research, Development, and Innovations Infrastructures" (CESNET LM2015042), is greatly appreciated.

Authorship

I hereby certify that the results presented in this thesis were achieved during my own research, in cooperation with my thesis advisor Jiřıi Matas, with Lukáš Picek published in [144,183,184], with Dmytro Mishkin published in [182], with Thomas Jeppesen and Jacob Heilmann-Clausen published in [184], and with Pierre Bonnet, Hervé Goeau, Siang Thye Hang, Mario Lasseck, Valéry Malécot, Philippe Jauzein, Jean-Claude Melet, Christian You and Alexis Joly published in [17].

Contents

1 Introduction 1
1.1 Problem Formulation 4
1.2 Contributions 7
1.3 Structure of the Thesis 8

2 Texture Recognition Approach 9
2.1 Related Work 9
  2.1.1 Leaf Recognition 9
  2.1.2 Tree Bark Recognition 11
  2.1.3 Texture Recognition 11
2.2 FFIRST 13
  2.2.1 Completed Local Binary Pattern and Histogram Fourier Features . . 13
  2.2.2 Multi-scale Description and Scale Invariance 14
  2.2.3 Support Vector Machine and Feature Maps 15
  2.2.4 Adding Rotational Invariants 16
  2.2.5 Recognition of Segmented Textural Objects 17
2.3 Datasets and Evaluation Methodology 18
  2.3.1 Tree Bark Dataset 18
  2.3.2 Leaf Datasets 18
  2.3.3 Texture Recognition Datasets 23
2.4 Results 28
  2.4.1 Texture Classification 28
  2.4.2 Tree Bark Classification 28
  2.4.3 Leaf Classification 29
2.5 Significance of Colors in Texture Datasets 30
  2.5.1 Color Statistics for Classification 31
  2.5.2 Selected Color Descriptors 32
  2.5.3 Experiments with Color Descriptors on Texture Datasets 32
  2.5.4 Discussion of the Results 34

3 Deep Learning for Species Recognition in the Wild 37
3.1 CNN Classifier Architectures 38
3.1.1 AlexNet 38
3.1.2 VGG 39
3.1.3 GoogLeNet / Inception v1 39
3.1.4 Inception v2 and Inception v3 39
3.1.5 ResNets 41
3.1.6 Inception v4 and Inception-ResNets 42
3.1.7 Efficient Architectures, Neural Architecture Search 42
3.1.8 Maxout Networks 42
3.2 Probabilistic Interpretation of CNN Outputs 43
3.2.1 Checking the Properties of Class Posterior Estimates 44
3.2.2 Over-confident Classifiers 46
3.3 Difference in Categorical Priors 48
3.3.1 New A Priori Class Distribution 48
3.3.2 ML Estimate of New A Priori Probabilities 49
3.3.3 MAP Estimate of New A Priori Probabilities 51

4 In-the-wild Classification Competitions 53
4.1 PlantCLEF 2016 53
4.1.1 The PlantCLEF 2016 Plant Identification Challenge 53
4.1.2 The Proposed Approach: Very Deep Residual Maxout Networks . . 54
4.1.3 Preliminary Results and Validation 55
4.1.4 Results on the Test Set 57
4.1.5 Discussion 57
4.2 PlantCLEF 2017 58
4.2.1 The PlantCLEF 2017 Plant Identification Challenge 58
4.2.2 Methods 59
4.2.3 Experiments 61
4.2.4 Description of the Submitted Runfiles 64
4.2.5 Post Challenge Evaluation with Correct Prediction Adjustment . . . 64
4.2.6 Discussion 65
4.3 Experts vs. Machines in Plant Identification 2017 66
4.3.1 Training Data and Method 66
4.3.2 Test Data and Evaluation Protocols 66
4.3.3 Results 68
4.4 ExpertLifeCLEF 2018 72
4.4.1 Methodology 73
4.4.2 Submissions 75
4.4.3 Results 75
4.4.4 Discussion 76
4.5 FGVC Challenges 2018 78
4.5.1 FGVCx Fungi Dataset 79
4.5.2 iNaturalist 2018 Dataset 79
4.5.3 FGVCx Flowers Dataset 80

4.5.4 Method . . . 80
4.5.5 Results . . . 83
4.5.6 Discussion . . . 88
4.6 PlantCLEF 2019 . . . 89
4.6.1 Dataset . . . 89
4.6.2 Methodology . . . 90
4.6.3 Results . . . 94
4.6.4 Discussion . . . 95
4.7 Summary of the Challenge Results . . . 96

5 Fungi Recognition for Citizen-Science 97
5.1 Related Work . . . 97
5.1.1 Fungi Recognition . . . 97
5.1.2 Crowd-based Image Collection and Identification . . . 98
5.2 Online Fungi Classification Service . . . 99
5.3 Results . . . 100
5.4 Discussion . . . 102

6 Evaluation of Adjusting to New Priors 105
6.1 Adjusting Predictions When Test-time Priors Are Known . . . 106
6.2 Estimation of New Priors From the Test Set . . . 106
6.2.1 Cross-validation of the Prior Estimate Likelihood . . . 109
6.3 Changing the Training Set Priors . . . 110
6.4 Adjusting Posterior Probabilities Online . . . 111
6.5 Temperature Scaling . . . 111
6.6 Discussion . . . 113

7 Conclusions 115

A Author's Publications 135
A.1 Publications Related to the Topic of the Thesis . . . 135
A.1.1 Impacted Journal Paper . . . 135
A.1.2 Conference and Workshop Papers . . . 135
A.1.3 Book Chapter . . . 136
A.2 Publications Not Related to the Topic of the Thesis . . . 137
A.2.1 US Patent . . . 137

B Author's Curriculum Vitae 139

Abbreviations

AFF	Austrian Federal Forest (dataset). 11, 18, 19
AI	Artificial Intelligence. 101
CN	Color Names (descriptor). 31–34
CNN	Convolutional Neural Network. iii, 7, 8, 12, 28, 30, 31, 37, 38, 42–48, 51, 54, 56, 59, 66, 72, 74, 75, 77, 81–88, 91, 95, 96, 99, 106–111, 113, 115, 116
COCO	Common Objects in Context (dataset, challenge). 37, 54
DD	Discriminative Color Descriptors. 31–33
DFT	Discrete Fourier Transform. 13, 16
EoL	Encyclopedia Of Life (web encyclopedia). 3, 58, 61, 62, 66, 72, 74, 89, 91, 105
FC	Fully Connected (layer). 60, 65
Ffirst	Fast Features Invariant to Rotation and Scale of Texture. iii, v, 8, 9, 13–17, 23, 28–31, 34, 115
GBIF	Global Biodiversity Information Facility. 90
GLCM	Grey-Level Co-Occurrence Matrix. 11
GPU	Graphics Processing Unit. 12, 30, 38, 55, 61, 115
i.i.d.	Independent and Identically Distributed (Sampling). 43, 44
IFV	Improved Fisher Vectors. 12
ILSVRC	ImageNet Large Scale Visual Recognition Challenge. 37–39, 41, 42, 54, 106
LBP	Local Binary Patterns. 11–14, 16, 17
MAP	Maximum A Posteriori (estimation). 48, 51, 93, 94, 108, 109, 112, 113
mAP	Mean Average Precision. v, 53, 55–57
MEW	Middle European Woods (dataset). 10, 22
MLE	Maximum Likelihood Estimation. v, 43, 44, 48, 93, 94, 108, 109, 112, 113
MRR	Mean Reciprocal Rank. 65, 66, 70
NLL	Negative Log Likelihood. 47
ResNet	Residual Neural Network. 38, 41, 54
REST	REpresentational State Transfer (API). 99

RGB	(Red, Green, Blue) color model. 12, 32
SIFT	Scale Invariant Feature Transform. 10, 11, 31
SVM	Support Vector Machine. 10, 11, 15, 17, 33, 55, 56
VGG	Visual Geometry Group at the University of Oxford; acronym also commonly used for the CNNs from [170]. 12, 38, 91
VOC	PASCAL Visual Object Classes. 26, 37

CHAPTER 1

Introduction

The aim of this thesis is to study, develop, and apply computer vision and machine learning algorithms for automating identification of plants and fungi from images – photos or scans.

Recognition of natural objects in their natural environment has been of great importance for the humankind since time immemorial. The skills of plant, fungi and animal species identification were crucial for survival throughout the human history: Until approximately 12 000 years ago, virtually all humanity lived as hunters and gatherers [113] for whom foraging was the only source of food. Recognizing edible species from poisonous species or identification of dangerous predators was a matter of life and death. The transition from forager to producer societies, enabled by plant and animal domestication, known as the Neolithic revolution, caused a major demographic shift [16]. Agriculture and its continuous improvement allowed demographic growth from around 6 million individuals [14] at the beginning of the transition to agriculture to around 7.7 billion today [36]. Without the basic human skill of plant and animal identification, the great shift towards producer societies would never be possible.

Precise identification of plant and fungi species is still important for many areas of human activity other than agriculture. For example, herbs have been used for medical purposes since the prehistoric times. Chemicals derived from herbs and other plants, phytochemicals, are commonly used by modern pharmaceutical industry. Textile and cosmetics industries have traditionally heavily relied on specific plant species. More recently, species incidence and biodiversity observations have been used to study different environmental factors [24], including climate change [47]. Such studies heavily rely on the collection of data on appearance and occurrence of species and annotations of such data, often with the help of citizen scientist communities [46, 86]. Recent publications stress that tens of thousands of plant species are currently threatened with extinction [38, 191].

The scientific approach to describing living nature lead biologists to grouping individuals into species, organizing them hierarchically into larger groups, and giving those groups names. This theory and practice form the field of *biological taxonomy* [92], and the categorization into taxa, i.e. groups of organisms, is commonly denoted as *biological classification*. The modern systematics for grouping organisms and naming them with bi-

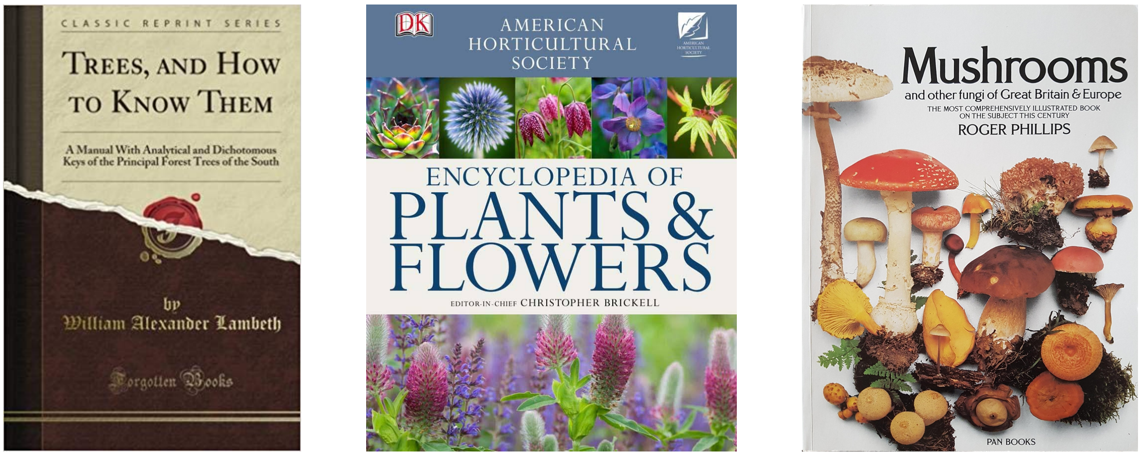
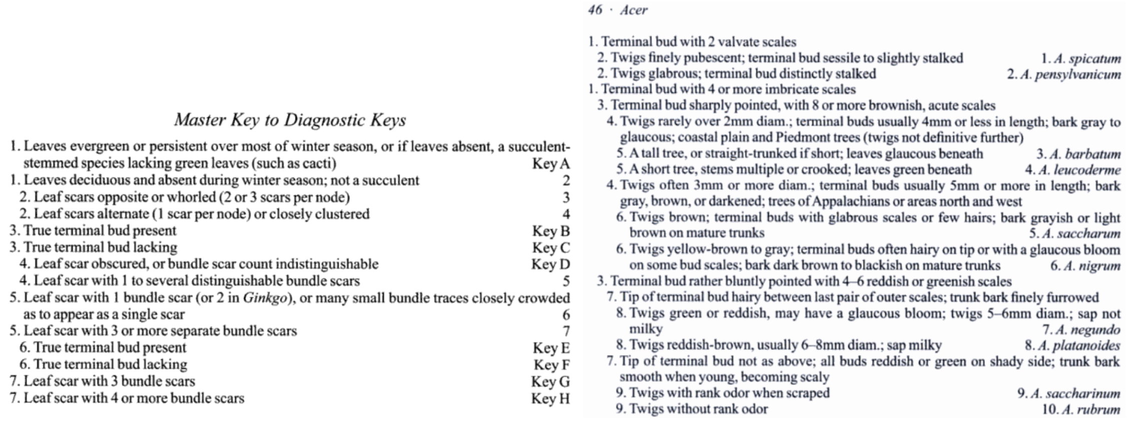

CLASSIC REPRINT SERIES
TREES, AND HOW TO KNOW THEM
A Manual With Analytical and Dichotomous Keys of the Principal Forest Trees of the South
by
William Alexander Lambeth
Forgotten Books
DK
AMERICAN HORTICULTURAL SOCIETY
ENCYCLOPEDIA OF PLANTS & FLOWERS
EDITOR-IN-CHIEF CHRISTOPHER BRICKELL
Mushrooms
and other fungi of Great Britain & Europe
THE MOST COMPREHENSIVELY ILLUSTRATED BOOK ON THE SUBJECT THIS CENTURY
ROGER PHILLIPS
PAN BOOKS

Figure 1.1: Examples of traditional species identification handbooks and encyclopedias [21, 105, 143]

nomial nomenclature[1] is based on the works of Carl von Linné - Species Plantarum [120] and Systema Naturae [121]. Contemporary biological taxonomies include enormous numbers of categories and species. For illustration, while von Linné's Species Plantarum [120] describes 5 940 plant species, currently the number of published and accepted plant species is over 310 000 [27]. The seven main taxonomic ranks are: *kingdom*, *phylum*, *class*, *order*, *family*, *genus* and *species*.

Assigning an observed specimen (organism) a species name is called *species identification*. The traditional form of species identification relied on guidebooks. For example, books with an *identification key* (see Figure 1.2) represent a decision tree, where each question offers several answers leading to a lower level of the decision tree or directly to a (candidate) species. In *dichotomous keys*, each question about the organism has two possible answers (therefore *dichotomous*). Other popular forms of traditional species identification literature include encyclopedias and atlases of species.



Master Key to Diagnostic Keys

1. Leaves evergreen or persistent over most of winter season, or if leaves absent, a succulent-stemmed species lacking green leaves (such as cacti) Key A
1. Leaves deciduous and absent during winter season; not a succulent 2
  2. Leaf scars opposite or whorled (2 or 3 scars per node) 3
  2. Leaf scars alternate (1 scar per node) or closely clustered 4
3. True terminal bud present Key B
3. True terminal bud lacking Key C
  4. Leaf scar obscured, or bundle scar count indistinguishable Key D
  4. Leaf scar with 1 to several distinguishable bundle scars 5
5. Leaf scar with 1 bundle scar (or 2 in *Ginkgo*), or many small bundle traces closely crowded as to appear as a single scar 6
5. Leaf scar with 3 or more separate bundle scars 7
  6. True terminal bud present Key E
  6. True terminal bud lacking Key F
7. Leaf scar with 3 bundle scars Key G
7. Leaf scar with 4 or more bundle scars Key H

46 · Acer

1. Terminal bud with 2 valvate scales
  2. Twigs finely pubescent; terminal bud sessile to slightly stalked 1. *A. spicatum*
  2. Twigs glabrous; terminal bud distinctly stalked 2. *A. pensylvanicum*
1. Terminal bud with 4 or more imbricate scales
  3. Terminal bud sharply pointed, with 8 or more brownish, acute scales
    4. Twigs rarely over 2mm diam.; terminal buds usually 4mm or less in length; bark gray to glaucous; coastal plain and Piedmont trees (twigs not definitive further)
      5. A tall tree, or straight-trunked if short; leaves glaucous beneath 3. *A. barbatum*
      5. A short tree, stems multiple or crooked; leaves green beneath 4. *A. leucoderme*
    4. Twigs often 3mm or more diam.; terminal buds usually 5mm or more in length; bark gray, brown, or darkened; trees of Appalachians or areas north and west
      6. Twigs brown; terminal buds with glabrous scales or few hairs; bark grayish or light brown on mature trunks 5. *A. saccharum*
      6. Twigs yellow-brown to gray; terminal buds often hairy on tip or with a glaucous bloom on some bud scales; bark dark brown to blackish on mature trunks 6. *A. nigrum*
  3. Terminal bud rather bluntly pointed with 4–6 reddish or greenish scales
    7. Tip of terminal bud hairy between last pair of outer scales; trunk bark finely furrowed
      8. Twigs green or reddish, may have a glaucous bloom; twigs 5–6mm diam.; sap not milky 7. *A. negundo*
      8. Twigs reddish-brown, usually 6–8mm diam.; sap milky 8. *A. platanoides*
    7. Tip of terminal bud not as above; all buds reddish or green on shady side; trunk bark smooth when young, becoming scaly
      9. Twigs with rank odor when scraped 9. *A. saccharinum*
      9. Twigs without rank odor 10. *A. rubrum*

Figure 1.2: Examples from an identification key for woody plants [106]: The identification process starts from level questions (left) and ends with species identification (right).

---

[1] Binomial nomenclature is the two-term naming system: The first term – the *generic name* – identifies the genus and the second term – the *specific name* – identifies the species within the genus.

Modern approaches to species identification include specialised mobile apps and citizen-scientist community websites, such as Encyclopedia of Life (EoL) [2, 142], iNaturalist [5], Pl@ntNet [7, 65], and Atlas of Danish Fungi [1]. Users of such services provide images of the observed specimens, optionally supplemented with additional information such as GPS location, to query for the matching species. The species recommendations are provided by other users - citizen-scientists and biologists - or by a computer vision algorithm.

Through user observations, the aforementioned citizen-science projects bring valuable data back to scientists: collection of data on appearance and occurrence of species and annotations of such data are crucial pillars for biological research focusing on biodiversity, climate change and species extinction [46, 86]. Involvement of citizen communities is a cost effective approach to large scale data collection. Citizen science contributions provide about 50% of all data accessible through the Global Biodiversity Information Facility [26]. This data has a strong taxonomic bias towards birds and mammals [192], leaving data gaps in taxonomic groups such as fungi and insects where species identification is considered non trivial. Species observation datasets collected by the broader public have already proven to add significant value for understanding both basic and more applied aspects of biology (e.g. [10, 138, 197]), and with growing participation in such programs the research potential will increase.

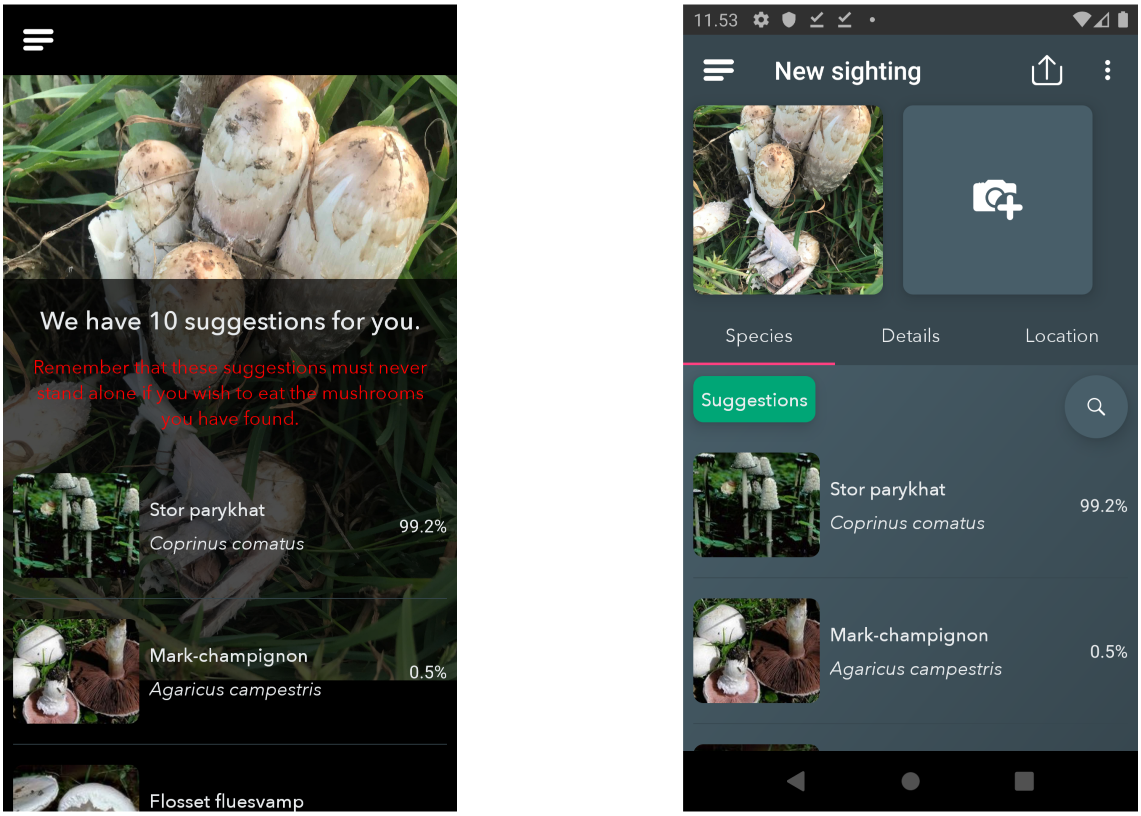
Correct species identification is a challenge in citizen science projects focusing on biodiversity. Some projects handle the issue by simply reducing complexity in the species identification process, e.g. by merging species into multitaxa indicator groups (e.g. [57]), by focusing only on a subset of easily identifiable species or by involving human expert validators in the identification process. Other projects involve the citizen science community in the data validation process. For instance, iNaturalist regards observations as having research grade if three independent users have verified a suggested taxon ID based on an uploaded photo.

The task of species identification is difficult even for human experts with the support of literature. Belhumeur et al. [11] note that the process of identifying a single organism using dichotomous keys may take hours or days, even for specialists (especially in locations with high biodiversity), and is exceedingly difficult to impossible for non-scientists. They propose to assist and speed up the plant identification process with a computer vision based search system.

We are interested in **automatic visual identification of plants and fungi using computer vision** methodology. From the machine learning point of view, the tasks of image-based plant and fungi recognition represent challenging cases of fine-grained classification[2]. Cui et al. [40] define *fine-grained classification* as distinguishing subordinate categories within an entry-level category. While this definition is rather subjective, depending on what we consider "entry-level", recognition of species – the most specific major taxonomic rank – is often used as an example of a fine-grained recognition task. Fine-grained classification often deals with high intra-class variability and very small inter-class

---

[2] Note the inter-disciplinary discrepancy in terminology: In biology, the term *classification* is used for the categorization (grouping) of living organisms, i.e. defining the classes and their hierarchy. In machine learning, *classification* has a different meaning, and is defined [130] as the mapping from inputs $\mathbf{x}$ to outputs $y \in \{1, \ldots, K\}$, i.e. the process of predicting a *class* (category) from a fixed set $\{1, \ldots, K\}$, given data point(s) $\mathbf{x}$ – in our case image(s) of an organism. This thesis will by default use the term *classification* in the later, machine learning, meaning.



We have 10 suggestions for you.

Remember that these suggestions must never stand alone if you wish to eat the mushrooms you have found.

Stor parykhat *Coprinus comatus* 99.2%

Mark-champignon *Agaricus campestris* 0.5%

Flosset fluesvamp

Figure 1.3: A shortlist of species suggestions in the Atlas of Danish Fungi mobile application, using the recognition system described in Chapter 5.

differences. This holds in our case: the appearance of specimens of the same species may vary significantly depending on age, genotype, local conditions, etc.; on the other hand, two species may have similar visual characteristics.

1.1 Problem Formulation

The thesis deals with different species identification tasks: from constrained tasks of recognizing a specific plant organ (leaf, bark) in controlled conditions (leaf on white background, cropped photo of tree bark) to a more complex "in the wild" scenario with unspecified view or organ type, natural background, possible clutter in the scene, etc.

We formulate each of the species identification tasks as a single-label classification problem on a closed set of K classes (in our case, species) $\mathcal{C} = \{1, 2, \ldots, K\}$. That means we assume that each observation (photo) $\mathbf{x}$ belongs to exactly one class (single-label) and that class belongs to C (closed set). Note that we are formulating the task as a flat classification of species, not a hierarchical classification (e.g. following the taxonomic hierarchy).

Given a *training set* $\mathcal{T} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_N, y_N)\}$ of images $\mathbf{x}_i$ labeled with corresponding class labels $y_i \in \mathcal{C}$, our goal is to train a classifier that predicts the unknown class labels $y \in \mathcal{C}$ for new observations $\mathbf{x}$. In cases where human supervision of the result is possible (e.g. in a mobile field guide), the classifier can return a scored list of species recommendations, sorted from the most likely prediction. The user can then display additional information for each species and choose the correct result. See Figure 1.3 for illustration.

We adopt the common metrics and loss functions used in machine learning and in the species recognition benchmarks, such as the top-1 error or the top-k error, i.e. for how many images in the test set is the correct species not among the top-k predictions. For all the metrics used within this thesis, the losses are the same for every species.

Note that while the assumptions we make are quite common for machine-learning definitions of classification problems, they introduce several limitations:

- The classifier does not recognize observations depicting a specimen of an unknown species $y' \notin \mathcal{C}$, and returns the most "similar" species from $\mathcal{C}$ instead.
- The classifier does not recognize observations without a specimen (e.g. plant or mushroom). In the practical application of the fungi recognition system, described later in Chapter 5, we observe that many users test the system by uploading out-of-domain images. See examples in Figure 1.4.
- In practical species recognition applications, the cost for misclassification may be species-dependent. For example, misclassification of two decorative plants is probably not as serious as misclassification of a poisonous fruit for an edible one, which may have very serious consequences.
- In some applications, the loss may depend on the correctness of higher classification ranks and may benefit from a hierarchical classification approach : e.g. if the *species* is not recognized correctly, but at least the *genus* is correct, the loss may be lower than when even a higher taxonomic rank is incorrect.

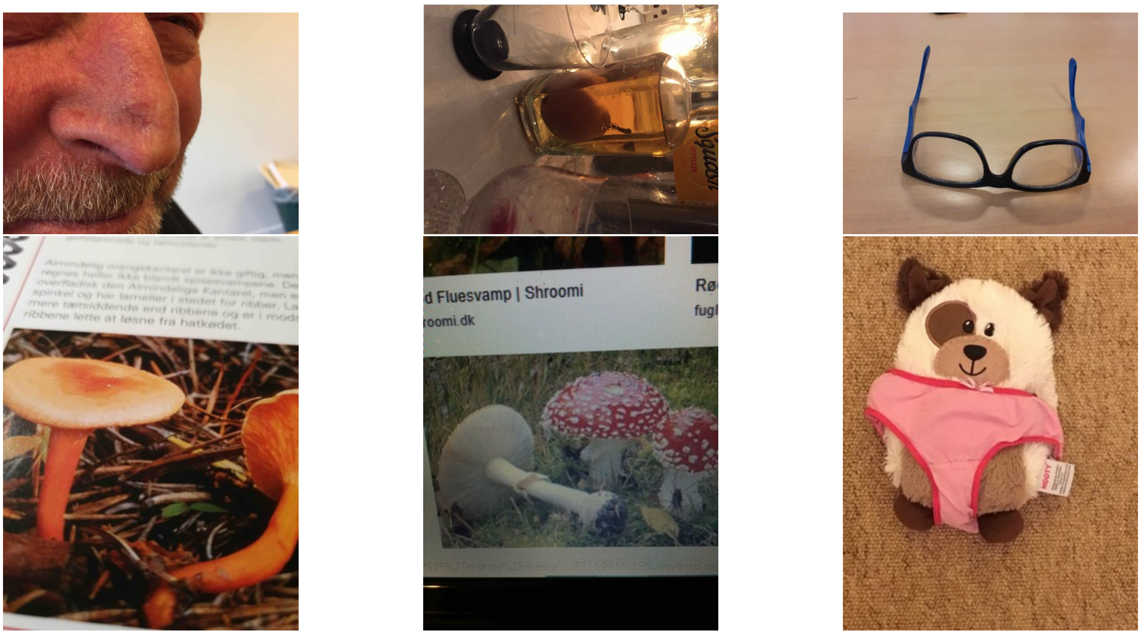
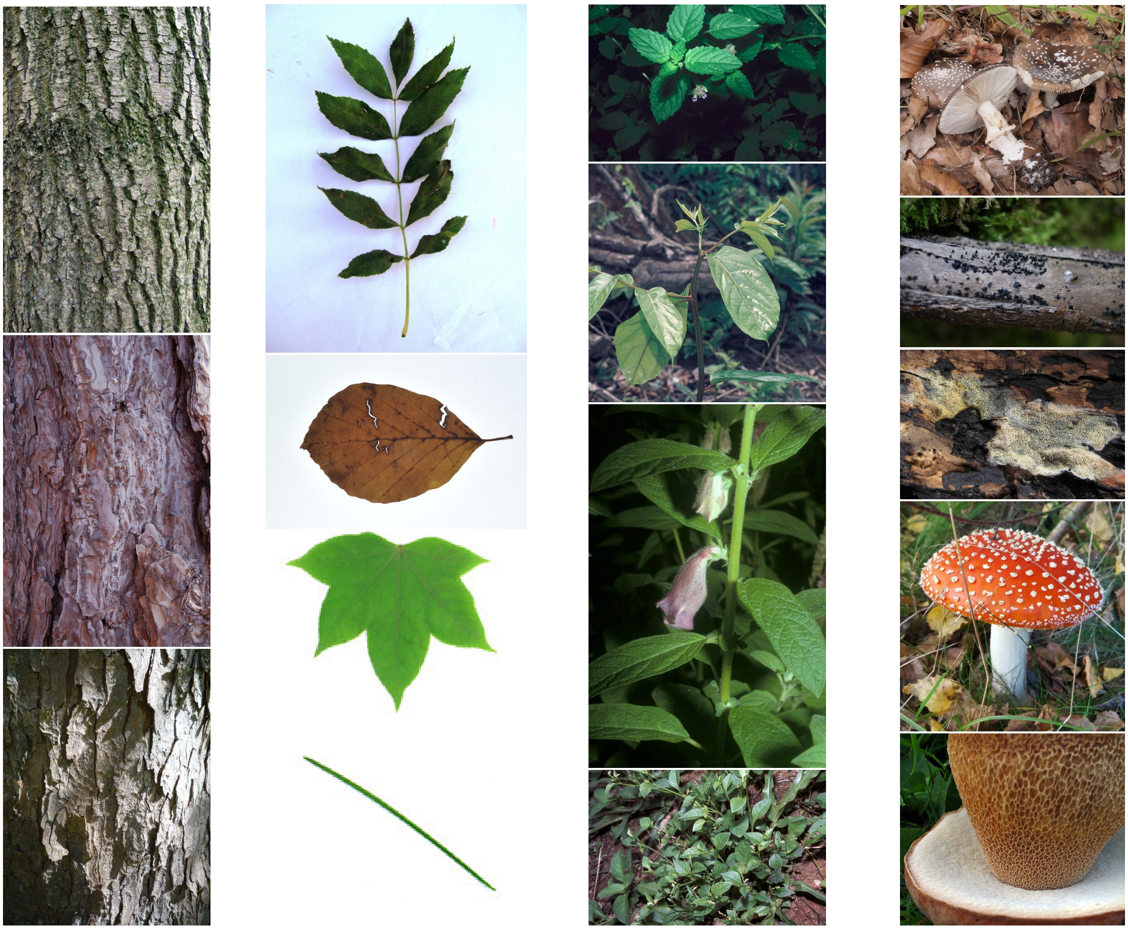


Figure 1.4: Examples of out-of-domain images users are submitting to the fungi recognition service described in Chapter 5. A number of images contain scenes without a mushroom. Some photos include mushrooms displayed on a computer screen or in a book.



(a) Tree bark (b) Leaf scans/photos on white background (c) Plants in the wild (d) Fungi in the wild

Figure 1.5: Examples of inputs for the considered species recognition tasks.

With this formulation of species classification, this thesis deals with the following tasks, differing in the considered set of species $\mathcal{C}$ and in the constraints on the content of images $\mathbf{x}$:

1. Recognition of tree species from a photo of tree bark. We assume the input picture is cropped so that it only captures the bark. See Figure 1.5a for illustration.

2. Recognition of plant species from images of leaves. We assume the input picture contains a leaf scan or a photo of the leaf on white background. See Figure 1.5b for illustration.

3. Recognition of plants species from photos “in the wild”. We only expect the plant or its part (e.g. leaf, fruit, flower,...) is pictured in the photograph. The scenes may contain complex natural background including different forms of clutter. See Figure 1.5c for illustration.

4. Recognition of fungi species from photos “in the wild”. Similarly to the previous task, we only expect the fungus is pictured in the photograph and do not place any further assumptions on the scene. See Figure 1.5d for illustration.

The listed tasks can be further extended by providing a set images of the same specimen observation, with or without the information about the view type (e.g. specifying the photographed plant organ). Additional meta-information such as the GPS coordinates or timestamp can bring useful information for plant identification. Mac Aodha et al. [125] propose a spatio-temporal prior that estimates the probability that a given species occurs a given geographical location and time.

We propose a texture recognition approach to the constrained tasks of leaf and bark recognition; and a deep learning approach to species recognition from "in the wild" photos, where a more complex model is needed. We focus on the problem of possible difference between training- and test-time class categorical priors, where species identification serves as an example of a challenging recognition task with highly unbalanced image datasets. For instance, training examples - images labeled with a species name - can be downloaded from an online encyclopedia, but the number of photographs of a species in the encyclopedia may not correspond to the species incidence in a given geographical location or to the frequency the species is queried in a plant identification service.

1.2 Contributions

The first contribution of the thesis relates to the problems of adjusting the outputs of probabilistic classifiers to new categorical prior probabilities (different from the training set) and estimating the new priors from an unlabeled set of images. In particular, the problem is addressed in the context of deep Convolutional Neural Network (CNN) classifiers. We discuss the interpretation of CNN classifiers trained by cross entropy minimization as estimators of posterior probabilities and experimentally validate some of their properties. For estimation of the new categorical priors, a Maximum Likelihood estimation approach is compared with a proposed Maximum a Posteriori estimator, adding a hyper-prior favouring dense prior distributions. We show that adding such hyper-prior increases the reliability of the estimate and increases the classification accuracy in several fine-grained classification tasks. The results suggest that calibration of over-confident classifiers by temperature scaling impairs some statistical properties of the posterior estimate, decreasing the performance of the prior estimation methods.

The second contribution is the development and evaluation of the state-of-the-art fine-grained classifiers for visual identification of plants and fungi, achieving the best results on several large scale datasets of recent international challenges: ExpertLifeCLEF 2018, FGVCx Fungi 2018, FGVCx Flowers 2018, PlantCLEF 2019. The results confirm the benefits of practices such as combining predictions from an ensemble of models, dealing with noisy labels in the data, data augmentation, saving the moving averages of the trained variables, and adjusting the predictions to new categorical priors. The accuracy of the proposed automatic plant recognition system has been compared against human experts in plant recognition. Experimental results show that the proposed classifiers achieve the human expert level of accuracy.

The third contribution is the application of the competition-winning method in a citizen-science project for fungi recognition, allowing users to get instant species recommendations and increasing the involvement of users in biodiversity data collection.

The fourth contribution of this thesis are the Fast Features Invariant to Rotation and

Scale of Texture (Ffirst) - an extension of a texture recognition method proposed in the author's master thesis [175], introducing new rotational invariants, and additional experiments on texture-based recognition. At the time of publication, the method achieved state-of-the-art results in texture classification as well as in its applications to plant leaf-scan and bark recognition.

As the fifth contribution of the thesis, we studied the importance of color in standard texture-classification datasets and the color-bias of the textures. A proposed improvement to the global color descriptors Color Names [196] and Discriminative Color Descriptors [101] noticeably increased the classification accuracy.

1.3 Structure of the Thesis

The following chapters of the thesis describe different tasks and problems:

Chapter 2 focuses on the constrained tasks of plant species from leaf and bark. We review and extend our previously proposed texture recognition method [174, 175], Fast Features Invariant to Rotation and Scale, for recognition of tree bark and plant leaves. Experiments showed state-of-the-art results among non-CNN-based methods on a number of leaf and bark datasets. The performance of the method is also evaluated on texture recognition datasets and the importance of missing color information is discussed and validated. A comparison with our later deep learning approach shows that the texture-based descriptor still provides competitive results for the constrained tasks of tree bark and plant leaf recognition, as well as the standard texture recognition task.

Chapter 3 introduces a deep learning approach to fine-grained visual classification of plants and fungi "in the wild" and studies the problem of adjusting the output of probabilistic classifiers, including Convolutional Neural Networks, to new a-priori probabilities on the tested data. Recent CNN classifier architectures are reviewed in Section 3.1. A probabilistic interpretation of the classifier outputs and the calibration of over-confident predictions are discussed in 3.2. Section 3.3.1 highlights the importance of adjusting the classifiers in case the new categorical prior probabilities are known. Test-time class prior estimation is addressed in Sections 3.3.2 and 3.3.3, describing an existing Maximum Likelihood approach and a proposed Maximum a Posteriori estimation using the Dirichlet distribution as a hyperprior. The methods are used in our competition submissions in Chapter 4 and evaluated in more detail in Chapter 6.

Our submissions to international challenges are described chronologically in Chapter 4, including the best results in PlantCLEF 2018, FGVCx Flowers 2018, FGVCx Fungi 2018 and PlantCLEF 2019. Sections 4.3 and 4.4 contain a comparison against human experts, showing that CNN classifiers are reaching the human expert level of accuracy in plant identification. Chapter 5 describes the application of the competition-winning method in a citizen-science project for fungi recognition, allowing users to get instant species recommendations and increasing the involvement of users in biodiversity data collection.

The technical overlap between methods based on "hand-crafted" features and the recent CNN classifiers is negligible. For the sake of clarity and readability of the thesis, **the related work is reviewed separately in the corresponding Chapters**.

CHAPTER 2

Texture Recognition Approach to Plant Recognition

We first focus on the recognition of specific plant organs[1], namely tree trunk (bark) and plant leaves. We choose to leverage the textured nature of these organs, and propose to approach plant identification from bark and leaves by texture recognition. While the choice of texture analysis is straightforward for tree bark recognition, it was rather uncommon for the recognition of leaves, where - prior to our work - shape-based approaches were dominating. Note that with uniform background, our texture descriptor also encodes the shapes at the border of the leaf at multiple scales. Related works in leaf recognition, bark recognition and in texture classification are reviewed in Section 2.1.

In order to describe texture independently of the size (distance) of the patterns and of the orientation in the image, a description invariant to rotation and scale is needed. For practical applications, we demand computational efficiency of the texture encoder and classifier. To satisfy the requirements on the method, we first proposed a multi-scale texture recognition method based on Local Binary Patterns and applied it to bark recognition [174], and later further extended and improved it into **Fast Features Invariant to Rotation and Scale of Texture (Ffirst)** [176, 177]. The proposed method is described in detail in Section 2.2.

2.1 Related Work

2.1.1 Leaf Recognition

Before the deep learning era opened the door to the more complex “in the wild” recognition tasks, leaf recognition was by far the most popular approach to plant recognition and a wide range of methods has been reported in the literature [8, 11, 53, 93, 94, 95, 96, 97, 104, 112, 132, 150, 171, 203, 205]. Recognition of leaves usually refers only to recognition of broad leaves, needles are treated separately. Several techniques have been proposed for leaf description,

---

[1] Plant organs include the leaf, stem, root, and reproductive structures.

often based on combining features of different character (shape features, color features, etc.).

A Bag of Words model with Scale Invariant Feature Transform (SIFT) [124] descriptors was applied to leaf recognition by Fiel and Sablatnig [53]. Several shape methods have been compared on leaf recognition by Kadir et al. [93]. Of the compared methods - geometric features, moment invariants, Zernike moments and Polar Fourier Transform - the last performed best on an unpublished dataset.

Kumar et al. [104] describe Leafsnap[2], a computer vision system for automatic plant species identification, which has been developed from the earlier plant identification system by Agarwal et al. [8] and Belhumeur et al. [11]. Kumar et al. [104] introduced a pre-filter on input images, numerous speed-ups and additional post-processing within the segmentation algorithm, the use of a simpler and more efficient curvature-based recognition algorithm. On the introduced Leafsnap database of 184 tree species, their recognition system finds correct matches among the top 5 results for 96.8% queries from the dataset. The resulting electronic Leafsnap field guide is available as a mobile app for iOS devices. The leaf images are processed on a server, internet connection is thus required for recognition, which may cause problems in natural areas with slow or no data connection. Another limit is the need to take the photos of the leaves on a white background.

A publicly available plant leaf database named Flavia was collected by Wu et al. [205], who designed a Probabilistic Neural Network for leaf recognition using 12 Digital Morphological Features, derived from 5 basic features (diameter, physiological length, physiological width, leaf area, leaf perimeter).

Karuna et al. [97] claim that the most valuable features for object recognition are shape and color, and design combination of hand-crafted shape and color features for leaf recognition, achieving 96.5% recognition accuracy on the Flavia dataset.

The Foliage dataset collected by Kadir et al. [94] consists of 60 classes of leaves, each containing 120 images. The best reported result on this dataset reported by Kadir et al. [96] was achieved by a combination of shape, vein, texture and color features processed by Principal Component Analysis before classification by a Probabilistic Neural Network.

Söderkvist [171] proposed a visual classification system of leaves and collected the so called Swedish dataset containing scanned images of 15 classes of Swedish trees. Qi et al. [6] achieve 99.38% accuracy on the Swedish dataset using a texture descriptor called Pairwise Rotation Invariant Co-occurrence Local Binary Patterns [150] with Support Vector Machine (SVM) classification.

A leaf recognition system, using Fourier descriptors of the leaf contour normalised to translation, rotation, scaling and starting point of the boundary, was designed by Novotný and Suk [132]. The authors collected a large leaf dataset called Middle European Woods (MEW) containing 153 classes of native or frequently cultivated trees and shrubs in Central Europe. Their method achieves 84.92% accuracy when the dataset is split into equally sized training and test set. MEW and Leafsnap are the most challenging leaf recognition datasets.

One possible application of leaf description is the identification of a disease. Pydipati et al. [149] proposed a system for citrus disease identification using Color Co-occurrence

---

[2] `http://leafsnap.com/` Last accessed 2nd Apr 2020.

Method (CCM), achieving accuracies of over 95% for 4 classes (normal leaf samples and samples with a greasy spot, melanose, and scab).

2.1.2 Tree Bark Recognition

The problem of automatic tree identification from photos of bark can be naturally formulated as texture recognition.

Several methods have been evaluated on datasets which are not publicly available. Chi et al. [30] proposed a method using Gabor filter banks. Wan et al. [200] performed a comparative study of bark texture features: the grey level run-length method, co-occurrence matrices method, histogram method and auto-correlation method. The authors show that the performance of all classifiers improved significantly when color information was added. Song et al. [172] presented a feature-based method for bark recognition using a combination of Grey-Level Co-Occurrence Matrix (GLCM) and a binary texture feature called Long Connection Length Emphasis. Huang et al. [82] relied on GLCM together with Fractal Dimension Features for bark description. The classification was performed by artificial neural networks.

Since the image data from in the experiments discussed above is not available, it is difficult to assess the quality of the results and to perform comparative evaluation.

Fiel and Sablatnig [53] worked on automated identification of tree species from images of the bark, leaves and needles. For bark description, they created a Bag of Words with SIFT descriptors in combination with GLCM and wavelet features. The vectors were classified by SVM with radial basis function kernel. The authors introduced the Österreichische Bundesforste AG (Austrian Federal Forests) bark dataset consisting of 1182 photos from 11 classes. We refer to this dataset as the AFF (Austrian Federal Forests) bark dataset. Recognition accuracy of 64.2% and 69.7% was achieved on this dataset for training sets with 15 and 30 images per class respectively.

Fiel and Sablatnig describe an experiment with two human experts, a biologist and a forest ranger, both employees of Österreichische Bundesforste AG. Their classification rate on a subset of the dataset with 9 images per class, 99 images in total, was 56.6% (biologist) and 77.8% (forest ranger). This means that the human experts, who probably have much better recognition accuracy “in situ”, usually identify the species based on other features than solely the bark texture.

Boudra et al. [19] review and compare different variants of multi-scale Local Binary Patterns based texture descriptors and evaluate their performance in tree bark image retrieval. The results show that multi-scale Local Binary Patterns (LBP) descriptors, including our variant of MS-LBP [174], outperform the basic LBP and Multi Resolution LBP [134], and that the best results are achieved at the low scale space levels.

2.1.3 Texture Recognition

Texture information is an essential feature for recognition of many plant organs. Texture analysis is a well-established problem with a large number of existing methods, many of them being described in surveys [29, 123, 129, 145, 209]. The notion of texture is hard to define. As noted by Liu et al. [123], the concept of *texture* may have different connotations or definitions depending on the given objective. Existing definitions of visual texture often lack formality and completeness. For illustration, let us quote an informal definition by Hawkins [76]:

Definition 1. *The notion of texture appears to depend upon three ingredients: (1) some local "order" is repeated over a region which is large in comparison the order's size, (2) the order consists in the non-random arrangement of elementary parts, and (3) the parts are roughly uniform entities having approximately the same dimensions everywhere within the textured region.*

Because of the number of existing surveys mentioned above, here we only review some of the most popular and best performing textural features and texture recognition methods.

Several recent approaches to texture recognition report excellent results on standard datasets, many of them working only with image intensity and ignoring the available color information. A number of approaches is based on the popular Local Binary Patterns (LBP) [135, 136], such as the recent Pairwise Rotation Invariant Co-occurrence Local Binary Patterns of Qi et al. [150] or the Histogram Fourier Features of Ahonen et al. [9, 210]. A cascade of invariants computed by scattering transforms was proposed by Sifre and Mallat [169] in order to construct an affine invariant texture representation. Mao et al. [128] use a bag-of-words model with a dictionary of so called active patches: raw intensity patches that undergo further spatial transformations and adjust themselves to best match the image regions. While the Active Patch Model does not use color information, the authors claim that adding color will further improve the results. The method of Cimpoi et al. [32] using Improved Fisher Vectors (IFV) for texture description shows further improvement when combined with describable texture attributes learned on the Describable Textures Dataset (DTD) and with color attributes.

Cimpoi et al. [33, 34] pushed the state-of-the-art in texture recognition using a new encoder denoted as FV-CNN-VD, obtained by Fisher Vector pooling of a very deep Convolutional Neural Network (CNN) filter bank pre-trained on ImageNet by Simonyan and Zisserman [170]. The CNN filter bank operates conventionally on preprocessed RGB images. This approach achieves state-of-the-art accuracy, yet due to the size of the very deep VGG networks it may not be suitable for real-time applications when evaluated without a high-performance Graphics Processing Unit (GPU) for massive parallelization.

Bello-Cerezo et al. [12] compared several hand-crafted texture descriptors against off-the-shelf CNN-based features for (color) texture classification in 2019. In terms of classification accuracy, most experiments indicate the superiority of deep networks, however, hand-crafted descriptors still performed better than CNN-based features in cases with little intra-class variability or in cases where the variability can be modelled explicitly (e.g. rotations handled by rotation-invariant texture descriptors). Liu et al. [123] remark that while CNNs generally outperform classical texture descriptors, it remains to be seen which approaches will be most effective in resource-limited contexts.

Note that most of the methodology in this chapter was developed prior to the publication of results with CNN-based features [12, 33, 123]. However, Section 2.4.3 contains a comparison of the proposed method with a state-of-the-art CNN classifier, that achieves almost perfect leaf recognition accuracy.

2.2 Fast Features Invariant to Rotation and Scale of Texture

2.2.1 Completed Local Binary Pattern and Histogram Fourier Features

The Ffirst description is based on the Local Binary Patterns [134, 135, 136]. The common LBP operator (later denoted as sign-LBP) locally computes the signs of differences between the center pixel and its P neighbours on a circle of radius R. With an image function $f(x, y)$ and neighbourhood point coordinates (x_p, y_p):

$$\begin{aligned} \mathrm{LBP}_{P,R}(x,y) &= \sum_{p=0}^{P-1} s(f(x,y) - f(x_p,y_p))2^p, \\ s(z) &= \begin{cases} 1, & \text{if } z \le 0, \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \tag{2.1}$$

To achieve rotation invariance[3], we adopt the so called LBP Histogram Fourier Features (LBP-HF) introduced by Ahonen et al. [9]. LBP-HF describe the histogram of uniform patterns using coefficients of the Discrete Fourier Transform (DFT). Uniform LBP are patterns with at most 2 spatial transitions (bitwise 0-1 changes). Unlike the simple rotation invariants using $\mathrm{LBP}^{\mathrm{ri}}$ [134, 146], which maps all uniform patterns with the same number of 1s into one bin, the LBP-HF features preserve the information about relative rotation of the patterns.

Denoting a uniform pattern $U_p^{n,r}$, where n is the "orbit" number corresponding to the number of "1" bits and r denotes the rotation of the pattern, the DFT for given n is expressed as:

$$H(n,u) = \sum_{r=0}^{P-1} h_I(U_p^{n,r}) e^{-\mathrm{i}\, 2\pi u r / P}, \tag{2.2}$$

where the histogram value $h_I(U_p^{n,r})$ denotes the number of occurrences of a given uniform pattern in the image.

The LBP-HF features are equal to the absolute value of the DFT magnitudes, and thus are not influenced by the phase shift caused by rotation).

$$\text{LBP-HF}(n,u) = |H(n,u)| = \sqrt{H(n,u)\overline{H(n,u)}}. \tag{2.3}$$

Since h_I are real, $H(n,u) = H(n, P-u)$ for $u = (1, \ldots, P-1)$, and therefore only $\lfloor \frac{P}{2} \rfloor + 1$ of the DFT magnitudes are used for each set of uniform patterns with n "1" bits for $0 < n < P$. Three other bins are added to the resulting representation, namely two for the "1-uniform" patterns (with all bins of the same value) and one for all non-uniform patterns.

The LBP histogram Fourier features can be generalized to any set of uniform patterns. In Ffirst, the LBP-HF-S-M description [210] is used, where the histogram Fourier features

---

[3] LBP-HF (as well as LBP^{ri}) are rotation invariant only in the sense of a circular bit-wise shift, such as rotation by multiples of 45° for $\mathrm{LBP}_{8,R}$. However, with some image rotations, sampling from other pixels may break the rotation invariance.

of both sign- and magnitude-LBP are calculated to build the descriptor. The magnitude-LBP [71] checks if the magnitude of the difference of the neighbouring pixel (x_p, y_p) against the central pixel (x, y) exceeds a threshold t_p:

$$\text{LBP-M}_{P,R}(x, y) = \sum_{p=0}^{P-1} s(|f(x, y) - f(x_p, y_p)| - t_p)2^p. \tag{2.4}$$

We adopted the common practice of choosing the threshold value (for neighbours at p-th bit) as the mean value of all m absolute differences in the whole image:

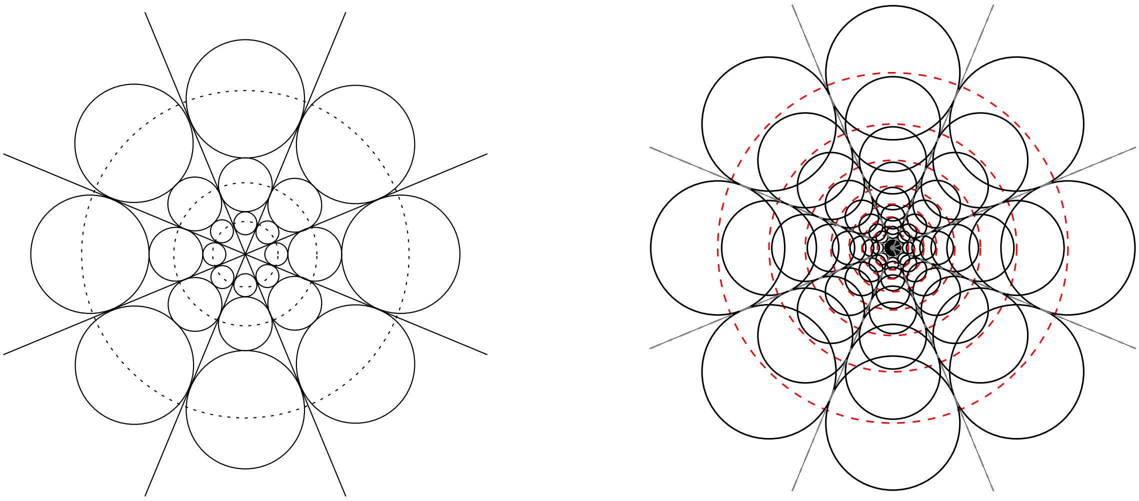
$$t_p = \sum_{i=1}^{m} \frac{|f(x_i, y_i) - f(x_{ip}, y_{ip})|}{m}. \tag{2.5}$$

The LBP-HF-S-M histogram is created by concatenating histograms of LBP-HF-S and LBP-HF-M (computed from uniform sign-LBP and magnitude-LBP).

2.2.2 Multi-scale Description and Scale Invariance

A scale space is built by computing LBP-HF-S-M from circular neighbourhoods with exponentially growing radius R. Gaussian filtering is used[4] to overcome noise.

Unlike the MS-LBP approach of Mäenpää and Pietikäinen [126], where the radii of the LBP operators are chosen so that the effective areas of different scales touch each other, Ffirst uses a finer scaling with a step of $\sqrt{2}$ between scales radii R_i, i.e. $R_i = R_{i-1}\sqrt{2}$. This radius change is equivalent to decreasing the image area to one half. The first LBP radius used is $R_1 = 1$, as the LBP with low radii capture important high frequency texture characteristics. Figure 2.1b displays the scale space of MS-LBP [126] and the scale space of Ffirst.



(a) Scale space from [126]

(b) Scale space used in Ffirst

Figure 2.1: The effective areas of filtered pixel samples in a multi-resolution $\text{LBP}_{8,R}$ operator.

---

[4] The Gaussian filtering is used for a scale i only if $\sigma_i > 0.6$, as filtering with lower σ_i leads to significant loss of information.

Similarly to [126], the filters are designed so that most of their mass lies within an effective area of radius r_i. We select the effective area diameter, such that the effective areas at the same scale touch each other: $r_i = R_i \sin \frac{\pi}{P}$.

LBP-HF-S-M histograms from c adjacent scales are concatenated into a single descriptor. Robustness to scale changes is increased by creating n_{conc} multi-scale descriptors for one image. See Algorithm 17 for the overview of the texture description method.

Algorithm 1 The Ffirst descriptor method in pseudocode.

```
function DESCRIPTOR(img, $n_{\text{conc}}$, $c$)
    $R_1 := 1$;
    for all scales $i := 1, \ldots, (n_{\text{conc}} + c - 1)$ do
        $\sigma_i := R_i \sin \frac{\pi}{P} / 1.3$
        if $\sigma_i > 0.6$ then
            imgB := gaussBlur(img, $\sigma_i$)
        on the original image
        end if
        extract LBP$_{P,R_i}$-S and LBP$_{P,R_i}$-M
        build LBP$_{P,R_i}$-HF-S-M
        for $j := 1, \ldots, n_{\text{conc}}$ do
            if $i \geq j$ and $i < j + c$ then
                attach LBP$_{P,R_i}$-HF-S-M
            to $j$-th multi-scale descriptor
            end if
        end for
        $R_{i+1} := R_i \sqrt{2}$
    end forreturn descriptors
end function
```

2.2.3 Support Vector Machine and Feature Maps

In most applications, a Support Vector Machine (SVM) classifier with a suitable non-linear kernel provides higher recognition accuracy then with a linear kernel, at the price of significantly higher time complexity and higher storage demands (dependent on the number of support vectors). An approach for efficient use of additive kernels via explicit feature maps is described by Vedaldi and Zisserman [199] and can be combined with a linear SVM classifier. Using linear SVMs on feature-mapped data improves the recognition accuracy, while preserving linear SVM advantages like fast evaluation and low storage (independent on the number of support vectors), which are both very practical in real time applications. In Ffirst we use the explicit feature map approximation of the histogram intersection kernel, although the χ^2 kernel leads to similar results.

The "One versus All" classification scheme is used for multi-class classification, implementing the Platt's probabilistic output [117, 147] to ensure SVM results comparability among classes. The maximal posterior probability estimate over all scales is used to determine the resulting class.

In our experiments we use a Stochastic Dual Coordinate Ascent [165] linear SVM solver implemented in the VLFeat library [198].

2.2.4 Adding Rotational Invariants

The LBP-HF features used in the proposed Ffirst description are usually built from the DFT magnitudes of differently rotated uniform patterns. We propose to use all LBP instead of just the subset of uniform patterns. Note that in this case, some orbits have a lower number of patterns, since some non-uniform patterns show symmetries, as illustrated in Figure 2.2.

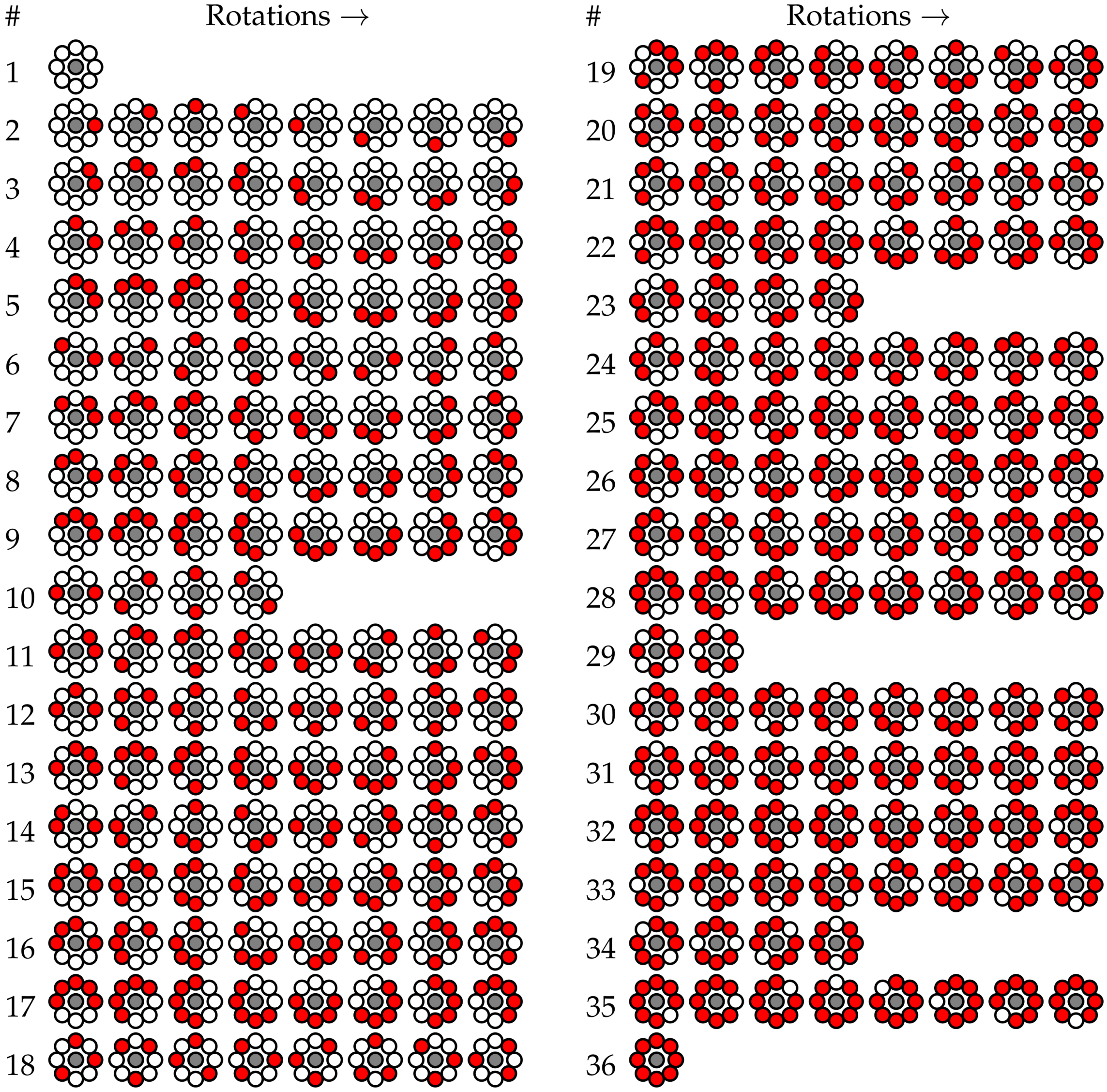


Figure 2.2: The full set of Local Binary Patterns divided into 36 orbits for the Histogram Fourier features. Patterns in one orbit only differ by rotation.

Another rotational invariants are computed from the first DFT coefficients for each orbit:

$$\text{LBP-HF}^{+}(n) = \sqrt{H(n,1)\overline{H(n+1,1)}} \tag{2.6}$$

Ffirst$^{\forall+}$ denotes the method using the full set of patterns for LBP-HF features and adding the additional LBP-HF^{+} features.

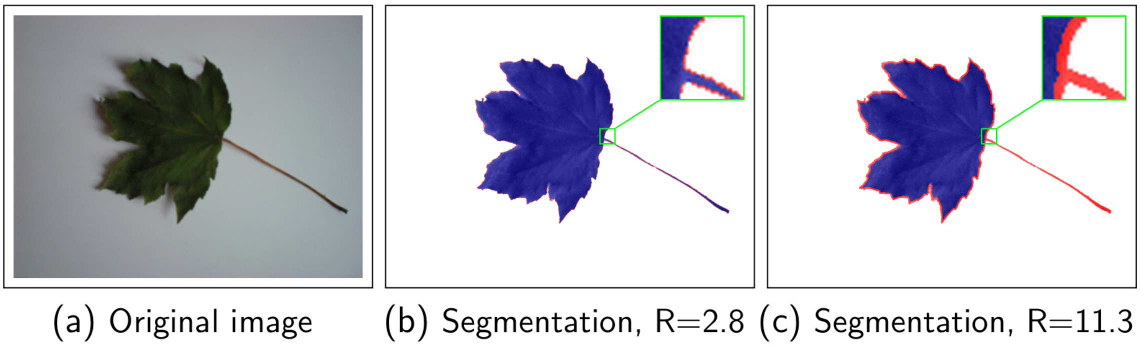

(a) Original image (b) Segmentation, R=2.8 (c) Segmentation, R=11.3

Figure 2.3: Segmentation of the leaf interior (blue) and border region (red) at different scales given by LBP radius R. The border region is defined as all points which have at least one neighbour (in $LBP_{P,R}$) outside of the segmented region.

2.2.5 Recognition of Segmented Textural Objects

We propose to extend Ffirst to segmented textural objects by treating the border and the interior of the object segment separately.

Let us consider a segmented object region $\mathbb{A}$. One may describe only points that have all neighbours at given scale inside $\mathbb{A}$. We show that describing a correctly segmented border, i.e. points in $\mathbb{A}$ with one or more neighbours outside $\mathbb{A}$ (see Figure 2.3), adds additional discriminative information.

We experiment with 5 variants of the recognition method, differing in the processing of the border region:

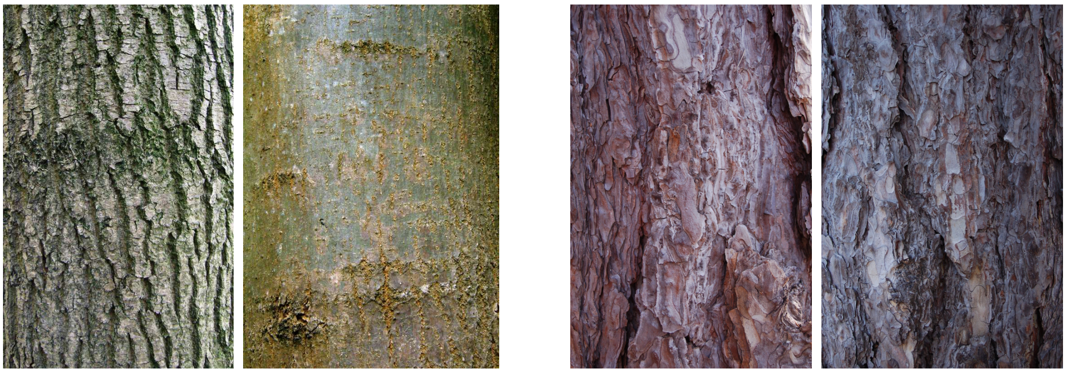
1. Ffirst_a describes all pixels in $\mathbb{A}$ and selects the multi-scale descriptor (one of n_conc) that maximizes the posterior probability estimate, i.e. SVM Platt's probabilistic output.
2. Ffirst_i describes only the segment interior, i.e. pixels in $\mathbb{A}$ with all neighbours in $\mathbb{A}$.
3. Ffirst_b describes only the segment border, i.e. pixels in $\mathbb{A}$ with at least one neighbour outside $\mathbb{A}$.
4. $\text{Ffirst}_{\text{ib}\sum}$ combines the Ffirst_i and Ffirst_b descriptors and selects the multi-scale descriptors that maximize the sum of their posterior probability estimates.
5. $\text{Ffirst}_{\text{ib}\prod}$ combines the Ffirst_i and Ffirst_b descriptors and selects the multi-scale descriptors that maximize the product of their posterior probability estimates .

The leaf databases contain images of leaves on an almost white background. Segmentation was done by thresholding using the Otsu's method [139].

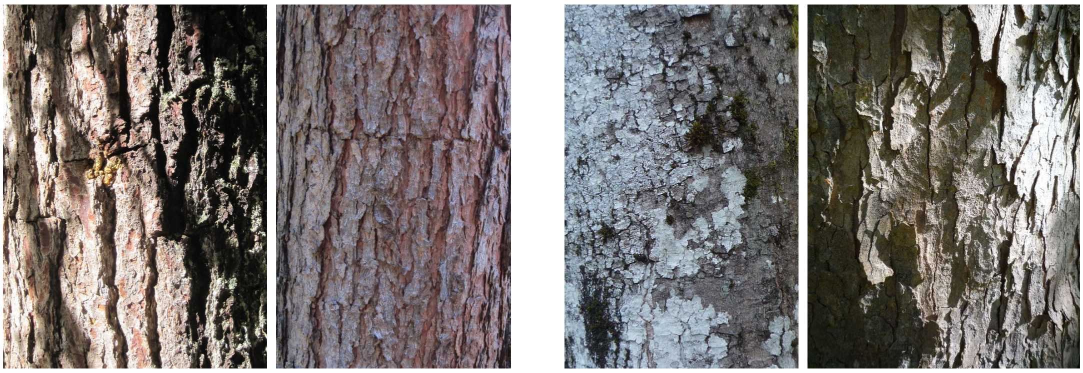
2.3 Datasets and Evaluation Methodology

2.3.1 Tree Bark Dataset

Bark recognition is evaluated on a dataset collected by *Österreichische Bundesforste – Austrian Federal Forests*, which was introduced in 2010 by Fiel and Sablatnig [52] and contains 1182 bark images from 11 classes. We denote it as **the Austrian Federal Forests (AFF) bark dataset**. The resolution of the images varies (between 0.4 Mpx and 8.0 Mpx). This dataset is not publicly available, but it was kindly provided by the Computer Vision Lab, TU Vienna, for academic purposes, with courtesy by Österreichische Bundesforste/Archiv.



(a) Ash (b) Black pine



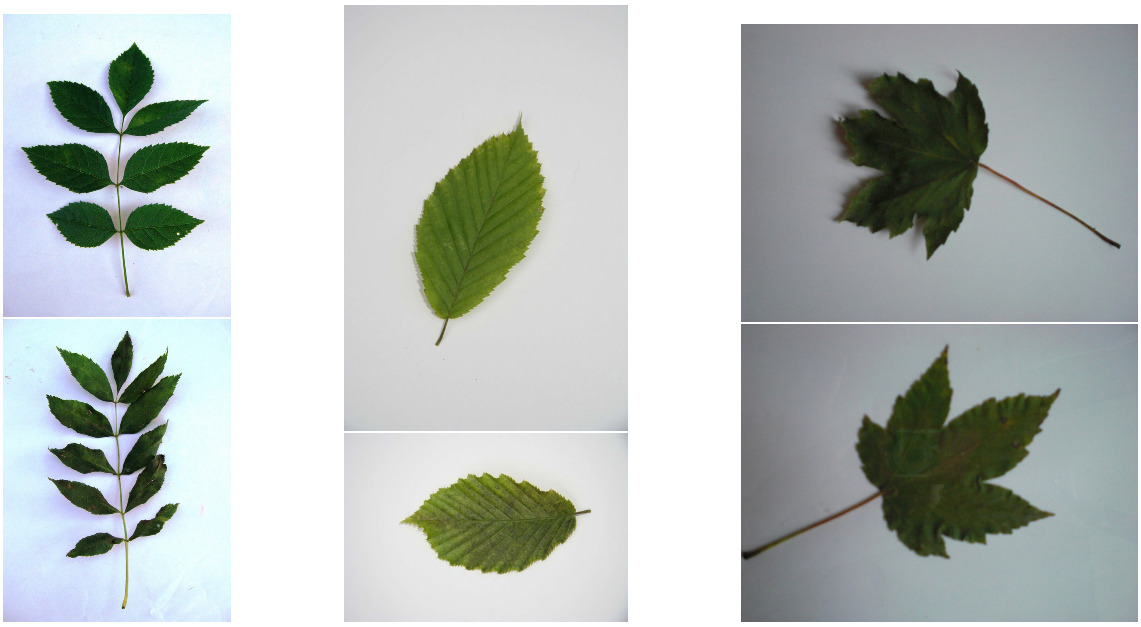
(c) Swiss stone pine (d) Sycamore maple

Figure 2.4: Examples of 4 tree species from the AFF bark database.

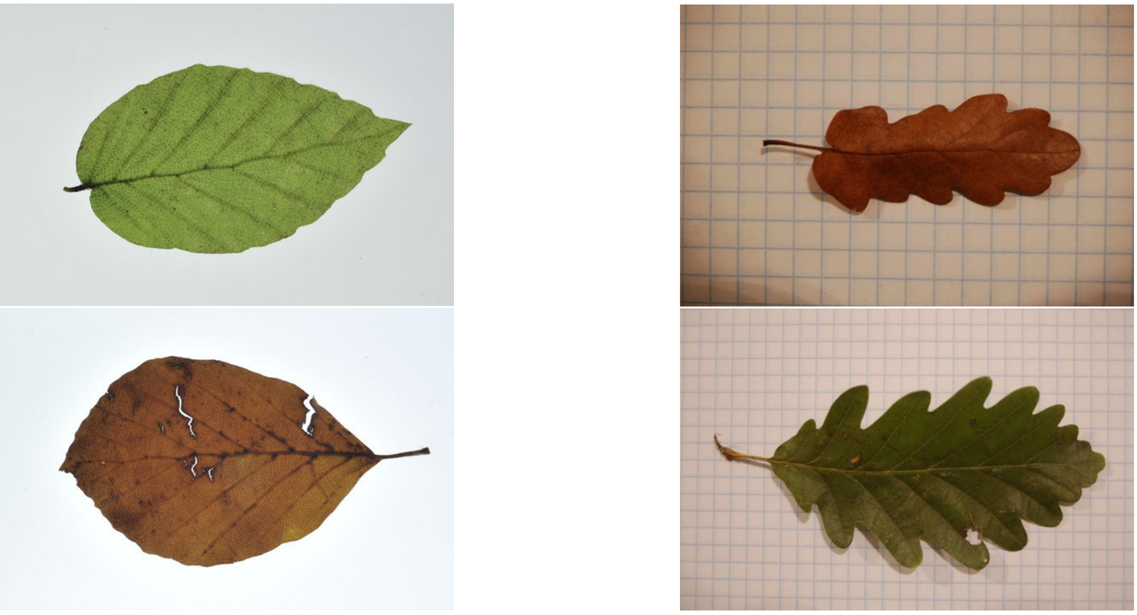
2.3.2 Leaf Datasets

Unlike in bark recognition, there is a number of existing datasets for leaf classification, most of them being publicly available. The datasets and their experimental settings are briefly described bellow:

The Austrian Federal Forest (AFF) leaf dataset was used by Fiel and Sablatnig [53] for recognition of trees, and was kindly provided together with the bark dataset described previously. It contains 134 photos of leaves of the 5 most common Austrian broad leaf trees. The leaves are placed on a white background. The results are compared using the protocol of Fiel and Sablatnig, i.e. using 8 training images per leaf class.



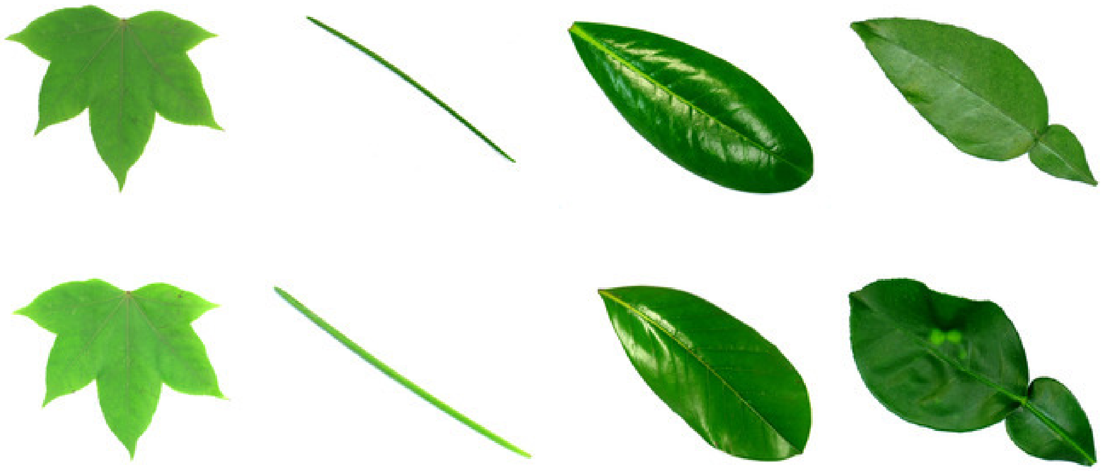
(a) Ash (b) Hornbeam (c) Sycamore maple



(d) Beech (e) Mountain oak

Figure 2.5: Examples from the AFF leaf dataset.

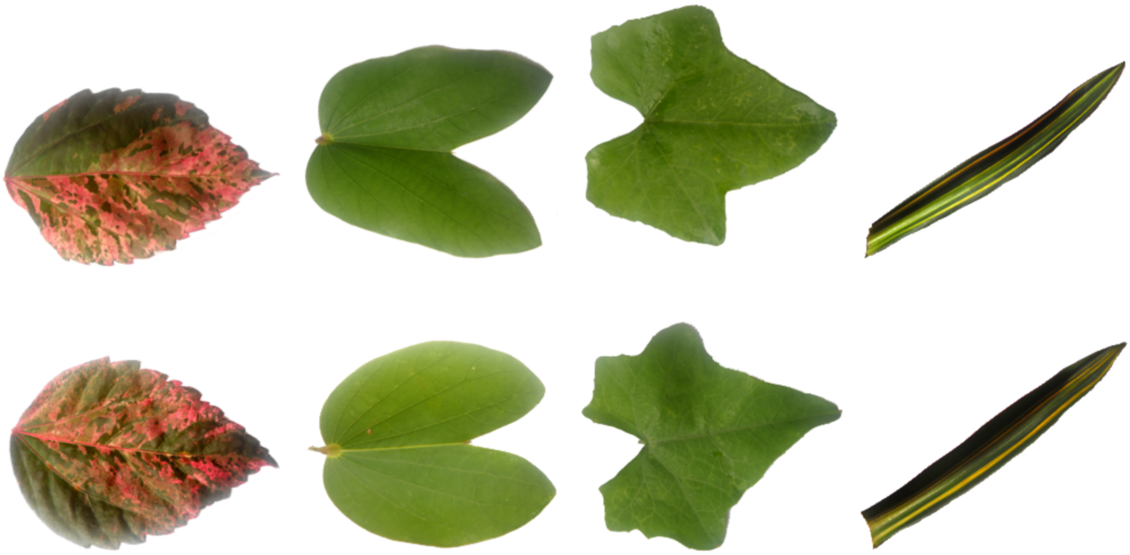
The Flavia leaf dataset contains 1907 images (1600x1200 px) of leaves from 32 plant species on white background, 50 to 77 images per class. The dataset was introduced by Wu et al. [205], who used 10 images per class for testing and the rest of the images for training. More recent publications use 10 randomly selected test images and 40 randomly selected training images per class, achieving better recognition accuracy even with the lower number of training samples. In the case of the best result reported by Lee et al. [112], 10 images per species are used for testing, but the number of training samples is not clearly stated. Some authors divide the set of images for each class into two halves, one for training and the other for testing.



(a) Castor aralia (b) Deodar (c) Southern magnolia (d) Tangerine

Figure 2.6: Examples of 4 classes from the Flavia leaf dataset.

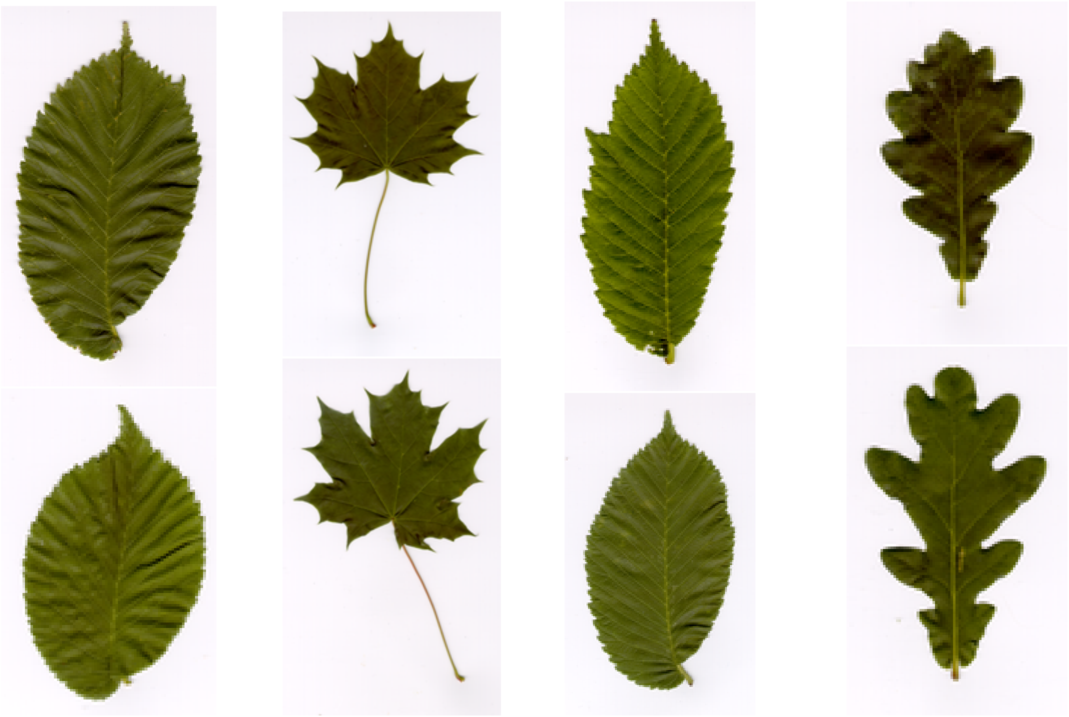
The Foliage leaf dataset by Kadir et al. [94, 95] contains 60 classes of leaves from 58 species. The dataset is divided into a training set with 100 images per class and a test set with 20 images per class.



(a) Hibiscus rosa-sinensis (b) Bauhinia acuminata (c) Ipomoea lacunose (d) Tradescantia spathacea "Vittata"

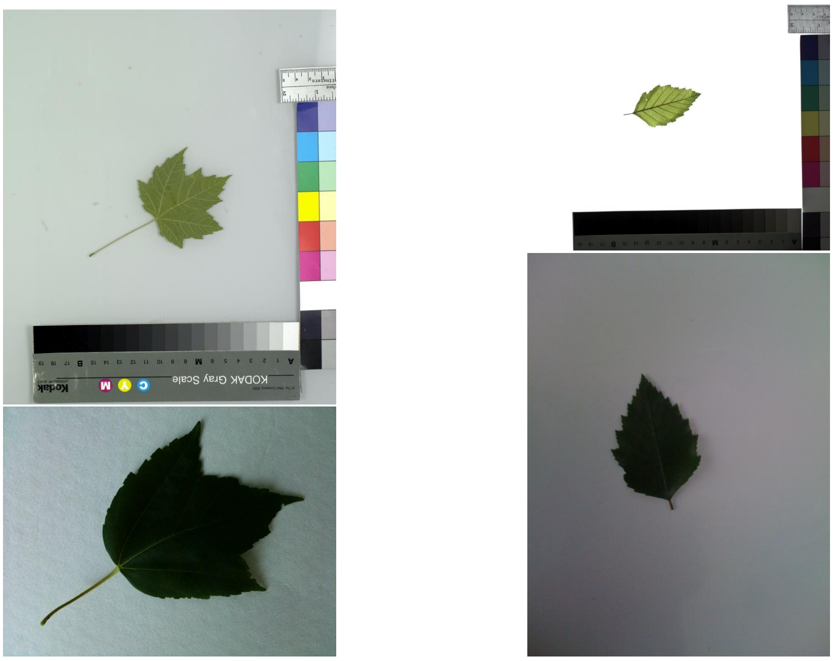
Figure 2.7: Examples of 4 classes from the Foliage dataset.

The Swedish leaf dataset was introduced in Söderkvist's diploma thesis [171] and contains images of leaves scanned using a 300 dpi color scanner. There are 75 images for each of 15 tree classes. The standard evaluation scheme uses 25 images for training and the remaining 50 for testing. Note: The best reported result of Qi et al. [150] is presented on the project homepage [6], not in the original paper [150].



(a) Ulmus carpinifolia (b) Acer (c) Salix aurita (d) Quercus

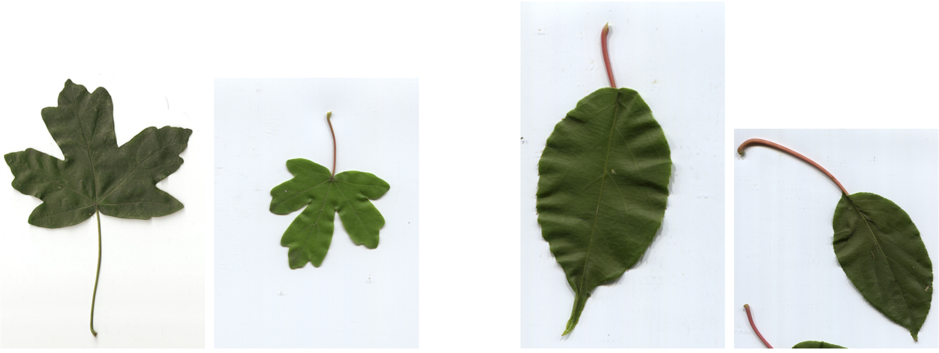
Figure 2.8: Examples of 4 classes from the Swedish dataset.



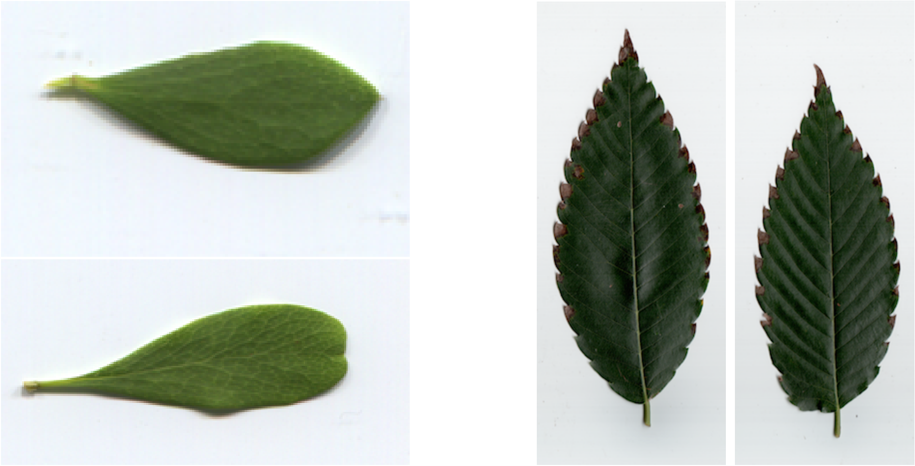
(a) Acer rubrum (b) Betula nigra

Figure 2.9: Examples from the Leafsnap dataset - Lab (top) and Field (bottom) images.

The Leafsnap dataset version 1.0 by Kumar et al. [104] was publicly released in 2014. It covers 185 tree species from the Northeastern United States. It contains 23 147 high quality Lab images and 7 719 Field images. The authors of Leafsnap note that the released dataset does not exactly match that used to compute results for the paper, nor the currently running version on their servers, yet it seems to be similar to the dataset used in [104] and should allow at least a rough comparison. In the experiments of [104], leave-one-image-out species identification has been performed, using only the Field images as queries, matching against all other Field and Lab images. Probability of the correct match appearing among the top 5 results is taken as the resulting score. Note: The score of [104] for the top-1 accuracy in Table 2.4 is estimated from a figure in [104]. Because leave-one-image-out testing scheme would demand to re-train our classifiers for each tested image, we rather perform 10-fold cross validation, i.e. divide the set of Fields images into 10 parts, testing each part on classifiers learned using the set of other parts together with the Lab images.



(a) Acer campestre (b) Actinidia arguta



(c) Berberis thunbergii (d) Zelkova serrata

Figure 2.10: Examples of 4 classes from the MEW dataset.

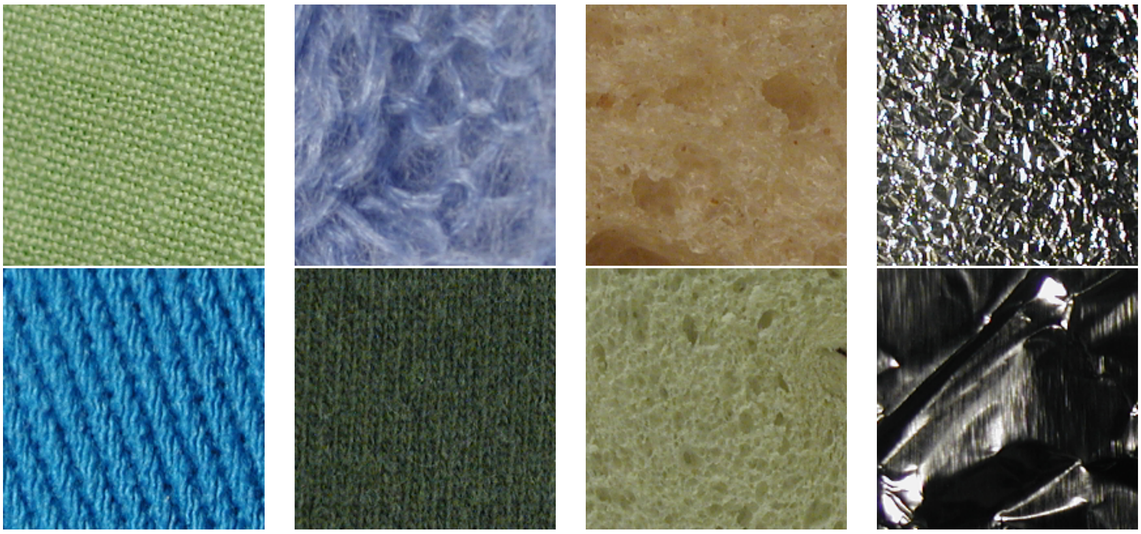
The Middle European Woods (MEW) dataset was introduced by Novotný and Suk [132]. It contains 300 dpi scans of leaves belonging to 153 classes (from 151 botanical species) of Central European trees and shrubs. There are 9745 samples in total, at least 50 per class. The experiments are performed using half of the images in each class for training and the other half for testing.

2.3.3 Texture Recognition Datasets

The Ffirst method for texture classification is tested using the standard evaluation protocols on the following texture datasets:

The KTH-TIPS texture database [54, 77] contains images of 10 materials. There are 81 images (200x200 px) of each material with different combination of pose, illumination and scale.

The standard evaluation protocol on the KTH-TIPS dataset uses 40 training images per material.



(a) Cotton (b) Wool (c) White bread (d) Aluminium foil

Figure 2.11: Examples of 4 texture classes from the KTH-TIPS2 database.

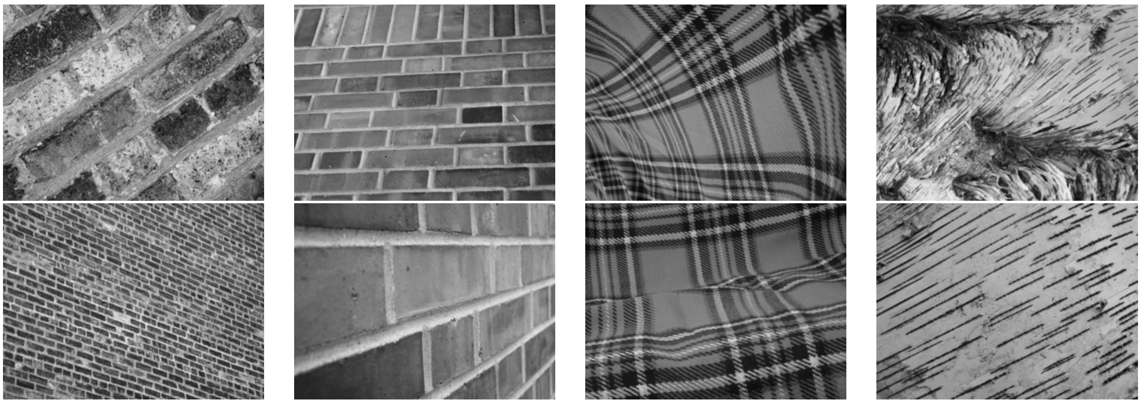
The KTH-TIPS2 database was published [25,127] shortly after KTH-TIPS. It builds on the KTH-TIPS database, but provides multiple sets of images - denoted as “samples“ - per material class (examples in Figure 2.11).

There are 4 “samples“ for each of the 11 materials in the KTH-TIPS2 database, containing 108 images per “sample“ (again with different combination of pose, illumination and scale). However, in the first version of this dataset, for 4 of those 44 “samples“ only 72 images were used. This first version is usually denoted as KTH-TIPSa, and the standard evaluation method uses 3 “samples“ from each class for training and 1 for testing. The “complete” version of this database, KTH-TIPSb, is usually trained only on 1 “sample“ per class and tested on the remaining 3 “samples“.

The Brodatz32 dataset [193] was published in 1998 and it contains low resolution (64x64 px) grey-scale images of 32 textures from the photographs published by Phil Brodatz [22] in 1966, with artificially added rotation (90°) and scale change (a 64x64 px scaled block obtained from 45x45 pixels in the middle). There are 64 images for each texture class in total. Even though the original images are copyrighted and the legality of their usage in academic publications is unclear[5], Brodatz textures are one of the most popular and broadly used sets in texture analysis.

---

[5] `http://graphics.stanford.edu/projects/texture/faq/brodatz.html` Last accessed 2nd Apr 2020.



(a) Brick 1 (b) Brick 2 (c) Plaid (d) Bark 3

Figure 2.12: Examples of 4 texture classes from the UIUCTex database.

The standard protocol for the Brodatz32 dataset simply divides the data into two halves (i.e. 32 images per class in the training set and 32 in the test set).

The UIUCTex database, sometimes referred to as the Ponce Group Texture Database, was published by Lazebnik et al. [108] in 2005 and features 25 different texture classes, 40 samples each. All images are in VGA resolution (640x480 px) and in grey-scale.

The surfaces included in the database are of various nature (wood, marble, gravel, fur, carpet, brick, ..) and were acquired with significant viewpoint, scale and illumination changes and additional sources of variability, including, but not limited to, non-rigid material deformations (fur, fabric, and water) and viewpoint-dependent appearance variations (glass). Examples of images from different classes are in Figure 2.12.

The results on this dataset are usually evaluated using 20 or 10 training images per class. In our experiments, the former case with a larger training set is performed.

The UMD dataset [207] consists of 1 000 uncalibrated, unregistered grey-scale images of size 1280x960 px, 40 images for each of 25 different textures. The UMD database contains non-traditional textures like images of fruits, shelves of bottles and buckets, various plants, or floor textures.

The standard evaluation protocol for UMD is dividing the data into two halves (i.e. 20 images per class in the training set and 20 in the test set).

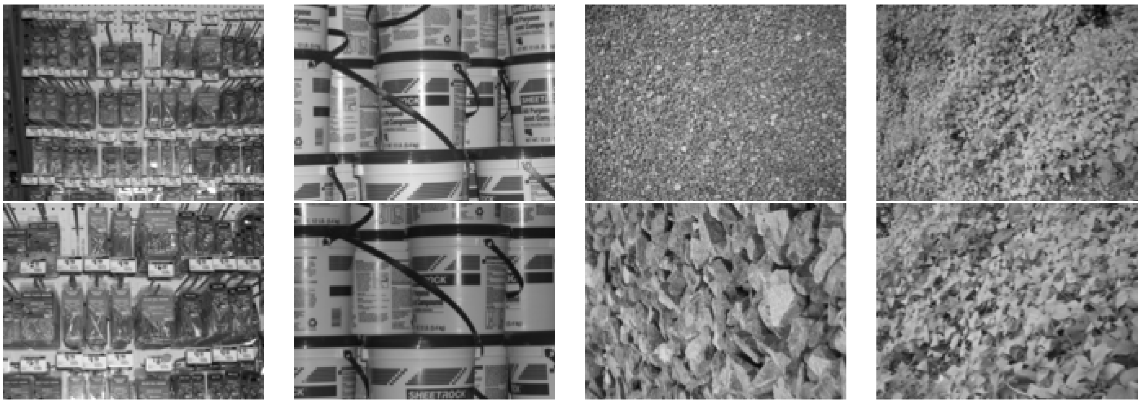
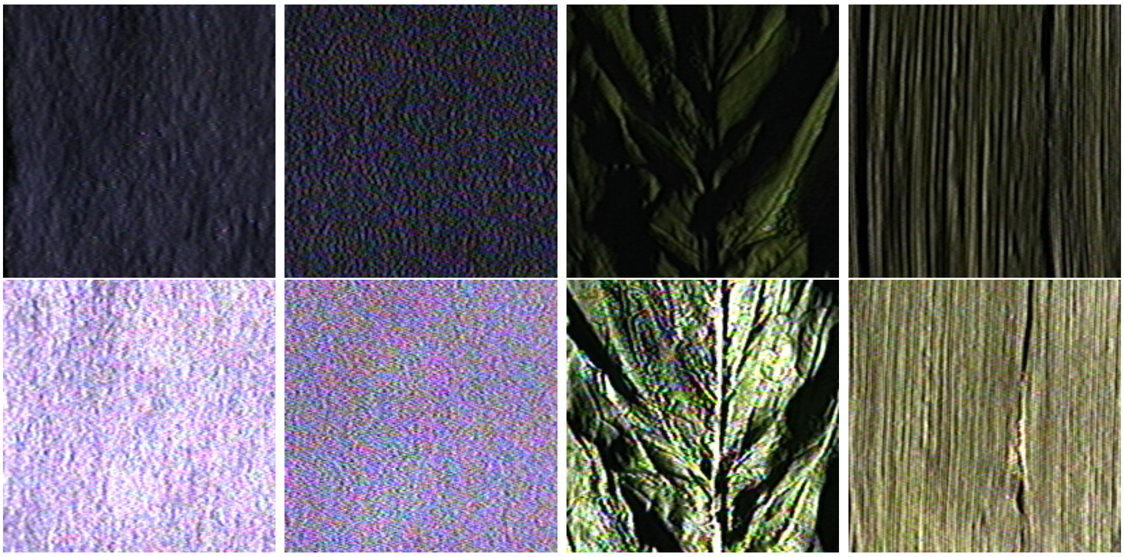


Figure 2.13: Examples of 4 texture classes from the UMD database.



(a) Felt (b) Polyester (c) Lettuce Leaf (d) Corn Husk

Figure 2.14: Examples of 4 texture classes from the CUReT database.

The CUReT image database [41] contains textures from 61 classes, each observed with 205 different combinations of viewing and illumination directions. In the commonly used version, denoted as the cropped CUReT database[6], only 92 images are chosen, for which a sufficiently large region of texture is visible across all materials. A central 200x200 px region is cropped from each of these images, discarding the remaining background. There are thus 61x92=5612 images in the cropped database.

CUReT also contains a BRDF (bidirectional reflectance distribution function) database. For the purpose of standard texture recognition methods, only the image database is used. We use 46 training images per class according to the standard evaluation protocol for the CUReT database.

The Amsterdam Library of Textures [23], denoted as ALOT, contains 250 texture classes. Each class contains 100 images obtained with different combinations of viewing and illumination directions and illumination color. To compare our results on the ALOT dataset to the state-of-the-art [152] we use 20 training images and 80 test images per class.

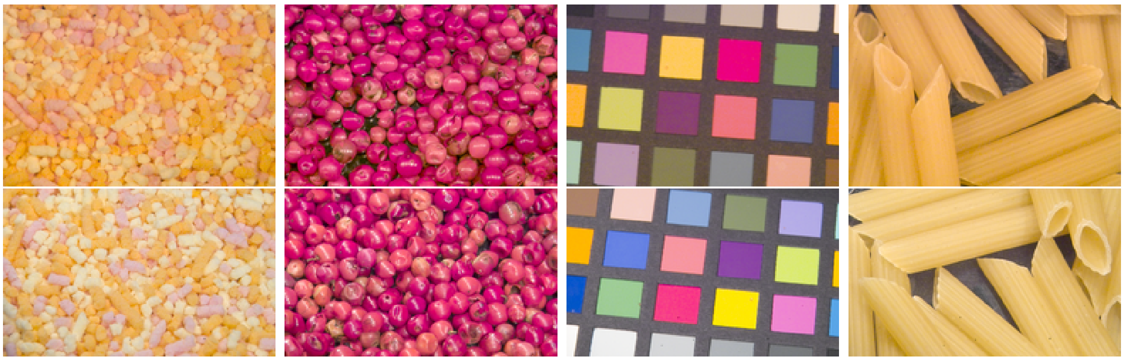
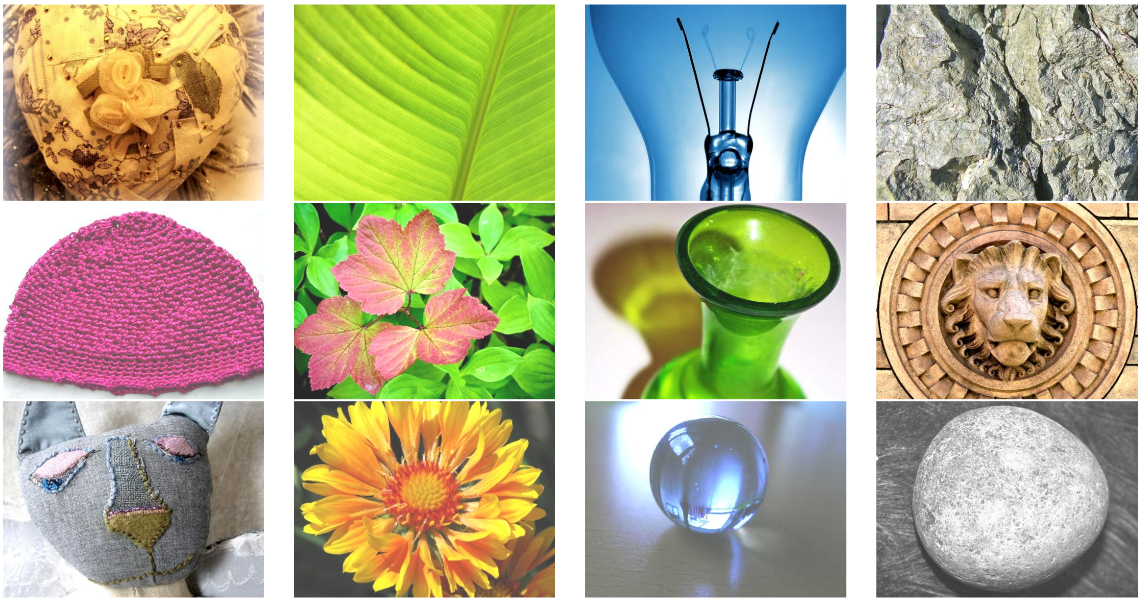


Figure 2.15: Examples of 4 texture classes from the ALOT database.

---

[6] http://www.robots.ox.ac.uk/~vgg/research/texclass/setup.html Last accessed 2nd Apr 2020.



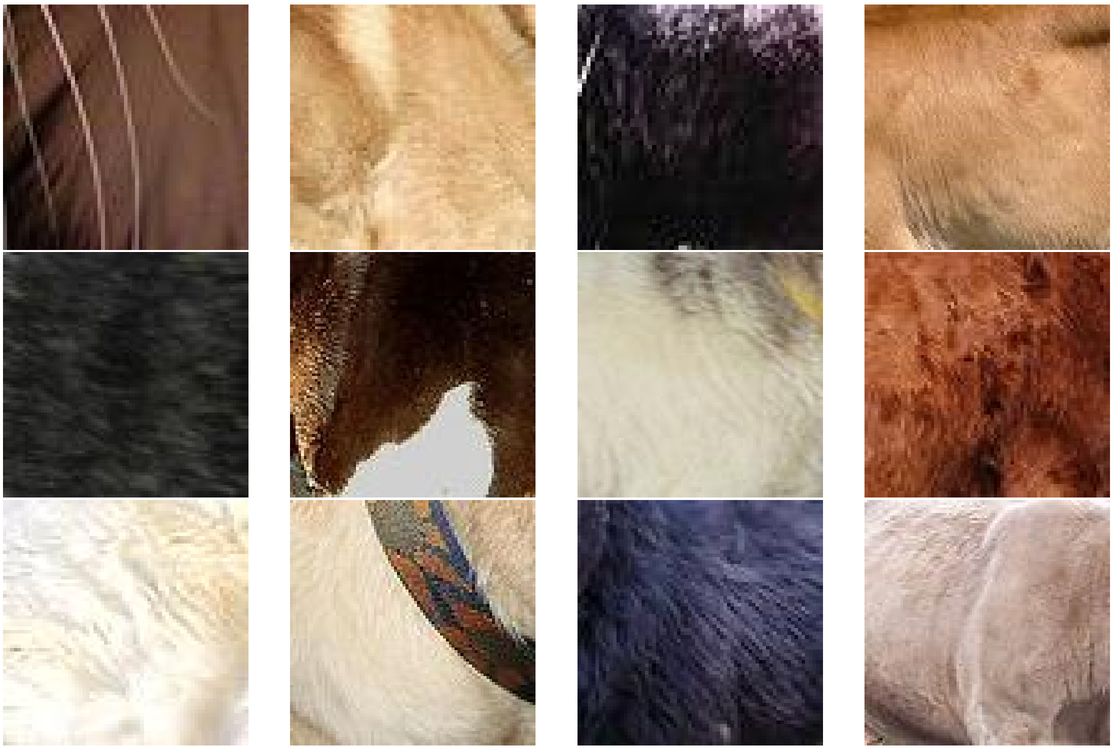
(a) Fabric (b) Foliage (c) Glass (d) Stone

Figure 2.16: Examples of four texture classes from the FMD database.

The Flickr Material database (FMD) was developed by Sharan et al. [166] with the intention of capturing a range of real world appearances of common materials. The dataset contains 1 000 images downloaded manually from Flickr.com (under Creative Commons license), belonging to one of the following materials: Fabric, Foliage, Glass, Leather, Metal, Paper, Plastic, Stone, Water and Wood. There are exactly 100 images for each of the 10 material classes. Unlike the dataset described above, FMD was not primarily created for texture recognition, and it includes images of objects with various textures for each material. The dataset includes binary masks for background segmentation. The standard evaluation protocol divides the images in each class into two halves, 50 images for training and 50 for testing. Examples from the FMD dataset are displayed in Figure 2.16.

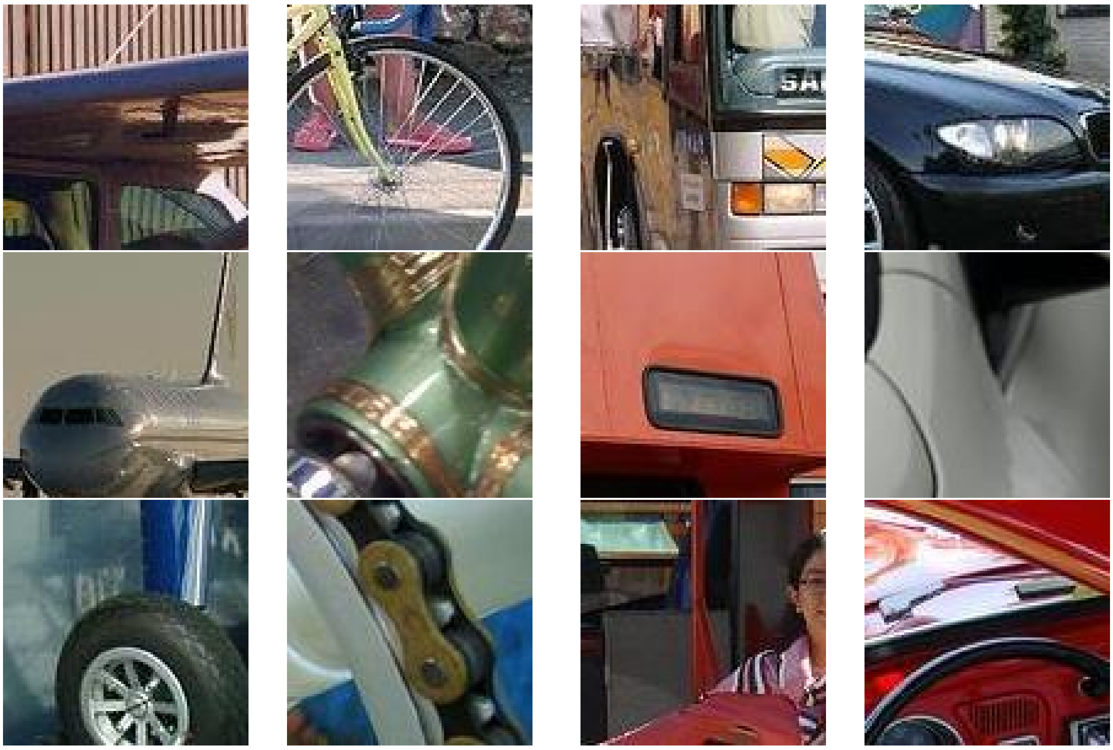
The Animal Texture dataset (AniTex) constructed by Mao et al. [128] contains 3120 texture patch images cropped randomly from the torso regions inside the silhouettes of different animals in the PASCAL VOC 2012 database. There are only 5 classes (cat, dog, sheep, cow and horse), 624 images each. The authors created the dataset to explore less homogeneous texture and appearance than available in standard texture datasets. The patches in the dataset come from images under different conditions such as scaling, rotation, viewing angle variations and lighting condition change. For evaluation, the dataset is randomly divided into 2496 training and 624 testing images. Examples from the AniTex dataset are displayed in Figure 2.17.

The Vehicle Appearance dataset (VehApp) [128] was created by the same authors and with the same intentions as AniTex. It contains 13 723 images cropped from PASCAL VOC images containing vehicles of 6 classes (aeroplane, bicycle, car, bus, motorbike, train). The images are evaluated in a way similar to AniTex: 80% images are randomly chosen into the training set, the remaining 20% is used for testing. Examples from the VehApp dataset are displayed in Figure 2.18.



(a) Cat (b) Dog (c) Sheep (d) Cow

Figure 2.17: Examples of four texture classes from the AniTex database.



(a) Plane (b) Bicycle (c) Bus (d) Car

Figure 2.18: Examples of four texture classes from the VehApp database.

2.4 Results

2.4.1 Texture Classification

The Fast Features Invariant to Rotation and Scale of Texture, proposed in Section 2.2, was first validated on the texture recognition datasets from Section 2.3.3.

The results in Tables 2.1 and 2.2 show more than 99% accuracy on the Brodatz32, UIUCTex, UMD, CUReT and KTH-TIPS datasets. This almost perfect precision basically retires most of the standard texture classification datasets. The accuracy on the KTH-TIPS2, FMD, AniTex and Vehapp datasets is lower as the tasks go beyond pure texture classification: The 87.9% and 76.6% accuracy on KTH-TIPS2a and KTI-TIPS2b respectively still present a state-of-the-art performance. The 50.2%, 45.7% and 54.4% scores on the FMD, AniTex and Vehapp datasets respectively are outperformed by other methods. The more recent CNN-based approach of Cimpoi et al. [34] further improves the classification scores on most texture recognition datasets.

Table 2.1: Recognition accuracy (%) of Ffirst and the state-of-the-art on the KTH-TIPS datasets.

	KTH-TIPS2a	*KTH-TIPS2b*	*KTH-TIPS*
Num. of classes	11	11	10
Ffirst$^{\forall+}$	**87.9**$_{\pm 6.1}$	76.6$_{\pm 4.3}$	99.5$_{\pm 0.5}$
FV-VGG-VD [34]	–	**81.8**$_{\pm 2.5}$	**99.8**$_{\pm 0.2}$
FV-VGG-M [34]	–	73.3$_{\pm 4.7}$	**99.8**$_{\pm 0.2}$
IFV$_{\text{SIFT}}$ [32]	82.5$_{\pm 5.2}$	69.3$_{\pm 1.0}$	99.7$_{\pm 0.1}$
IFV$_{\text{SIFT}}$ + DeCAF	84.4$_{\pm 1.8}$	76.0$_{\pm 2.9}$	**99.8**$_{\pm 0.2}$
IFV$_{\text{SIFT}}$ + DeCAF+ DTD$_{\text{RBF}}$	–	77.4$_{\pm 2.2}$	–
IFV$_{\text{SIFT}}$ + DeCAF + Subcat. Prob. [173]	–	79.3±2.7	–
Scattering [169]	–	–	99.4$_{\pm 0.4}$
LHS [168]	73.0$_{\pm 4.7}$	–	–
SR-EMD-M [114]	–	–	**99.8**
PLS [152]	–	–	98.4
Active Patches [128]	75.7	–	–

2.4.2 Tree Bark Classification

Results of our texture recognition approach to tree bark classification on the Austrian Federal Forest bark dataset are compared with the best published results in Table 2.3. Note that the method from [174] assumes the orientation is fixed, which seems to be a valid assumption in the case of this dataset. However, unlike Ffirst, it does not provide rotation invariance. Because the bark dataset is very small, we skip experiments with CNNs, which need a considerably higher amount of data for the standard training procedures.

Table 2.2: Recognition accuracy (%) of Ffirst on standard texture datasets, compared with the state-of-the-art methods.

	Brodatz	*UIUC*	*UMD*	*CUReT*	*ALOT*	*FMD*	*AniTex*	*VehApp*
Num. of classes	32	25	25	61	250	10	5	6
Ffirst$^{\forall+}$	$99.4_{\pm0.3}$	$99.4_{\pm0.4}$	$99.3_{\pm0.3}$	$99.7_{\pm0.1}$	$96.4_{\pm0.2}$	$50.2_{\pm1.9}$	$45.7_{\pm1.8}$	$54.4_{\pm0.7}$
FV-VGG-VD [33]	–	**$99.9_{\pm0.1}$**	**$99.9_{\pm0.1}$**	$99.0_{\pm0.2}$	**$98.5_{\pm0.1}$**	$79.8_{\pm1.8}$	–	–
FV-VGG-M [33]	–	$99.6_{\pm0.4}$	**$99.9_{\pm0.1}$**	$98.7_{\pm0.2}$	$97.8_{\pm0.2}$	$73.5_{\pm2.0}$	–	–
IFV$_{\text{SIFT}}$ [32]	–	$97.0_{\pm0.9}$	$99.2_{\pm0.4}$	$99.6_{\pm0.3}$	–	$58.2_{\pm1.7}$	–	–
IFV$_{\text{SIFT}}$+DeCAF [32]	–	$99.0_{\pm0.5}$	$99.5_{\pm0.3}$	**$99.8_{\pm0.2}$**	–	$65.5_{\pm1.3}$	–	–
Scattering [169]	–	$99.4_{\pm0.4}$	$99.7_{\pm0.3}$	–	–	–	–	–
LHS [168]	**$99.5_{\pm0.2}$**	–	–	–	–	–	–	–
SR-EMD-M [114]	–	–	**99.9**	99.5	–	–	–	–
PLS [152]	–	96.6	98.9	–	93.4	–	–	–
Active Patches [128]	–	–	–	–	–	–	**50.8**	**63.4**

Table 2.3: Bark classification accuracy (%) of Ffirst and the state-of-the-art methods. Evaluation schemes using 10 fold cross validation, or 15 and 30 training images per class.

	AFF 10 fold	*AFF* 15 train.	*AFF* 30 train.
Ffirst$^{\forall+}$	**$96.5_{\pm1.2}$**	**$84.9_{\pm2.5}$**	**$90.4_{\pm1.6}$**
MS-LBP-HF-KlSVM [174]	$92.2_{\pm2.7}$	$74.4_{\pm3.4}$	–
MS-LBP-KlSVM [174]	$96.5_{\pm2.7}$	**$85.5_{\pm2.7}$**	–
Fiel, Sablatnig [52, 53]	–	64.2	69.7

2.4.3 Leaf Classification

Application of the proposed Fast Features Invariant to Rotation and Scale of Texture to identification of leaves [177] lead to excellent results on standard leaf recognition datasets, proposing a novel approach to visual leaf identification: A leaf is represented by a pair of local feature histograms, one computed from the leaf interior, the other from the border, see Figure 2.3. This description utilizing Ffirst outperforms the state-of-the-art on all tested leaf datasets, see Table 2.4. The proposed method achieves excellent recognition rates above 99% on the Austrian Federal Forests dataset, the Flavia dataset, the Foliage dataset, the Swedish dataset and the Middle European Woods dataset.

Leaf Classification with Deep Convolutional Neural Networks is hard to apply to experiment with small leaf datasets. To get a comparison with our textural method, we performed our experiment on the Middle European Woods dataset, fine-tuning from an

Table 2.4: Classification accuracy (%) of Ffirst on available leaf datasets: Austrian Federal Forests, Flavia, Foliage, Swedish, Middle European Woods and Leafsnap.

	AFF	*Flavia* 10 × 40	*Flavia* $\frac{1}{2} \times \frac{1}{2}$	*Foliage*	*Swedish*	*MEW*	*Leafsnap*	*Leafsnap* top 5
Num. of classes	5	32	32	60	15	153	185	185
Ffirst$^{\forall+}_{\mathrm{a}}$ (1)	97.1$_{\pm1.5}$	99.4$_{\pm0.3}$	99.2$_{\pm0.2}$	99.2	99.7$_{\pm0.3}$	98.8$_{\pm0.2}$	81.2$_{\pm1.8}$	95.9$_{\pm1.5}$
Ffirst$^{\forall+}_{\mathrm{i}}$ (2)	97.3$_{\pm1.6}$	99.3$_{\pm0.3}$	98.9$_{\pm0.3}$	98.1	99.7$_{\pm0.3}$	98.4$_{\pm0.2}$	73.1$_{\pm2.3}$	92.4$_{\pm1.7}$
Ffirst$^{\forall+}_{\mathrm{b}}$ (3)	99.5$_{\pm0.6}$	99.3$_{\pm0.4}$	99.0$_{\pm0.2}$	98.3	99.4$_{\pm0.5}$	97.9$_{\pm0.2}$	77.2$_{\pm1.9}$	94.8$_{\pm1.5}$
Ffirst$^{\forall+}_{ib\sum}$ (4)	100.0$_{\pm0.0}$	99.7$_{\pm0.3}$	99.6$_{\pm0.1}$	99.3	99.8$_{\pm0.2}$	99.3$_{\pm0.1}$	81.8$_{\pm1.2}$	96.5$_{\pm1.1}$
Ffirst$^{\forall+}_{ib\prod}$ (5)	**100.0**$_{\pm0.0}$	**99.8**$_{\pm0.3}$	**99.7**$_{\pm0.1}$	**99.3**	**99.8**$_{\pm0.3}$	**99.5**$_{\pm0.1}$	**83.7**$_{\pm1.1}$	**97.3**$_{\pm1.1}$
Inc.-ResNet-v2 +maxout	–	–	–	–	–	**99.9**+	–	–
Kumar et al. [104]	–	–	–	–	–	–	≈ 73	96.8
Fiel, Sablatnig [53]	93.6	–	–	–	–	–	–	–
Novotný, Suk [132]	–	–	91.5	–	–	84.9	–	–
Karuna et al. [97]	–	–	96.5	–	–	–	–	–
Kadir et al. [96]	–	95.0	–	95.8	–	–	–	–
Lee et al. [112]	–	97.2	–	–	–	–	–	–
Qi et al. [150]	–	–	–	–	99.4	–	–	–

ImageNet-pretrained model. Note that due to high computational complexity and limited GPU resources, we only evaluated this method on one random data split (in both directions), while Ffirst was evaluated on 10 random splits. We used the Inception-ResNet-v2 network with maxout, described later in Section 4.2.1. After 200 000 training steps, this CNN outperforms previous results significantly, achieving **99.9% and 100.0% accuracy** respectively.

The excellent results presented above show that with a sufficient amount and quality of training data, the task of leaf recognition[7] is solved nearly perfectly by the proposed methods.

2.5 Significance of Colors in Texture Datasets

The results presented in Section 2.4.1 show that Deep Convolutional Neural Networks (CNNs) [33, 34] achieve state-of-the-art accuracy in texture classification, yet the hand crafted features still achieve competitive results and may be preferable in real-time applications for their performance without parallel processing. Although it has been shown that several texture description methods can benefit from adding color information [99], a large

---

[7] The task of leaf recognition constrained to leaf scans or photos of leaves on a white background.

number of the pre-CNN texture recognition techniques, including our Ffirst method proposed in Section 2.2, has been evaluated only on gray-scale images. Since many publicly available datasets used for texture recognition contain color information, we decided to evaluate the accuracy of color-statistics based methods to measure the significance of color information in the datasets.

We study the significance of color information in available datasets commonly used for evaluation of texture recognition methods. In total we evaluate 8 texture datasets, namely FMD (Flickr Material Database), ALOT (A Lot Of Textures), KTH-TIPS (Textures under varying Illumination, Pose and Scale), KTH-TIPS2a, KTH-TIPS2b, CUReT (Columbia-Utrecht Reflectance and Texture), VehApp (Vehicle Appearance) and AniTex (Animal Texture).

We improve the existing color descriptors, Discriminative Color Descriptors (DD) [101] and Color Names (CN) [196]. DD and CN are based on partitioning of the color space into clusters and assigning each color the probabilities of belonging to individual clusters. Our extension to the DD and the CN descriptors adds the standard deviation for each color cluster to the descriptor. This leads to an improvement in recognition rates on all 8 tested datasets, as shown in the experiments in Section 2.5.3. In the experiments, we combine our state-of-the-art Ffirst descriptor with the improved CN$^\sigma$ descriptor, leading to further increase in recognition accuracy.

Section 2.5.1 reviews the state of the art in texture and color recognition, respectively. The selected "color-only" image descriptors and our extension to them are introduced in Section 2.5.2. Section 2.5.3 describes the experiments and presents the results.

2.5.1 Color Statistics for Classification

Color information is processed by many state-of-the-art descriptors in Computer Vision, including the neurocodes of Deep CNNs or different extensions of SIFT incorporating color. Yet we are interested in simpler color statistics, not making use of spatial information.

Standard approaches to collect color information include color histograms (based on different color representations), color moments and moment invariants. Sande et al. [194] provide an extensive evaluation of such descriptors. The Color Names (CN) descriptor by Weijer et al. [196] is based on models learned from real-world data obtained from Google by searching for 11 color names in English. The Color Names have shown to be a successful color attribute for object detection [98] and recognition [100]. The model assigns each pixel the probability of belonging to one of the 11 color clusters. A similar approach is used by the Discriminative Color Descriptor (DD) of Khan et al. [101], where the color values are clustered together based on their discriminative power in a classification problem with the objective to minimize the drop of mutual information of the final representation.

Khan et al. [99] study the strategies of combining color and texture information. They carried out a comparison of "color-only" descriptors on the publicly available KTH-TIPS2a, KTH-TIPS2b, and FMD datasets, and on another small dataset denoted as Texture-10. Since the results of Color Names and Discriminative Color Descriptors outperformed other color descriptors in texture classification, we will describe the usage of CN and DD in more detail in Section 2.5.2 and use the models in our experiments in Section 2.5.3.

2.5.2 Selected Color Descriptors

Based on the findings of Khan et al. [99] and on our preliminary results, we consider the Color Names [196] and Discriminative Color Descriptors [101] the best match for our experiments for their superior classification accuracy.

While each of the approaches creates the color models based on a different criteria, the result is a soft assignment of clusters to each RGB value. In both cases the assignment is performed using a lookup table, which creates a mapping from RGB values to probabilities over C clusters c_i, i.e. $p\left(c_i \mid x\right)$. In this work we use the lookup tables provided by the authors of the methods, i.e. the 11-dimensional Color Names representation by [196] and the universal color 11-, 25- and 50-dimensional representations by [101].

The models assume uniform prior over the color names $p(c_i)$. The conditional probabilities for each cluster c_i given an image I are computed as an average over all N pixels x_n in the region:

$$p\left(c_i \mid I\right) = \frac{1}{N} \sum_{x_n \in I} p\left(c_i \mid x_n\right) \tag{2.7}$$

The standard descriptor D for image I is then a vector containing the probability of each cluster:

$$D(I) = \begin{bmatrix} p\left(c_1 \mid I\right) \\ p\left(c_2 \mid I\right) \\ \vdots \\ p\left(c_C \mid I\right) \end{bmatrix} \tag{2.8}$$

We propose to add another statistics to the color descriptor, the standard deviation of the color cluster probabilities in the image:

$$\sigma(c_i \mid I) = \sqrt{\frac{1}{N} \sum_{x_n \in I} \left[p\left(c_i \mid x_n\right) - p\left(c_i \mid I\right)\right]^2} \tag{2.9}$$

We concatenate the standard deviations to the original descriptor to get the extended representation:

$$D^{\sigma}(I) = \begin{bmatrix} p\left(c_1 \mid I\right) \\ p\left(c_2 \mid I\right) \\ \vdots \\ p\left(c_C \mid I\right) \\ \sigma(c_1|I) \\ \sigma(c_2|I) \\ \vdots \\ \sigma(c_C|I) \end{bmatrix} \tag{2.10}$$

2.5.3 Experiments with Color Descriptors on Texture Datasets

We compute 8 descriptors for each image in every database: the standard 11-dimensional Color Name descriptor CN and its proposed 22-dimensional extension CN$^{\sigma}$; the 11-, 25- and 50- Discriminative Color Descriptors DD11, DD25, DD50 and the extended versions DD11$^{\sigma}$, DD25$^{\sigma}$, DD50$^{\sigma}$ of double dimensionality.

The multi-class classification is then performed for each descriptor separately by combining binary SVM classifiers in a One-vs-All scheme. Linear SVM classifiers were used together with an approximate feature map of Vedaldi and Zisserman [199]. The χ^2 kernel approximations and the histogram intersection kernel approximations were considered, the latter was chosen based on slightly superior performance in preliminary experiments. The Platt's probabilistic output [117, 147] was used in order to estimate the posterior class probabilities to choose the result in the One-vs-All scenario. To minimize the effect of the random splits into training and testset, each experiment is performed 10 times on a different split, with the exception of the KTH-TIPS2 databases with 4 experiments based on the 4 material samples.

Table 2.5: Recognition accuracy (%) of selected color descriptors on publicly available databases commonly used for texture recognition.

	CUReT	TIPS	TIPS2a	TIPS2b	ALOT	FMD	AniTex	VehApp
# classes	61	10	11	11	250	10	5	6
CN	85.9±0.6	99.3±0.9	46.7±2.0	39.0±2.5	51.0±0.5	26.3±2.4	38.0±2.0	34.7±1.0
DD11	68.7±0.9	95.5±1.3	43.5±6.5	36.1±1.0	38.2±0.4	24.0±1.1	32.4±1.6	33.2±1.0
DD25	83.4±0.8	96.8±0.9	44.0±7.6	36.0±2.3	60.9±0.5	23.9±1.4	36.0±1.7	36.9±0.6
DD50	87.7±1.0	99.0±0.7	46.9±4.8	38.5±1.5	65.5±0.4	22.6±1.4	37.4±1.1	39.1±1.0
CN$^{\sigma}$	**94.2±0.6**	**99.8±0.3**	51.7±5.7	**42.6±1.4**	73.9±0.5	**28.0±2.2**	**41.7±1.8**	39.1±0.7
DD11$^{\sigma}$	81.9±0.8	97.6±1.0	48.5±3.8	38.3±1.9	60.1±0.5	22.7±1.6	35.9±2.1	35.8±0.5
DD25$^{\sigma}$	88.9±0.7	99.4±0.3	49.1±3.7	39.9±4.5	75.0±0.5	23.9±1.1	39.9±1.6	39.3±0.7
DD11$^{\sigma}$	91.0±0.7	99.6±0.2	**53.2±4.6**	42.0±2.8	**78.0±0.5**	25.3±1.7	38.9±0.8	**41.2±0.9**
FV-CNN [33]	99.0±0.2	–	–	81.8±2.5	98.5±0.1	79.8±1.8	–	–
Pure-texture	99.8±0.1 [169]	99.7±0.1 [32]	88.2±6.7 [176]	76.0±2.9 [176]	95.9±0.5 [176]	57.4±1.7 [151]	50.8 [128]	63.4 [128]

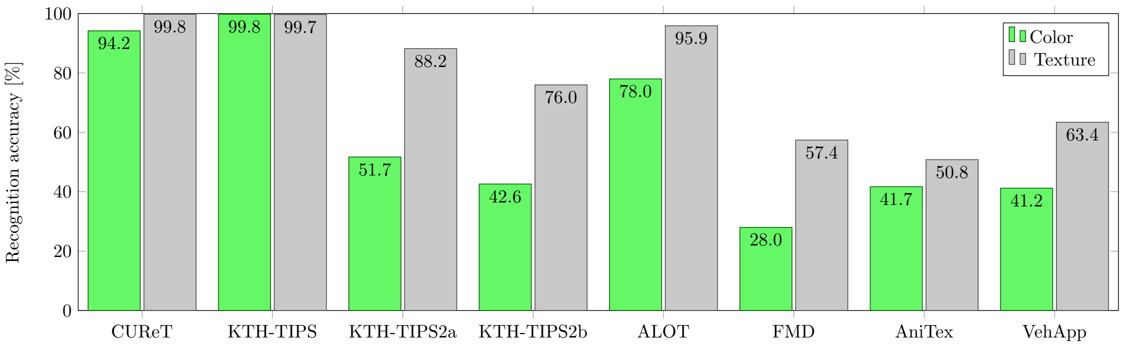


Figure 2.19: Comparison of the best published results of "texture-only" descriptors and the best results obtained using "color-only" descriptors.

All 8 color descriptors are compared in terms of class recognition accuracy in Table 2.5. The best published results of "texture-only" (color-less) methods and the results of the state-of-the-art FV-CNN [33] method are attached to the table for comparison. The comparison of the best "color-only" and "texture-only" results on all 8 datasets is illustrated in Figure 2.19.

An experiment on combining efficient classifiers of "texture-only" and "color-only" was performed as follows: Each image was described using the CN^σ color descriptor (using the same method as above) and the Ffirst texture descriptor (with $n_{\text{conc}} = 3$ descriptors per image, each describing $c = 7$ consecutive scales). An approximate intersection kernel map is applied to both color and texture descriptors, which are then classified using the One-vs-All Support Vector Machines with Platt's probabilistic outputs. The final scores in Table 2.6 were then combined using 3 axiomatic approaches, denoted as:

1. PROD: The product of both of the scores is used for final decision.
2. SUM: The sum of both of the scores is used for final decision.
3. $\text{SUM}_{0.3}$: The weighted sum of both of the scores is used for final decision, where the weight of color is only 30% of the weight of texture, taking into account the lower performance of the color descriptors on most datasets.

In terms of combining probability distributions [37], the SUM and $\text{SUM}_{0.3}$ schemes represent a *linear opinion pool* and the PROD scheme represents a *logarithmic opinion pool.*

Table 2.6: Recognition accuracy (%) of combinations of "texture-only" (Ffirst) and "color-only" (CN^σ) descriptors.

	CUReT	TIPS	TIPS2a	TIPS2b	ALOT	FMD	AniTex	VehApp
# cls	61	10	11	11	250	10	5	6
CN^σ	94.24±0.60	99.83±0.31	51.73±5.71	42.64±1.43	73.86±0.46	27.98±2.20	41.67±1.77	39.07±0.67
Ffirst	99.65±0.09	99.51±0.53	88.29±6.77	76.60±4.29	96.43±0.23	50.22±1.90	45.72±1.78	54.41±0.66
PROD	99.41±0.15	99.98±0.08	68.13±5.06	60.12±4.06	94.65±0.20	46.58±2.37	49.97±1.50	56.47±0.76
SUM	99.04±0.20	**100.00**±0.00	77.59±5.87	60.35±5.13	92.06±0.29	45.70±2.47	**50.08±1.56**	56.56±0.98
$\text{SUM}_{0.3}$	**99.68±0.12**	99.85±0.26	**88.76±6.40**	**77.17±4.23**	**97.05±0.14**	**52.24±1.68**	48.99±1.83	**56.62±0.92**

2.5.4 Discussion of the Results

Experimental results show that using only color descriptors is sufficient for almost perfect recognition accuracy of 99.8% on the KTH-TIPS dataset, where materials of the same color appear in both training and test data. In other words, KTH-TIPS is an extremely color-biased dataset. High accuracy scores of 94.2% and 78.0% were obtained using color descriptors on the CUReT and ALOT datasets respectively. The KTH-TIPS2a and KTH-TIPS2b datasets are more difficult for "color-only" classification, as testing data may come from material samples of different colors than training data, as illustrated in Figure 2.11. The FMD, AniTex and VehApp datasets are quite difficult for their heterogeneous nature, both in terms of texture and color. Yet the color statistics still provide useful information when combined with other descriptors.

An extension to the Color Names (CN) and Discriminative Color Descriptors (DD), denoted as CN^σ, DD^σ), significantly improved the recognition accuracy on all 8 tested datasets. In the experiments, Color Names outperform even the higher-dimensional Discriminative Color Descriptors DD25 on 6 out of the 8 experimented datasets, although the opposite may be expected from the findings on different tasks [101]. The improved CN^σ outperforms other "color-only" descriptors on 5 out of 8 datasets, the best results on the remaining 3 datasets are achieved by the improved $\mathrm{DD50}^\sigma$ descriptor.

Combining the "texture-only" *Ffirst* classifier with the "color-only" classifier of CN^σ leads to an improvement on all 8 tested datasets. Note that 100% accuracy was achieved on the KTH-TIPS dataset by combining the classifiers.

The state-of-the-art "texture-only" and "color-only" classifiers and their combinations obtain excellent results in the simpler texture-recognition tasks. The more recent deep learning models [33] perform better in the more difficult tasks. The simple "texture-only" and "color-only" descriptors may, however, still be favourable in applications, where low computational complexity is crucial.

CHAPTER 3

Deep Learning for Species Recognition in the Wild

This chapter deals with image-based recognition of plants and fungi “in the wild”, i.e. without constraints on acquisition conditions – such as lighting, scene background and clutter – and on the view type – entire plant or mushroom, a specific organ, etc. Relaxing constraints on input images increases the complexity of the identification: with complex background and possible clutter in the scene, including other specimen, the specimen of interest can not be segmented with a simplistic approach as in Section 2.2.5. Various views of a plant or a fungus, as opposed to a canonical view of a specified organ, require a recognition method that generalizes well to the varying appearance of the species observations and that is robust to clutter in the scene and to large differences in the acquisition conditions. Moreover, the datasets considered in this Chapter contain high numbers of species – up to 10 000 plant species in the LifeCLEF challenges [88,90,91] – further increasing the complexity of the recognition task.

We take a deep learning approach and use Deep Convolutional Neural Networks (CNNs), which have become the state-of-the-art approach to a number of computer vision tasks, often those related to complex fine-grained recognition [79, 102, 170, 186, 213] and detection [78, 118, 155, 157]. Successful learning of deep CNNs is typically conditioned by the existence of large-scale databases of annotated images. Such datasets were published with computer vision challenges – ImageNet Large Scale Visual Recognition Challenge(s) (ILSVRC) [45, 160], PASCAL Visual Object Classes (VOC) [50, 51] or Microsoft Common Objects in Context (COCO) [119]. Large scale datasets for fine-grained recognition of plants in the wild were published with computer vision challenges as well. The PlantCLEF [60,61,64] and ExpertLifeCLEF [63] datasets were published as challenges organized with the LifeCLEF workshops [88, 89, 90, 91]. The FGVCx Flowers 2018 [3] dataset was published as a challenge posted on Kaggle[1] and organized with the Fine-Grained Visual Categorization (FGVC) workshop at CVPR 2018. Similarly, for the fine-grained classification of fungi species, the FGVCx Fungi 2018 dataset was published as a challenge posted on Kaggle[2] and organized in conjunction the FGVC workshop at CVPR 2018. The datasets

---

[1] https://www.kaggle.com/c/fgvc2018-flower Last accessed 2nd Apr 2020.

[2] https://www.kaggle.com/c/fungi-challenge-fgvc-2018 Last accessed 2nd Apr 2020.

and challenges are described in detail later in Chapter 4.

Given the enormous popularity of deep learning in the last years and the volume of available deep learning literature (e.g. [67, 109, 164]), this chapter will assume the reader is familiar the with principles of deep learning such as back propagation, convolutional networks, commonly used activation functions and pooling layers, etc. Section 3.1 only briefly describes recent CNN architectures, of which several will be experimented in Chapters 4, 5 and 6.

In the species recognition datasets and competitions, there are often large differences between the class frequencies in the training and in the test data. In order to address this problem, Section 3.2 discusses the interpretation of CNN classifier outputs as posterior probabilities of classes given the image observations. Section 3.3 shows how to adjust the predictions by the new categorical priors, and deals with estimation of the new categorical priors at test-time from the CNN predictions.

3.1 CNN Classifier Architectures

The concept of Convolutional Neural Networks dates back to the 1980s with Fukushima's Neocognitron [56], followed i.a. by the fundamental works of LeCun et al. on handwritten digit recognition [110, 111]. The successful application of deep learning to large scale image recognition problems in the 21st century was – among other aspects – enabled by increasing hardware performance and by fast implementations on Graphics Processing Units (GPUs), allowing dramatic gains in computational performance thanks to utilizing the GPU parallelism [28, 35, 133].

The success of Krizhevsky's network [103] (later denoted as *AlexNet*) in the ImageNet 2012 Image Classification challenge increased the interest in deep learning for computer vision and started the deep learning era for large scale image recognition problems: The following ILSVRC classification challenges became a showcase of advances in convolutional neural networks architectures: *ZFNet* [208] in ILSVRC 2013, *GoogLeNet* (Inception v1) [187] and VGG networks [170] in 2014, Residual Neural Networks (ResNet) [79] in 2015. In the meantime, Sharif et al. [167] showed that the features extracted from ImageNet-pretrained CNNs provide strong baselines for a diverse range of recognition tasks like object image classification, scene recognition, fine grained recognition, attribute detection and image retrieval across a diverse set of datasets.

3.1.1 AlexNet

The network of Krizhevsky et al. [103] consisted of 5 convolutional layers, some of which were followed by max-pooling layers, and 3 fully-connected layers with a final 1000-way softmax classifier for the 1000 ImageNet classes.The network had 60 million parameters in total. On the ILSVRC 2012 validation set[3], the network achieved top-1 and top-5 error rates of 40.7% and 18.2% respectively. An ensemble of 7 networks achieved the best results in the competition with 15.3% top-5 error on the test set, while the best "pre-CNN" submission scored 26.2% top-5 error.

---

[3] Because the labels of the ILSVRC 2012 test set are not publicly available, publication results are commonly reported on the validation set. The results on the test and validation sets tend to correlate well.

3.1.2 VGG

Simonyan and Zisserman [170] introduced deeper convolutional networks, with up to 19 weight layers (16 convolutional layers and 3 fully connected layers) with a very small filters (3×3). Although stacking layers with small receptive fields decreases the number of parameters compared to layers with large receptive fields, the VGG networks have up to 144 million parameters. The deepest proposed network, commonly known as VGG-19, achieved the top-1 and top-5 error rates of 24.8% and 7.5% on the ILSVRC 2012 validation set (with a multi-crop settings of 150 crops per image).

3.1.3 GoogLeNet / Inception v1

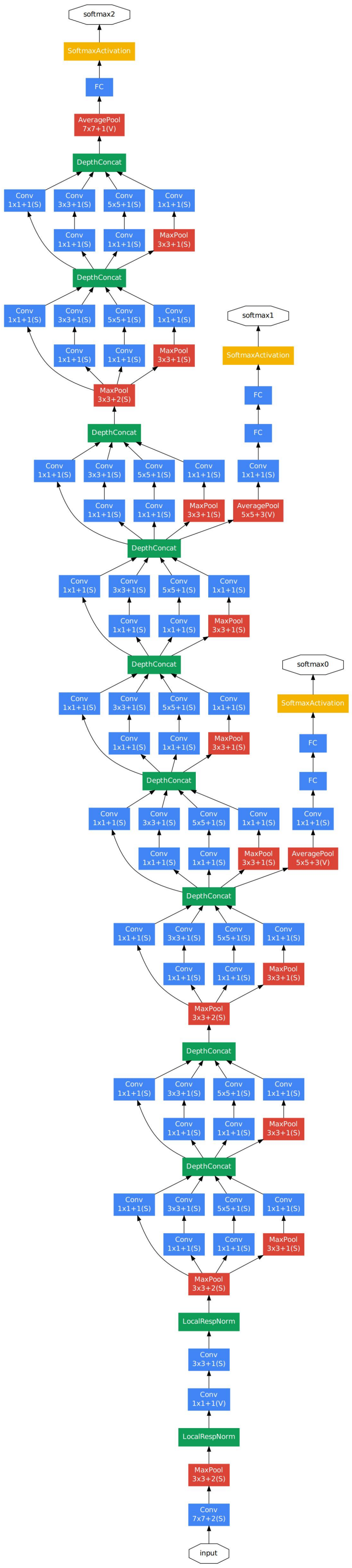
Szegedy et al. [187] focused on the utilization of computing resources and increased the depth and width of the network, while using less than 7 million parameters and about 1.5 billion operations. The GoogleNet architecture is based on stacking "Inception modules", in which the input is processed in parallel by several convolutional pathways with different filter types and sizes, with their output filter banks concatenated into a single output. Dimensionality reduction prior to expensive convolutions with larger patch sizes is achieved by adding a 1×1 convolution with a lower number of filters. Two auxiliary classification heads were added in the intermediate layers in order to combat the vanishing gradient problem while providing regularization. The architecture is visualized in Figure 3.1. Note that the Inception modules stacked in GoogleNet differ in the numbers of convolutional filters as well as in the input/output resolutions. With multi-crop setting (144 crops per test image], the 22 layers deep GoogLeNet achieves 7.89% top-5 error in the ILSVRC 2012 validation set.

3.1.4 Inception v2 and Inception v3

Several updates to the Inception architecture, increasing the accuracy and decreasing the computational complexity, were proposed in [188], introducing the Inception v2 and Inception v3 architectures. These updates included:

- avoiding representational bottlenecks (low representation dimensionality), especially early in the network
- factorization of convolutions with larger filter size into several smaller filters
- factorization of layers with medium grid-sizes into asymmetric convolutions ($n \times 1$)
- balancing the number of filters per stage (i.e. the "width" of the network) and the depth of the network
- batch normalization [84] of the layers in auxiliary classifiers (in v3)
- additional regularization by label smoothing (in v3)

Three new Inception modules with different combinations of filters were proposed following the above mentioned principles. Inception v3, the better performing architecture, achieved 21.2% top-1 and 5.6% top-5 error on ILSVRC 2012 val set with a single crop setting. With 144 test crops, it achieved top-1 and top-5 error of 18.77% and 4.2% respectively.



https://blog.csdn.net/qq_27464321

Figure 3.1: The GoogleNet architecture from [187].

3.1.5 ResNets

He et al. [79] proposed a residual learning framework, in which shortcut connections - skipping one or more layers - are used to make the layers learn residual functions w.r.t. the inputs, and tackle the vanishing gradient problem of deep networks at the same time. The residual blocks are illustrated in Figure 3.2. The residual learning allowed to train networks that are substantially deeper than those used previously. The authors evaluated ResNets of different depths (18, 34, 50, 101, 152 layers) on the ILSVRC 2012, where the deepest ResNet-152 (single model) achieved 19.38% top-1 and 4.49% top-5 error.

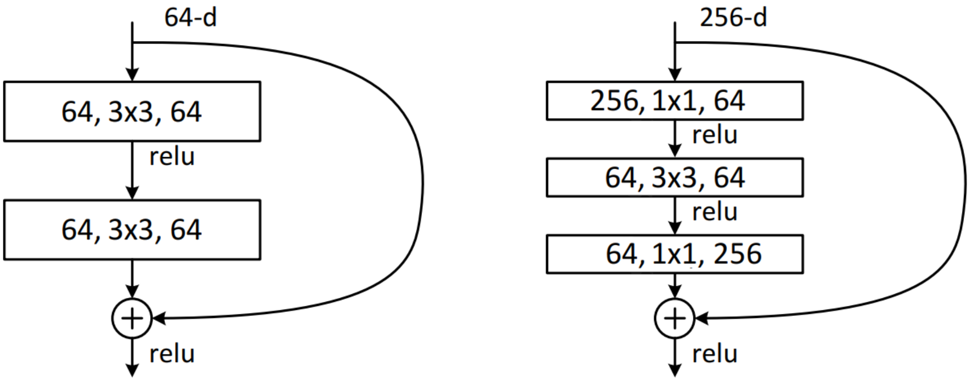


Figure 3.2: The residual blocks used for ImageNet Classification in ResNets [79] with similar time complexity. The "bottleneck" block on the right is used for speed up in the deeper ResNet-50/101/152 networks.

Xie et al. [206] proposed ResNeXt, where the residual blocks from Figure 3.2 are replaced with a block that aggregates a set of transformations with the same topology, as displayed in Figure 3.3.

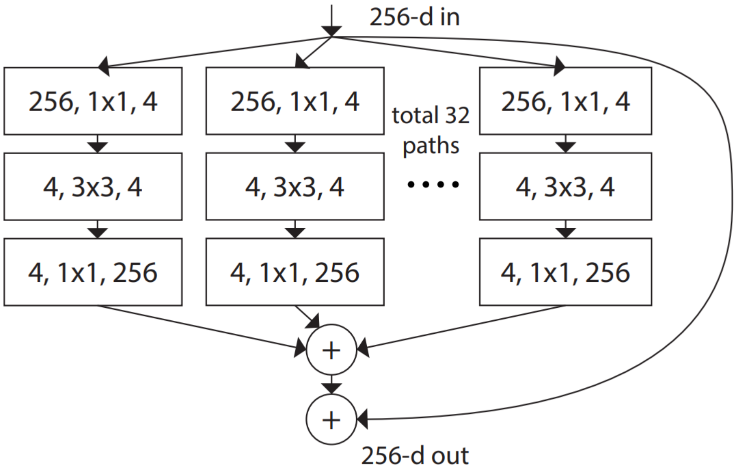


Figure 3.3: A block of ResNeXt [206] with cardinality 32.

3.1.6 Inception v4 and Inception-ResNets

In 2017, Szegedy et al. [186] proposed modifications to the Inception architectures discussed before. Inception-v4 includes a modification of the stem[4] of the network and an introduction of *reduction blocks* changing the width and height of the grid. The same paper introduced Inception-ResNets with residual connections, inspired by the work of He et al. [79]. Inception-ResNet v1 and v2 were designed to have similar computational costs to Inception v3 and Inception v4 respectively. With single-crop evaluations in the ILSVRC 2012 dataset, the Inception-v4 achieved 20.0% top-1 and 5.0% top-5 error and the Inception-ResNet-v2 achieved slightly lower errors of 19.9% and 4.9% respectively.

3.1.7 Efficient Architectures, Neural Architecture Search

Significant efforts have been made in the architecture of efficient models such as MobileNets [81] and their improvements [162], allowing fast inference and smaller model sizes. In parallel with the development of smaller and more efficient network architectures, decreasing the model size is possible via model compression with techniques such as network pruning [73], quantization of weights from full precision (32 bit floating point) into lower bit-depth representations [66, 72, 85, 204, 211] and Huffman coding [72]. Efficient architecture design together with such model compression techniques can decrease the model size enormously [83].

More recently, the CNN architecture engineering is partly being automated on datasets of interest - this includes works such as the Neural Architecture Search [213] proposing a set of NASNet architectures, a more efficient Progressive Neural Architecture Search [122] introducing PNASNets. The principles from [122, 213] have later been used to develop a family of efficient models called EfficientNets [190].

3.1.8 Maxout Networks

Several models proposed in Chapter 4 utilize a less common activation function - *maxout* [68]. Given an input $x \in \mathcal{R}^d$, a maxout hidden layer implements the following function:

$$\forall i \in \{1, \ldots, m\} : \qquad h_i(x) = \max_{j \in \{1,\ldots,s\}} z_{ij}, \tag{3.1}$$

where $z_{ij} = \mathbf{x}^{\mathrm{T}} \mathbf{W}_{\cdot ij} + b_{ij}$ is the j-th part of the hidden layer with learned parameters $\mathbf{W} \in \mathbb{R}^{d \times m \times s}$, $b \in {}^{m \times s}$ and m is the output size. In other words, maxout takes the maximum over s slices of the hidden layer.

One can understand maxout as a piece-wise linear approximation to a convex function, specified by the weights of the previous layer. This is illustrated in Figure 3.4.

Goodfellow et al. [68] designed *maxout* to leverage the *dropout* technique. When training with *dropout*, the element-wise multiplication with the dropout mask is applied immediately prior to the multiplication by the weights W.

---

[4] The term *stem* is used in [186] for the set of operations before the Inception modules.

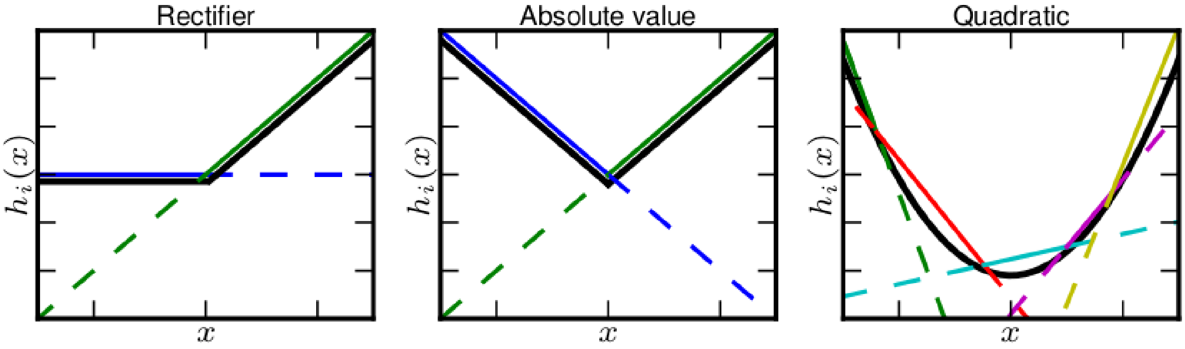


Figure 3.4: Illustration of maxout as a piece-wise approximation to a convex function, from Goodfellow et al. [68].

3.2 Probabilistic Interpretation of CNN Outputs

A CNN classifier model ended by a K-way softmax function can be expressed as a mapping from the image space to the *probability simplex*[5] Δ^{K-1}:

$$\mathbf{f}_{\text{CNN}} : \mathbb{R}^{W\times H\times 3} \mapsto \Delta^{K-1}, \tag{3.2}$$

in other words, the K outputs of the classifier are non-negative and sum to one. For a CNN classifier with parameters θ, let us use the notation $f_{\text{CNN}}(k|\mathbf{x};\theta)$ for the individual scalar outputs of the classifier, i.e.:

$$\mathbf{f}_{\text{CNN}}(\mathbf{x};\theta) = (f_{\text{CNN}}(1|\mathbf{x};\theta), \ldots, f_{\text{CNN}}(K|\mathbf{x};\theta)) \tag{3.3}$$

The common approach to training CNN classifiers is Stochastic Gradient Descent minimization of the *cross-entropy loss* L_{CE}:

$$L_{\text{CE}} = -\sum_{i=1}^{N}\sum_{k=1}^{K} c_{ik} \log f_{\text{CNN}}(k|\mathbf{x}_i;\theta)) \tag{3.4}$$

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

$$c_{ik} = \begin{cases} 1 \text{ if } k = y_i \\ 0 \text{ otherwise} \end{cases} \tag{3.5}$$

In this Section we show that minimizing the cross entropy loss L_{CE} is equivalent to the Maximum Likelihood Estimation (MLE) of parameters θ for an estimator of posterior probabilities $p_{\text{model}}(y|\mathbf{x})$.

Let X be a random variable representing images $\mathbf{x} \in \mathbb{R}^{W\times H\times 3}$ and Y be a random variable representing class labels $y \in \{1, \ldots, K\}$. Let us assume that the training set $\mathcal{T} = \{(\mathbf{x}_1, y_1), \ldots (\mathbf{x}_N, y_N)\}$ contains independent and identically distributed (i.i.d.) samples $(\mathbf{x}_i, y_i)$ from a distribution $p(\mathbf{x}, y)$. We can model the true distribution $p(\mathbf{x}, y)$ by a parametric model:

$$p_{\text{model}}(\mathbf{x}, y; \theta) = p_{\text{model}}(y|\mathbf{x};\theta) \cdot p_X(\mathbf{x}). \tag{3.6}$$

---

[5] The $(K-1)$-dimensional *probability simplex* Δ^{K-1} is the set of vectors $\left\{\mathbf{q} \in \mathbb{R}^K : \sum_{i=1}^{K} q_i = 1,\ \mathbf{q} \geq 0\right\}$.

The model parameters θ can be optimized using the framework of Maximum Likelihood Estimation (MLE). With the i.i.d. sampled training set $\mathcal{T}$ that means:

$$\theta_{\text{MLE}} = \arg\max_{\theta} \prod_{i=1}^{N} p_{\text{model}}(\mathbf{x}_i, y_i; \theta) = \arg\max_{\theta} \prod_{i=1}^{N} p_{\text{model}}(y_i|\mathbf{x}_i; \theta) \cdot p_X(\mathbf{x}_i). \tag{3.7}$$

Maximizing the likelihood is equivalent to maximizing the log-likelihood:

$$\begin{aligned} \theta_{\text{MLE}} &= \arg\max_{\theta} \left(\sum_{i=1}^{N} \log p_{\text{model}}(y_i|\mathbf{x}_i; \theta) + \sum_{i=1}^{N} \log p_X(\mathbf{x}_i) \right) \\ &= \arg\max_{\theta} \sum_{i=1}^{N} \log p_{\text{model}}(y_i|\mathbf{x}_i; \theta) \end{aligned} \tag{3.8}$$

We can see that training the CNN classifiers by minimizing the *cross-entropy loss* L_{CE} from Equation 3.4 is equivalent to the maximizing the log-likelihood in Equation 3.8:

$$\begin{aligned} \arg\min_{\theta} - \sum_{i=1}^{N} \sum_{k=1}^{K} c_{ik} \log f_{\text{CNN}}(k|\mathbf{x}_i; \theta) &= \arg\min_{\theta} - \sum_{i=1}^{N} \log f_{\text{CNN}}(y_i|\mathbf{x}_i; \theta) \\ &= \arg\max_{\theta} \sum_{i=1}^{N} \log f_{\text{CNN}}(y_i|\mathbf{x}_i; \theta) \end{aligned} \tag{3.9}$$

In other words, CNN classifiers are trained with cross-entropy loss in order to estimate the posterior probabilities:

$$f_{\text{CNN}}(y|\mathbf{x}; \theta) \approx p(y|\mathbf{x}) \tag{3.10}$$

3.2.1 Checking the Properties of Class Posterior Estimates

The true posterior probabilities $p(k|\mathbf{x})$ for an observation $\mathbf{x}$ are unknown, and thus we are not able to directly evaluate the accuracy of the posterior probability estimator $f_{\text{CNN}}(k|\mathbf{x}; \theta)$ from Equation 3.10. Let us check at least some properties that should hold if the trained classifier estimates the posterior probability $p(k|\mathbf{x})$ well. For now we fix the trained classifier parameters θ and use a simpler notation $f_{\text{CNN}}(k|\mathbf{x})$.

To experiment with the properties of the CNN predictions, we use the **CIFAR-100** [102] dataset, 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. Examples from the dataset are displayed in Figure 3.5. We sampled subsets of CIFAR-100 that follow pre-defined distributions from the exponential family. A 32-layer Residual Network [79] was trained on the training subsets.

First, we will check if averaging the predictions on training and test data estimates the class priors $p_Y(k)$ well:

$$\frac{1}{N} \sum_{i=1}^{N} f_{\text{CNN}}(k|\mathbf{x}_i) \approx \frac{1}{N} \sum_{i=1}^{N} p(k|\mathbf{x}_i) = p_Y(k). \tag{3.11}$$

On the labeled datasets, we will assume $p_Y(k) = \frac{N_k}{N}$, where $N_k = \sum_{i=1}^{N} c_{ik}$ is the number of images of class k and N is the number of all images in the dataset.

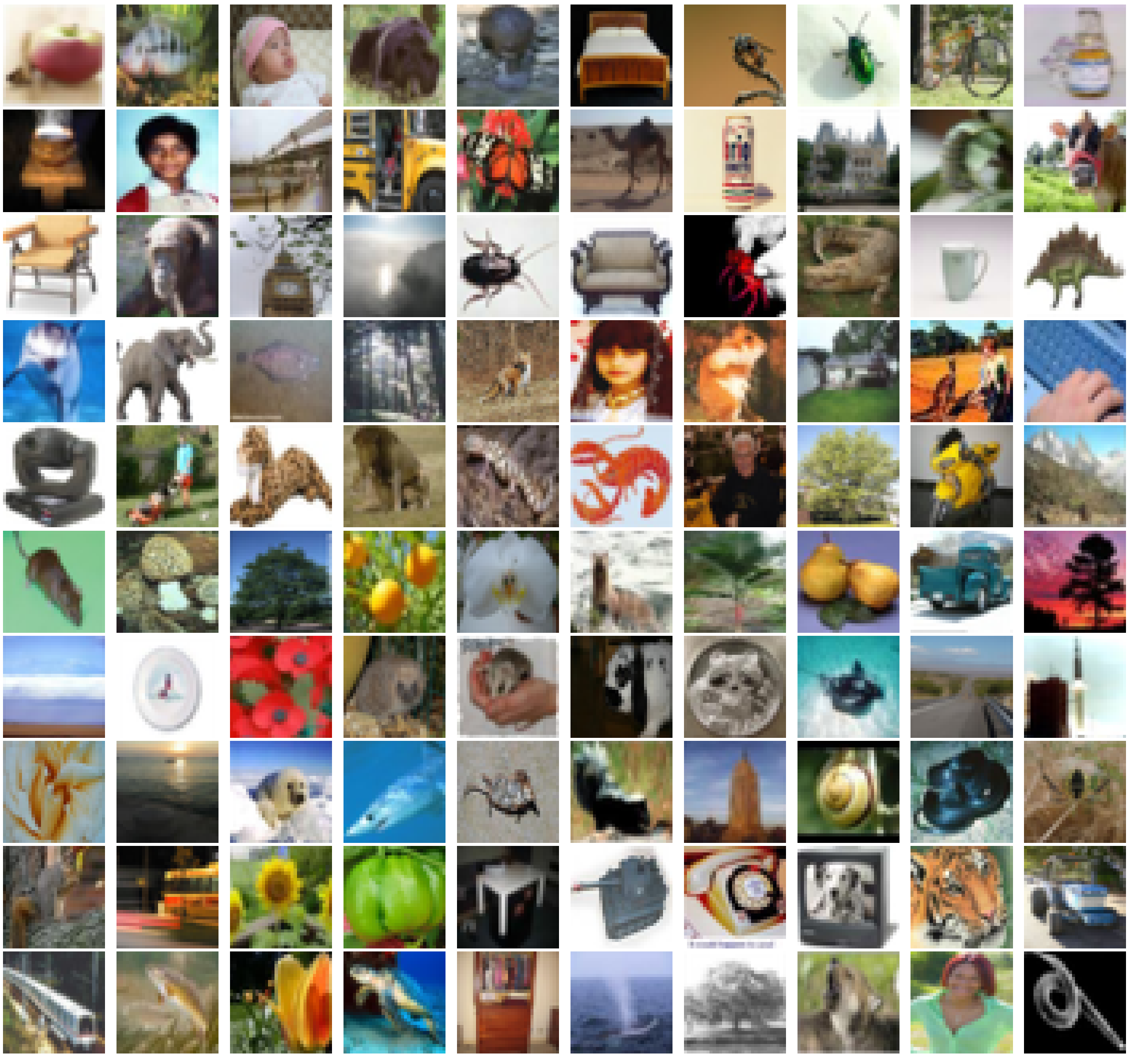


Figure 3.5: Examples from the CIFAR-100 dataset: one example per class.

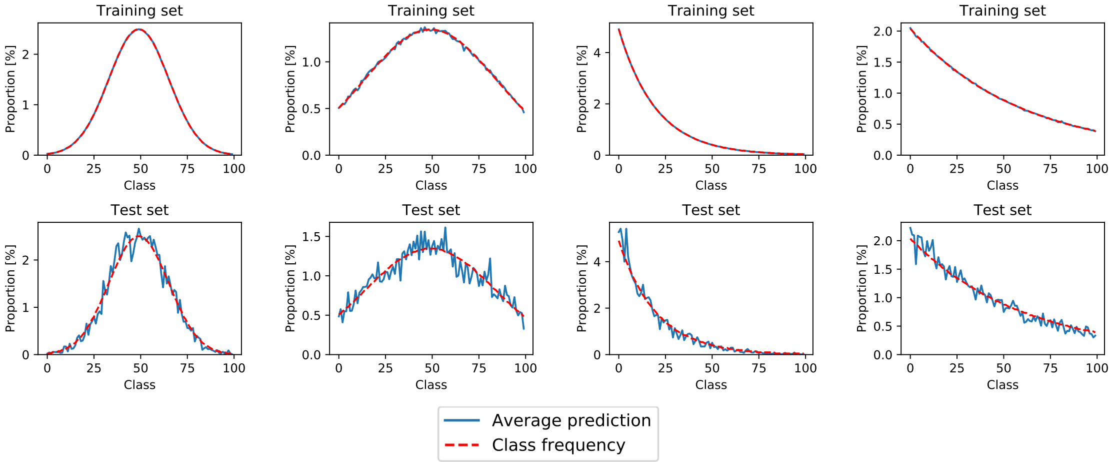


Figure 3.6: Comparison of class frequency and the average of CNN outputs over all images in the train- and test- sets sampled from CIFAR-100.

The comparison of empirical class frequencies and the estimates obtained by averaging the CNN predictions is displayed in Figure 3.6. The training set class distributions are estimated almost perfectly. The estimates on the test set are more noisy, but approximate the class frequencies well.

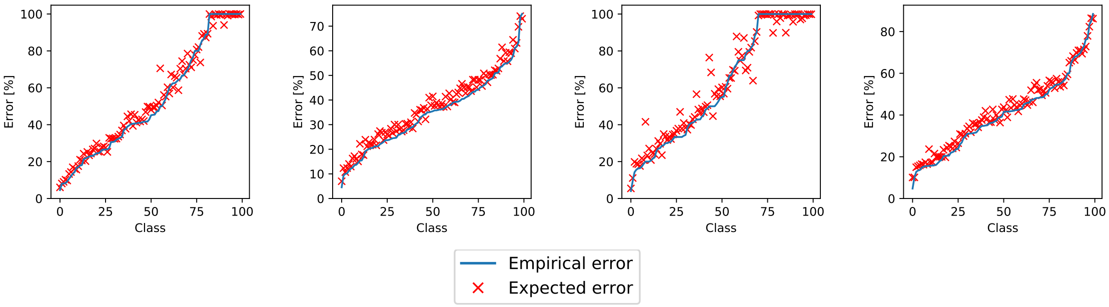


Figure 3.7: Test set empirical error ϵ_k^{emp} and the estimated error ϵ_k, classes sorted by ϵ_k^{emp}.

In the second validation experiment, we will look at the empirical error rate on images belonging to class k:

$$\epsilon_k^{\text{emp}} = \frac{1}{N_k} \sum_{i:y_i=k} [\![k \neq \arg\max_{k^*} f_{\text{CNN}}(k^*|\mathbf{x}_i)]\!]. \tag{3.12}$$

An unbiased estimator f_{CNN} should correctly classify an image $\mathbf{x}$ belonging to class k with probability $f_{\text{CNN}}(k|\mathbf{x})$ and incorrectly with probability $1 - f_{\text{CNN}}(k|\mathbf{x})$. Let us see if ϵ_k^{emp} can be approximated by averaging the probabilities of incorrect classification:

$$\epsilon_k = \frac{1}{N_k} \sum_{i:y_i=k} \left[1 - f_{\text{CNN}}(k|\mathbf{x}_i)\right], \tag{3.13}$$

The results of this comparison are shown in Figure 3.7: we can see that the estimated error ϵ_k and the empirical error ϵ_k^{emp} are aligned fairly well.

Note that both experiments were based on averaging the CNN outputs over a set of images, and are thus not sufficient to claim that that $f_{\text{CNN}}(k|\mathbf{x})$ provides a reliable estimate of the posterior probability $p(k|\mathbf{x})$. For example, both experiments could end up well even for a classifier that always predicts 100% confidence for the top 1 result while having non-zero classification error, if misclassifications happen in the right ratio.

3.2.2 Over-confident Classifiers

Due to over-fitting on the training set, "over-confident" predictions are a common problem of CNN classifiers: The CNN outputs for the top predictions tend to be higher than the expected correctness. Guo et al. [70] study the representations of such bias and compare a number of post-processing methods for calibrating CNN predictions.

To visualize the reliability of prediction confidence, i.e. the posterior estimate for the predicted class $\arg\max_k f_{\text{CNN}}(k|\mathbf{x})$[6], we use **Reliability Diagrams** [44, 70, 131], which plot the average sample accuracy as a function of confidence. Predictions are grouped by

---

[6] In the rare case of multiple occurrences of the maximum value $\max_k f_{\text{CNN}}(k|\mathbf{x}_i)$, we only assume the first occurrence returned by arg max.

confidence into M bins, such that samples with top-1 prediction confidence from interval $\left(\frac{m-1}{M}, \frac{m}{M}\right]$ fall into bin B_m.

The expected accuracy and average confidence of B_m are:

$$\begin{aligned} \operatorname{acc}(B_m) &= \frac{1}{|B_m|} \sum_{i \in B_m} [\![y_i = \arg\max_k f_{\text{CNN}}(k|\mathbf{x}_i)]\!] \\ \operatorname{conf}(B_m) &= \frac{1}{|B_m|} \sum_{i \in B_m} \max_k f_{\text{CNN}}(k|\mathbf{x}_i)) \end{aligned} \tag{3.14}$$

The reliability diagrams of the four CIFAR-100 classifiers from Section 3.2.1 displayed in Figure 3.8 show that the classifiers are indeed over-confident: the average accuracy of predictions from each bin B_m is much lower than the prediction confidence.

From the calibration methods compared in [70], Guo et al. conclude that **temperature scaling** is the most effective and simplest at the same time. It uses a single scalar parameter T called *temperature* to adjust the inputs into the softmax function

$$\sigma_k(\mathbf{z}) = \frac{\exp(z_k)}{\sum\limits_{j=1}^{K} \exp(z_j)} \tag{3.15}$$

applied to logits $\mathbf{z}$ (typically the outputs of a fully connected layer) as follows:

$$\hat{f}_k^{\text{TS}}(\mathbf{z}) = \frac{\exp(z_k/T)}{\sum\limits_{j=1}^{K} \exp(z_j/T)} \tag{3.16}$$

The parameter T is commonly optimized (on a fixed CNN) by minimization of the cross entropy loss - also denoted [70] as Negative Log Likelihood (NLL) when used with hard labels.

Note that while temperature scaling calibrates the classifier confidences, it is an order-preserving function, and thus does not affect the accuracy if the scaled predictions are used for classification directly. It may, however, affect the results if the predictions are used for further computation as posterior probabilities – e.g. in an empirical Bayesian strategy or when adjusting to new categorical priors, as proposed in Section 3.3. The effect of temperature scaling on the estimation of new categorical priors will be experimented in Section 6.5 t.

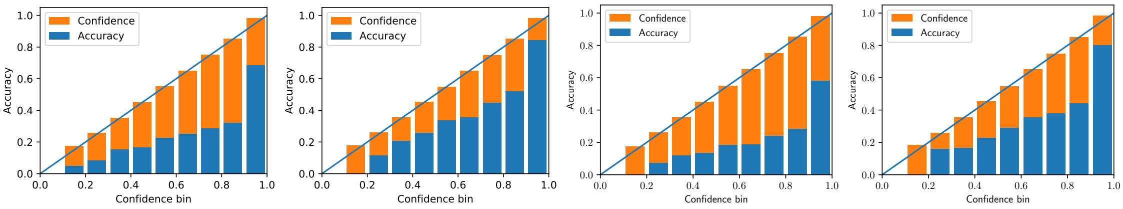


Figure 3.8: Reliability diagrams of the CIFAR-100 classifiers in the same order as in Figure 3.6.

3.3 Difference in Categorical Priors

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 [15, 67, 75]. 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 is typically not related to the frequency a species is queried in a plant identification service, where the frequency can also change in time and space.

Problems related to the differences between training- and test-set domains are studied in the field of domain adaptation [39, 140]. We are interested in the special case when statistical properties of observations from the same class stay the same (i.e. appearance does not change), and the only assumed difference is in the class priors $p_Y(k)$.

Methods [48,161] for adjusting classifier outputs to new and unknown a-priori probabilities have a long history, 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 [159], 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.

We focus mainly on the case when multiple observations are classified at once. Adopting the Maximum Likelihood Estimation (MLE) approach [48, 161], 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.

The rest of this Section formulates the compensation for the change in a-priori class probabilities in Section 3.3.1 and the estimation of the new a-priori probabilities using the frameworks of Maximum Likelihood in Section 3.3.2 and Maximum a Posteriori in Section 3.3.3.

Experimental evaluation, presented later in Chapter 6, shows 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. The importance of adaptation to new class prior probabilities is also shown by several contributions to classification challenges in Chapter 4.

3.3.1 New A Priori Class Distribution

Let us assume the probability density function $p^e(\mathbf{x}|k)$, describing the statistical properties of observations $\mathbf{x}$ of class k on the validation or test[7] set, remains unchanged from $p(\mathbf{x}|k)$ on the training set :

$$p(\mathbf{x}|k) = p^e(\mathbf{x}|k) \tag{3.17}$$

---

[7] We use index e to denote all evaluation-time distributions.

$$\frac{p(k|\mathbf{x}) \cdot p_X(\mathbf{x})}{p_Y(k)} = \frac{p^e(k|\mathbf{x}) \cdot p^e_X(\mathbf{x})}{p^e_Y(k)} \tag{3.18}$$

When the prior class probabilities $p^e_Y(k)$ in a validation or test set differ from the training set $p_Y(k)$, then the posterior $p^e(k|\mathbf{x})$ differs from $p(k|\mathbf{x}) \approx f_{\text{CNN}}(k|\mathbf{x})$. The new posterior probabilities can then be computed as

$$p^e(k|\mathbf{x}) = p(k|\mathbf{x}) \frac{p^e_Y(k) p_X(\mathbf{x})}{p_Y(k) p^e_X(\mathbf{x})} \tag{3.19}$$

Since $\sum\limits_{k=1}^{K} p^e(k|\mathbf{x}) = 1$, we can get rid of the unknown probabilities $p_X(\mathbf{x}), p^e_X(\mathbf{x})$ of fixed sample $\mathbf{x}$:

$$p^e(k|\mathbf{x}) \propto p(k|\mathbf{x}) \frac{p^e_Y(k)}{p_Y(k)} \tag{3.20}$$

The class priors $p_Y(k)$ can be empirically estimated as the fraction of images labeled with class k in the training set, $\frac{N_k}{N}$. The test-time priors $p^e_Y(k)$ are, however, often unknown at test time.

3.3.2 ML Estimate of New A Priori Probabilities

Saerens et al. [161] proposed to approach the estimation of unknown test-time a-priori probabilities by maximizing the likelihood of the set of test observations $\mathcal{E} = \{\mathbf{x}_1, \dots, \mathbf{x}_{N^e}\}$:

$$L(\mathcal{E}) = \prod_{\mathbf{x} \in \mathcal{E}} p^e_X(\mathbf{x}) = \prod_{\mathbf{x} \in \mathcal{E}} \sum_{k=1}^{K} p^e(\mathbf{x}, k) \quad = \prod_{\mathbf{x} \in \mathcal{E}} \sum_{k=1}^{K} p(\mathbf{x}|k) p^e_Y(k) \tag{3.21}$$

or equivalently maximizing the log-likelihood:

$$\ell(\mathcal{E}) = \log L(\mathcal{E}) = \sum_{\mathbf{x} \in \mathcal{E}} \log \sum_{k=1}^{K} p(\mathbf{x}|k) p^e_K(k). \tag{3.22}$$

To compute an estimate $\hat{\mathbf{p}} = (\hat{p}_1, \dots, \hat{p}_K) \in \Delta^{K-1}$ of the new prior probabilities $(p^e_Y(1), \dots, p^e_Y(K))$, Saerens et al. [161] derive a simple EM algorithm comprising of the following steps:

$$p^{(s)}(k|\mathbf{x}) = \frac{p(k|\mathbf{x}) \frac{\hat{p}^{(s)}_k}{p_Y(k)}}{\sum\limits_{j=1}^{K} p(j|\mathbf{x}) \frac{\hat{p}^{(s)}_j}{p_Y(j)}} \tag{3.23}$$

$$\hat{p}^{(s+1)}_k = \frac{1}{N^e} \sum_{\mathbf{x} \in \mathcal{E}} p^{(s)}(k|\mathbf{x}) \tag{3.24}$$

where Eq. 3.23 is the Expectation-step, Eq. 3.24 is the Maximization-step, and $\hat{p}^{(0)}_k$ may be initialized, for example, by the training set relative frequency $\frac{N_k}{N}$.

Du Plessis and Sugiyama [48] proved that this procedure is equivalent to fixed-point-iteration minimization of the KL divergence between the test observation density $p^e_X(\mathbf{x})$ and its model $q^e_X(\mathbf{x}) = \sum\limits_{k=1}^{K} \hat{p}_k p(\mathbf{x}|k)$ on the test set.

$$\begin{aligned}\mathrm{KL}(q_X^e \| p_X^e) &= \sum_{\mathbf{x}\in\mathcal{E}} p_X^e(\mathbf{x}) \log \frac{p_X^e(\mathbf{x})}{q_X^e(\mathbf{x})} \\ &= \sum_{\mathbf{x}\in\mathcal{E}} p_X^e(\mathbf{x}) \log p_X^e(\mathbf{x}) - \sum_{\mathbf{x}\in\mathcal{E}} p_X^e(\mathbf{x}) \log \sum_{k=1}^{K} \hat{p}_k p(\mathbf{x}|k)\end{aligned} \tag{3.25}$$

Note that estimating the priors $\hat{\mathbf{p}} = (\hat{p}_1, \ldots, \hat{p}_K)$ by minimization of the KL divergence on the test set $\mathcal{E}$ can be rewritten as maximization of the log-likelihood $\ell(\mathcal{E}) = \log L(\mathcal{E})$ of the observed data given the prior probability estimates $\hat{p}_k \approx p_Y^e(k)$:

$$\begin{aligned}&\arg\min_{\mathbf{p}} \mathrm{KL}(q_X^e \| p_X^e) = \arg\min_{\mathbf{p}} \left(\frac{1}{N^e} \sum_{\mathbf{x}\in\mathcal{E}} \log p_X^e(\mathbf{x}) - \frac{1}{N^e} \sum_{\mathbf{x}\in\mathcal{E}} \log \sum_{k=1}^{K} p_k p(\mathbf{x}|k) \right) \\ &= \arg\max_{\mathbf{p}} \frac{1}{N^e} \underbrace{\sum_{\mathbf{x}\in\mathcal{E}} \log \sum_{k=1}^{K} p_k p(\mathbf{x}|k)}_{\ell} = \hat{\mathbf{p}}^{\mathrm{MLE}} \\ &\text{s.t.} \sum_{k=1}^{K} p_k = 1;\ \forall k : p_k \geq 0\end{aligned} \tag{3.26}$$

The final optimization objective is then:

$$\begin{aligned}\hat{\mathbf{p}}^{\mathrm{MLE}} &= \arg\max_{\mathbf{p}} \sum_{\mathbf{x}\in\mathcal{E}} \log \sum_{k=1}^{K} p_k \frac{p(k|\mathbf{x}) p_X(\mathbf{x})}{p_Y(k)} = \arg\max_{\mathbf{p}} \sum_{\mathbf{x}\in\mathcal{E}} \log \sum_{k=1}^{K} p_k \underbrace{\frac{p(k|\mathbf{x})}{p_Y(k)}}_{a_{ik}} \\ &\text{s.t.} \sum_{k=1}^{K} p_k = 1;\ \forall k : p_k \geq 0\end{aligned} \tag{3.27}$$

As shown in [48], using the EM algorithm from Eq. 3.23, 3.24 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. 3.26 using the projected gradient descent algorithm [20]. At each step s, we update the variables as follows:

$$\hat{p}_k^{(s+1)} = \pi \left(\hat{p}_k^{(s)} + \lambda \frac{\partial \ell(\mathcal{E})}{\partial \hat{p}_k} \right), \tag{3.28}$$

where λ is the learning rate, π represents the projection onto the unit simplex, and the partial derivatives are:

$$\frac{\partial \ell(\mathcal{E})}{\partial \hat{p}_k} = \sum_{\mathbf{x}\in\mathcal{E}} \frac{a_{ik}}{\sum\limits_{j=1}^{K} \hat{p}_j a_{ij}} \tag{3.29}$$

To compute the Euclidean projection π onto the unit simplex, we use the efficient algorithm from [49, 202].

3.3.3 MAP Estimate of New A Priori Probabilities

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

$$\begin{aligned}
\hat{\mathbf{p}}^{\text{MAP}} &= \arg\max_{\mathbf{p}} p(\mathbf{p}|\mathcal{E}) \\
&= \arg\max_{\mathbf{p}} p(\mathbf{p}) \prod_{\mathbf{x}\in\mathcal{E}} p(\mathbf{x}|\mathbf{p}) \\
&= \arg\max_{\mathbf{p}} \left[\log p(\mathbf{p}) + \sum_{\mathbf{x}\in\mathcal{E}} \log p(\mathbf{x}|\mathbf{p}) \right] \\
\text{s.t. } & \sum_{k=1}^{K} p_k = 1;\ \forall k : p_k \geq 0
\end{aligned} \tag{3.30}$$

Note that the second term is the log-likelihood from the previous section, $\ell(\mathcal{E}) = \sum\limits_{\mathbf{x}\in\mathcal{E}} \log p(\mathbf{x}|\mathbf{p})$.

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

$$p(\mathbf{p}) = \frac{1}{B(\alpha)} \prod_{k=1}^{K} p_k^{\alpha-1} \tag{3.31}$$

parameterized by $\alpha > 0$, where the normalization factor for the symmetric case is

$$B(\alpha) = \frac{\Gamma(\alpha)^K}{\Gamma(\alpha K)}. \tag{3.32}$$

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 3.3.2 by adding the following gradient components:

$$\frac{\partial \log p(\hat{\mathbf{p}})}{\partial \hat{p}_k} = \frac{\partial (\alpha-1)\log(\hat{p}_k) - \log B(\alpha)}{\partial \hat{p}_k} = \frac{\partial (\alpha-1)\log(\hat{p}_k)}{\partial \hat{p}_k} = \frac{\alpha-1}{\hat{p}_k} \tag{3.33}$$

The adjustment of CNN outputs to new priors $p_Y^{\mathrm{e}}(k)$ proposed in Section 3.3, including the Maximum Likelihood and Maximum A Posteriori estimation of the new priors from Sections 3.3.2 and 3.3.3 respectively, will be used in some of our submissions to computer vision challenges described in Chapter 4. A more comprehensive evaluation of the methods is in Chapter 6 - a reader interested only in the adjustment of CNN priors may skip Chapters 4 and 5.

CHAPTER 4

In-the-wild Classification Competitions

As discussed in Chapter 3, many recent advances in image recognition have been enabled by the publication of large-scale datasets [45,50,51,119,160] for computer vision challenges and competitions. Image-based species recognition, a complex example of fine-grained classification, has been no exception: extensive image datasets have been published with computer vision challenges and competitions.

Sections 4.1, 4.2, 4.4, 4.5, 4.6 chronologically describe our submissions to the LifeCLEF plant identification challenges 2016-2019 and the FGVCx Fungi 2018, iNaturalist 2018 and FGVCx Flowers 2018 recognition challenges. Sections 4.3 and 4.6 evaluate the accuracy of the computer vision algorithms for plant classification in comparison to human experts.

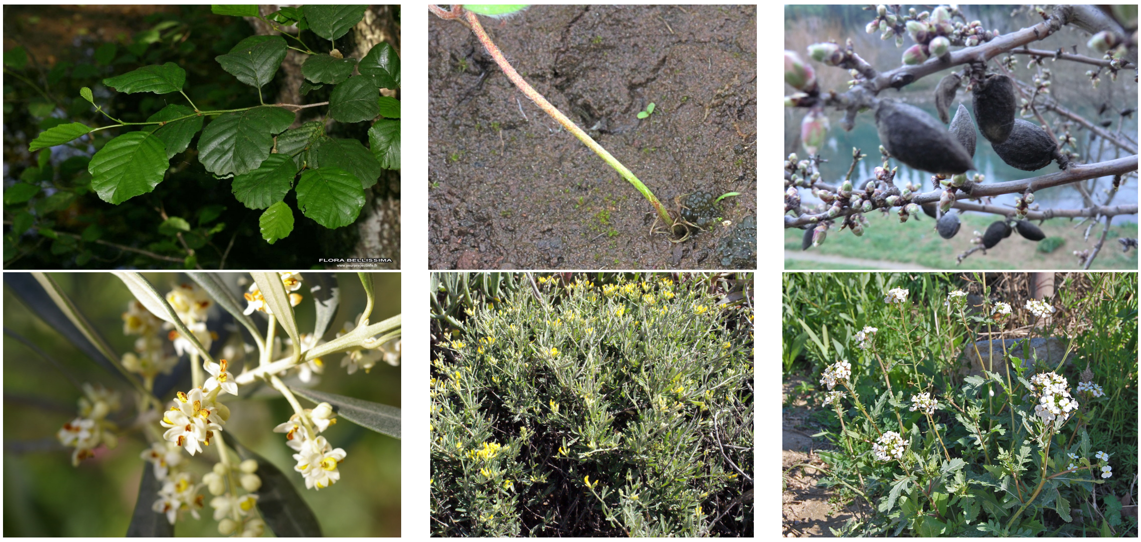
4.1 PlantCLEF 2016

4.1.1 The PlantCLEF 2016 Plant Identification Challenge

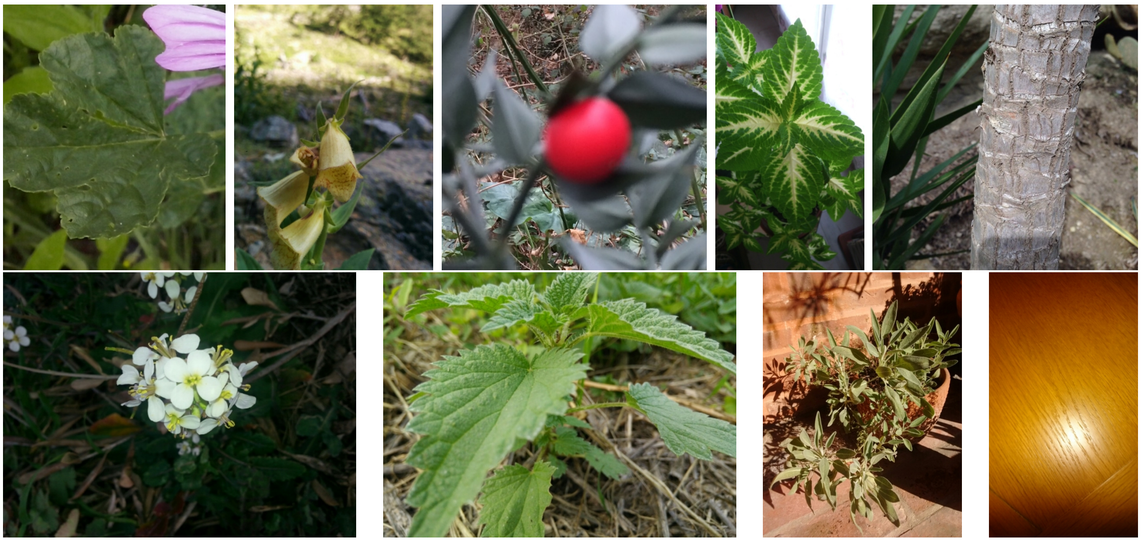
The task of the PlantCLEF 2016 [60,89] challenge is to automatically identify plant species from photos of different plant organs or the whole plant. The challenge deals with recognition of 1 000 plant species including herbs, trees and ferns.

The training data consist of 113 205 images of observations belonging to 1 000 species, and includes images from the training and test sets of PlantCLEF 2015. All images were annotated with the taxonomic species as well as other meta-data such as type of view (*Leaf*, *LeafScan*, *Flower*, *Fruit*, *Stem*, *Branch*, *Entire*), date of acquisition, author ID or GPS coordinates (if available). The test set contains 8 000 images. Examples from the training and test sets are displayed in Figure 4.1.

The 2016 challenge evaluation addressed the task as an open-set or open-world recognition problem [13, 163]: the test data contained distractors of unseen categories. The evaluation metric for the challenge is the mean average precision (mAP) over all species, where the average precision for each species is computed from the list of all test images sorted by the classifier output for the given species.



(a) Examples from the training set.



(b) Examples from the test set

Figure 4.1: Examples from the PlantCLEF 2016 challenge.

4.1.2 The Proposed Approach: Very Deep Residual Maxout Networks

The very deep residual networks of He et al. [79], briefly described in Section 3.1.5, gained a lot of attention after achieving the best results in both the ILSVRC 2015 and the COCO 2015 Detection Challenge. The residual learning framework allows to efficiently train networks that are substantially deeper than the previously used CNN architectures. Our networks are based on the ResNet models pre-trained on ImageNet, which are publicly available[1] for the Caffe deep learning framework [87].

To further improve the classification accuracy, we made a small change in the network architecture: an additional fully-connected layer with 512 neurons was added on top of the network, right before the softmax classifier. The activation function in the new layer is maxout [68], described in Section 3.1.8, with 4 linear pieces ($s = 4$). Dropout with a ratio

---

[1] https://github.com/KaimingHe/deep-residual-networks Last accessed 2nd Apr 2020.

of 50% is applied after the maxout layer and before the classifier.

The final layer is a standard 1000-way softmax classifier corresponding to the number of plant species in the challenge. Glorot [58] initialization was used for the two fully connected layers.

We have fine-tuned the networks for submissions *CMP Run 1*, *CMP Run 2* and *CMP Run 3* for 150 000, 150 000, and 370 000 iterations respectively, all with the following hyper-parameters:

- The learning rate was set to 10^{-3} and lowered by a factor of 10 each 100 000 iterations.
- The momentum was set to 0.9, weight decay to $2 \cdot 10^{-4}$.
- The effective batch size was set to 28 (either computed at once on NVIDIA Titan X, or split into more batches using Caffe's *iter_size* parameter when used on lower-memory GPUs).
- A horizontal mirroring of input images was performed during training.

Beyond fine-tuning the network, we performed bagging, inspired by its impact on the PlantCLEF 2015 results, where an interesting margin was gained with bagging of 5 networks by Sungbin Choi [31]. Due to computational limits at training time, we only used bagging of 3 networks, although we believe that using a higher number of more diverse networks would further improve the accuracy. The voting was done by taking species-wise maximum of output probabilities.

We have also experimented with training Support Vector Machines (SVMs) on L2 normalized outputs of the last pooling layer "kernelized" using an approximate χ^2 feature map [199]. Platt's probabilistic output was used [117, 147] to obtain comparable results with One-vs-All arrangement of the binary classifiers.

4.1.3 Preliminary Results and Validation

For our preliminary experiments and validation, we used the PlantCLEF 2016 training set, which is the union of previous year's (PlantCLEF 2015) training and test sets. Our validation process was threefold:

First, we validated networks trained on the PlantCLEF 2015 training set by evaluating them on the PlantCLEF 2015 test set using the previous year's metric [59]: the average classification rate per author of test observations. The goal of this phase was to validate that the ResNet-152 architecture is superior to the GoogleNet networks fine-tuned by the winners of the PlantCLEF 2015 challenge. While the best submission without bagging in the PlantCLEF 2015 challenge achieved a score of 59.4%, a fine-tuned ResNet-152 (without maxout) scored 62.1%. Applying SVMs on top of the last pooling layer pushed the score further to 62.5%, and training different SVMs for different groups of view types (always one for *Leaf* and *LeafScan*, second for *Flower* and *Fruit*, and third for *Stem*, *Branch* and *Entire*) gives another small improvement to 62.7%. The results are summarized in Table 4.1. Note that the meta information about view type was available in the test set too.

Second, we tested networks fine-tuned on PlantCLEF 2015 training data by evaluating them on the test set of PlantCLEF 2015 with the mAP metric used in the PlantCLEF 2016 challenge. This experiment evaluated the difference between the metrics used in

Table 4.1: Validation of the fine-tuned ResNet-152 using the PlantCLEF 2015 [59] score. *LR* denotes the learning rate, *it.* denotes the number of iterations, *sepSVM* is the set of SVM classifiers, each trained only on images of certain types (leaves, flowers & fruits, stem & branch & entire).

Method	*PlantCLEF'15 score*
LR = 0.001, 70K it.	58.7%
LR = 0.001, step size 100K, 150K it.	62.1%
LR = 0.001, step size 100K, 150K it. + SVM	62.5%
LR = 0.001, step size 100K, 150K it. + sepSVM	**62.7%**

Table 4.2: Validation of the fine-tuned ResNet-152 using the PlantCLEF 2016 [60] mAP score. *LR* denotes the learning rate, *it.* denotes the number of iterations, *sepSVM* is the set of SVM classifiers, each trained only on images of certain types (leaves, flowers & fruits, stem & branch & entire).

Method	*mAP*
ResNet-50 (150K it.)	50.3%
ResNet-152 (150K it.)	52.2%
ResNet-152 (150K it.) + SVM	51.8%
ResNet-152 (150K it.) + sepSVM	50.6%
ResNet-152 + maxout (130K it.)	56.8%
ResNet-152 + maxout (130K it.) + 10-view test aug. [103]	56.9%
ResNet-152 + maxout (130K it.) + fully convolutional eval.	55.9%
ResNet-152 + maxout (370K it.)	**57.3%**

PlantCLEF 2015 and PlantCLEF 2016. While deploying the SVMs slightly improved the 2015 score, using the CNN SoftMax output worked better for the 2016 metric. This is probably due to better comparability of SoftMax outputs among different samples, which was not important for the 2015 score, but which is crucial for the mAP used in PlantCLEF 2016. Fine-tuning ResNet-152 for 150 000 iterations lead to 52.2% mAP, while fine-tuning for 130 000 iterations with maxout lead to 56.8% mAP, proving that using maxout improves the accuracy significantly. The training set size is big enough to fine-tune networks for a larger number of iterations without overfitting, which brings additional 0.5% of mAP points, as shown in Table 4.2. The test-time 10-view image augmentation [103] (4 corner-crops, center crop and their mirrored versions) added only 0.2% of mAP, which is a smaller improvement than the ImageNet competition [103]. Evaluating the network in a fully convolutional style on images scaled to 448 pixels (in the longer dimension) surprisingly decreased the mAP by 0.9%.

Third and lastly, we performed bagging on the selected pipeline by dividing the 2016 training set into three folds and fine-tuning 3 networks, each using a different fold for validation and the remaining two folds for fine-tuning. The goal of this validation was to check that fine-tuning of all 3 networks converged to a meaningful solution.

4.1.4 Results on the Test Set

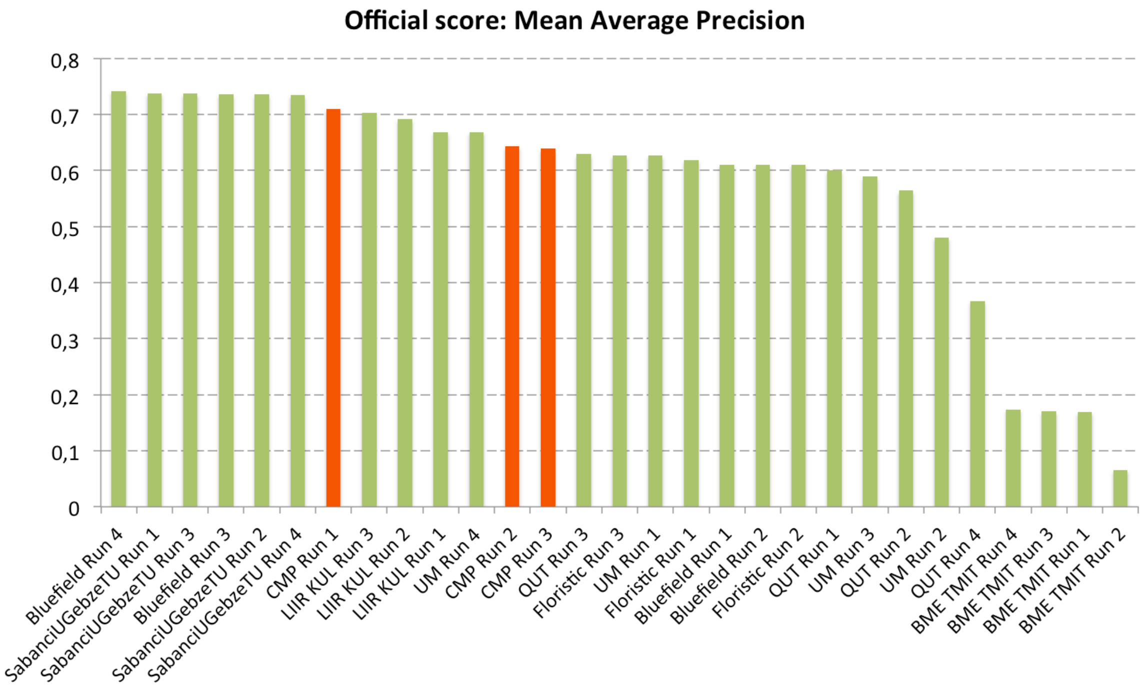


Figure 4.2: Results of the main task in PlantCLEF 2016 [60]. CMP results are in orange, our primary submission (CMP Run 1) scored among the 3 best performing teams.

Our primary submission, CMP Run 1 - the bagging of 3 deep residual (ResNet-152) networks with maxout fine-tuned each on two thirds of the PlantCLEF 2016 training set, scored 71.0% mAP and placed among the top 3 teams in the challenge and among the top 7 runs.

The second submission, CMP Run 2 - only one of the three residual networks from CMP Run 1, scored 64.4% mAP. The difference between the first two submissions is 6.6%, underlining the benefit of ensembling.

The third submission, CMP Run 3 - one network pretrained only on the LifeCLEF 2015 training set for a higher number of iterations, scored 63.9% mAP.

The official results of the PlantCLEF 2016 challenge are visualized in Figure 4.2. The winning submission of Bluefield (KDE TUT) [74] scored 74.2% mAP thanks to averaging the scores of images with the same *ObservationID* meta-information connecting images of the same specimen observation, i.e. transforming the task from single-image recognition to multiple-image recognition. While we did not realize it, combining image predictions across the test set was allowed by the challenge rules.

In a post-challenge experiment, we averaged the predictions in CMP Run 1 belonging to the same specimen observation, achieving 78.8% mAP, exceeding the winning score by 4.6%.

4.1.5 Discussion

Our results confirm the suitability of very deep convolutional networks for plant recognition in the wild, allowing to use a unified end-to-end pipeline for recognition of different plant organs and the whole plants in an uncontrolled environment. Significant improvements, compared to the most successful approaches from the 2015 challenge, were achieved by deploying a very deep (152-layer) residual network and placing a maxout layer in front of the classifier. The post-challenge experiment averaging the prediction scores per specimen observation points out the advantages of multiple-image recognition, increasing the mAP of our primary submission by 7.8%.

4.2 PlantCLEF 2017

The task of PlantCLEF2017 was again automatic identification of plants using computer vision. While a similar task has been the subject of previous challenges [59,89], PlantCLEF 2017 aims at a significantly larger scale: recognizing plants from 10 000 species with two sets of training data: a smaller "trusted" training set and a noticeably larger web-based "noisy" training set.

The challenge task and data are summarized in Section 4.2.1, the deep learning approach and all proposed modifications are described in Section 4.2.2. Preliminary experiments are presented and evaluated in Section 4.2.3. Post-processing steps are described in Section 4.2.3. The runfiles submitted to PlantCLEF are listed in 4.2.4. The results are discussed in Section 4.2.6.

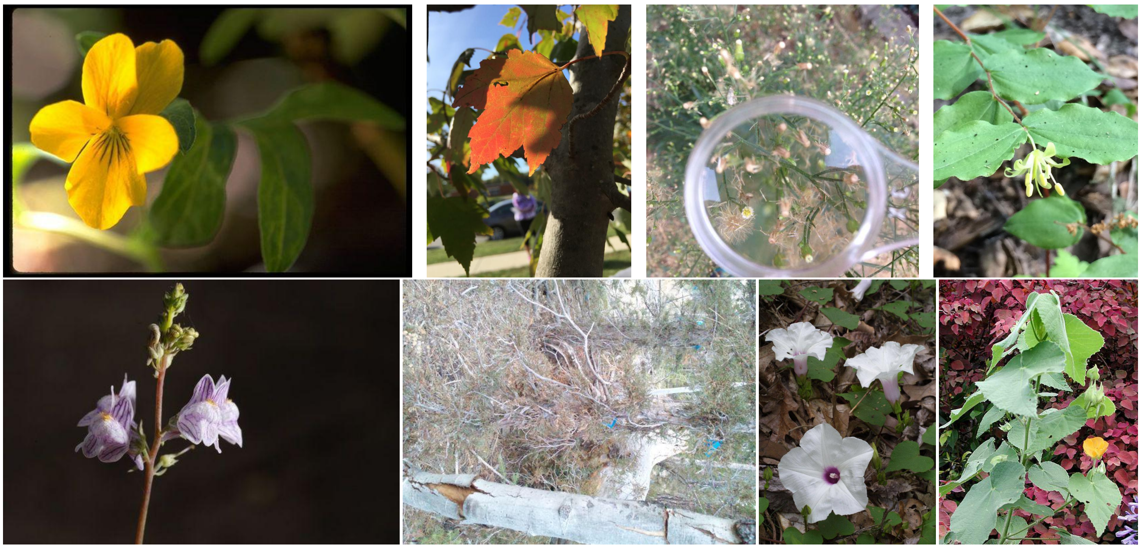
4.2.1 The PlantCLEF 2017 Plant Identification Challenge

Two sets of training data covering the same 10 000 plant species were provided by the organizers:

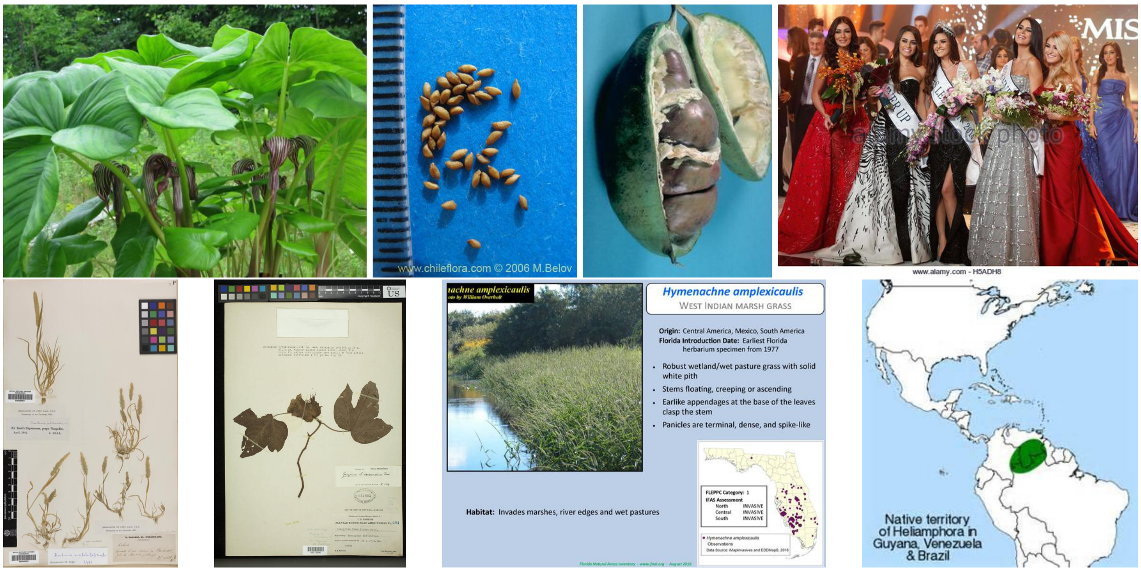
1. A set based on the online collaborative Encyclopedia Of Life (EoL) containing 256 287 images and corresponding xml files with meta-information. An important field in the meta-information is the "Observation ID", which is an identifier connecting images of the same specimen (object of observation). This dataset is considered "trusted", i.e. the ground truth labels should all be 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 1 442k image URLs. We obtained nearly 1 405k images from the list, the remaining images failed to download.

Examples from the "trusted" training set and the "noisy" (web) training set are displayed in Figure 4.3. The evaluation was performed on a test set containing 25 170 images of 13 471 observations (specimen).



(a) Examples from the "trusted" training set.


www.chileflora.com © 2006 M.Belov
www.alamy.com - H5ADH8
Hymenachne amplexicaulis
West Indian Marsh Grass
Origin: Central America, Mexico, South America
Florida Introduction Date: Earliest Florida herbarium specimen from 1977
Robust wetland/wet pasture grass with solid white pith
Stems floating, creeping or ascending
Earlike appendages at the base of the leaves clasp the stem
Panicles are terminal, dense, and spike-like
Habitat: Invades marshes, river edges and wet pastures
Native territory of Heliamphora in Guyana, Venezuela & Brazil

(b) Examples from the "noisy" training set.

Figure 4.3: Examples from the PlantCLEF 2017 challenge.

4.2.2 Methods

Inception-ResNet-v2

The submitted models were based on the Inception-ResNet-v2 [186] convolutional neural network architecture described in Section 3.1.6. Preliminary experiments showed that this network architecture lead to results superior to other state-of-the-art CNN architectures at the time. The network weights were initialized from a publicly available[2] Tensorflow model pre-trained on ImageNet. The main hyper-parameters used for training are summarized in Table 4.3.

---

[2] https://github.com/tensorflow/models/tree/master/research/slim/#pretrained Last accessed 2nd Apr 2020.

Table 4.3: Hyper-parameters used for training the Inception-Resnet-v2 model for PlantCLEF 2017.

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

MaxOut

We experimented with adding maxout to the end of the network, described in Section 3.1.8, which showed to be helpful in our submission to PlantCLEF 2016: We added an additional fully connected (FC) layer with 4096 units before the classification FC layer. The maxout activation operates over $s = 4$ 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 10 000-way softmax classifier corresponding to the number of plant species in the 2017 challenge.

We observed is that the additional FC layer has to be batch normalized [84]. Without batch normalization, the architecture became unstable, leading to an unexpected drop in accuracy.

Noisy Labels

In order to improve learning from noisy labels, Reed et. al. [156] proposed a simple consistency objective, which does not require an explicit information about the noise distribution. The new objective makes a linear combination of the noisy labels t_k with the network predictions, and takes the combination coefficient β as a hyper-parameter. There are two variants of the objective, denoted as *bootstrapping*:

- **soft bootstrapping** uses the softmax predictions q_k directly:

$$L_{\text{soft}}(\mathbf{q}, \mathbf{t}) = \sum_{k=1}^{K} [\beta t_k + (1-\beta) q_k] \log q_k \tag{4.1}$$

  Reed et al. [156] point out that the objective is equivalent to softmax regression with minimum entropy regularization, which was previously studied in [69]; encouraging high confidence in predicting labels.

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

$$L_{\text{hard}}(\mathbf{q}, \mathbf{t}) = \sum_{k=1}^{K} [\beta t_k + (1-\beta) z_k] \log q_k \tag{4.2}$$

The search for the optimal value of the hyper-parameter β was omitted for computational reasons and limited time for the competition. Instead, we set β according to the best results of Reed et al. [156]: $\beta = 0.8$ for hard bootstrapping and $\beta = 0.95$ for soft bootstrapping.

4.2.3 Experiments

We evaluated the proposed methods on a subset of the test data from the previous challenge, PlantCLEF 2016. Only 2583 images from the previous year dataset, for which we found corresponding species in the 2017 task, were used. This 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 the corresponding design choices:

Fine-tuning vs. Training from Scratch

The first question was whether the network should be trained from scratch, or fine-tuned from an ImageNet-pretrained checkpoint. We compared the two scenarios by training only on the "trusted" dataset. As illustrated in Figure 4.4, training from scratch (red) converges very slowly. After 150k trainning steps, fine-tuning (blue) leads to 65.1% accuracy, while training from scratch only gets to 44.5%, making fine-tuning the preferred approach. For illustration, 150k training iterations took approximately 65 hours on the NVIDIA Titan X GPU.

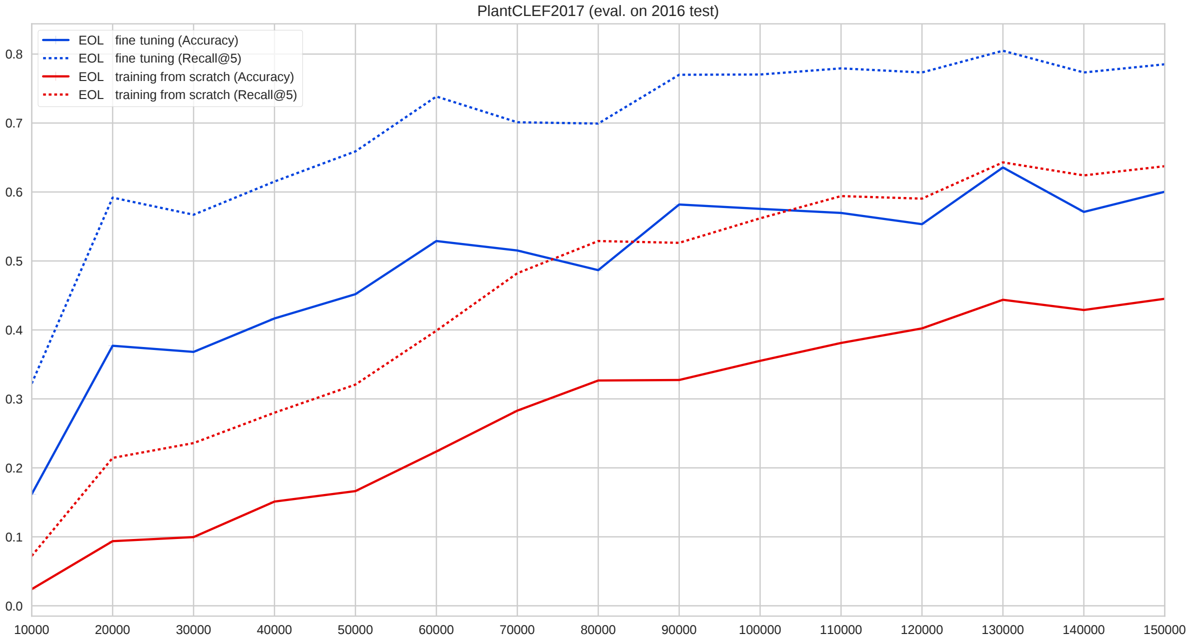


Figure 4.4: Accuracy (solid) and recal@5 (dotted) when fine-tuning (red) and training from scratch (blue).

Training on Trusted and Noisy Data. We fine-tuned five Inception-ResNet-v2 networks from the ImageNet-pretrained checkpoint: The first network was a vanilla Inception-ResNet-v2 fine-tuned only on the the "trusted" (EoL) data. The second network had the additional maxout layer and was also fine-tuned only on the the "trusted" (EoL) data. The other three networks were fine-tuned on all available data, including the "noisy" (web) samples, optimizing the standard cross entropy and adding soft- and hard-bootstrapping respectively.

The two networks trained only on the "trusted" data performed slightly better, and are therefore used in the follow-up experiments.

Filtering the Noisy Data and Further fine-tuning. In order to filter the "noisy" part of the training set, we used the network pre-trained on the "trusted set" to predict the labels from images. We assume that samples where the annotation and prediction agree are likely to have correct labels. One could consider accepting annotations corresponding to any of the top t predictions, where $t << K$, which should still discard out-of-domain (non plant) images and add harder examples to the training process. On the other hand, many similar species may have noisy samples belonging to the same genus or family, which share keywords used in the web search. The correct choice of t should then be based on an analysis of the way noisy labels are generated - e.g. the proportion of out-of-domain images and the proportion of mislabeled species belonging to the same higher taxonomic rank. Because of the limited time for the competition, we did not evaluate different choices of t and decided to continue with a low risk of false positive examples with $t = 1$: Only images, where the network top-1 prediction was equal to the annotation 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 4.2.3 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 divided 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.
- **Net #9,#10,#11:** Fine-tuned from #1 for 50k iterations on the trusted and filtered noisy data.
- **Net #12,#13,#14:** Fine-tuned from #1 for 50k iterations on the trusted and filtered noisy data, with hard bootstrapping.
- **Net #15,#16,#17:** Fine-tuned from #2 for 50k iterations on the trusted and filtered noisy data, with maxout.

Figure 4.5 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 4.2.3. The strength here is in combination of the differently fine-tuned networks. The red dashed line in 4.5 shows the final accuracy (after 50k it. of fine-tuning) of their combination.

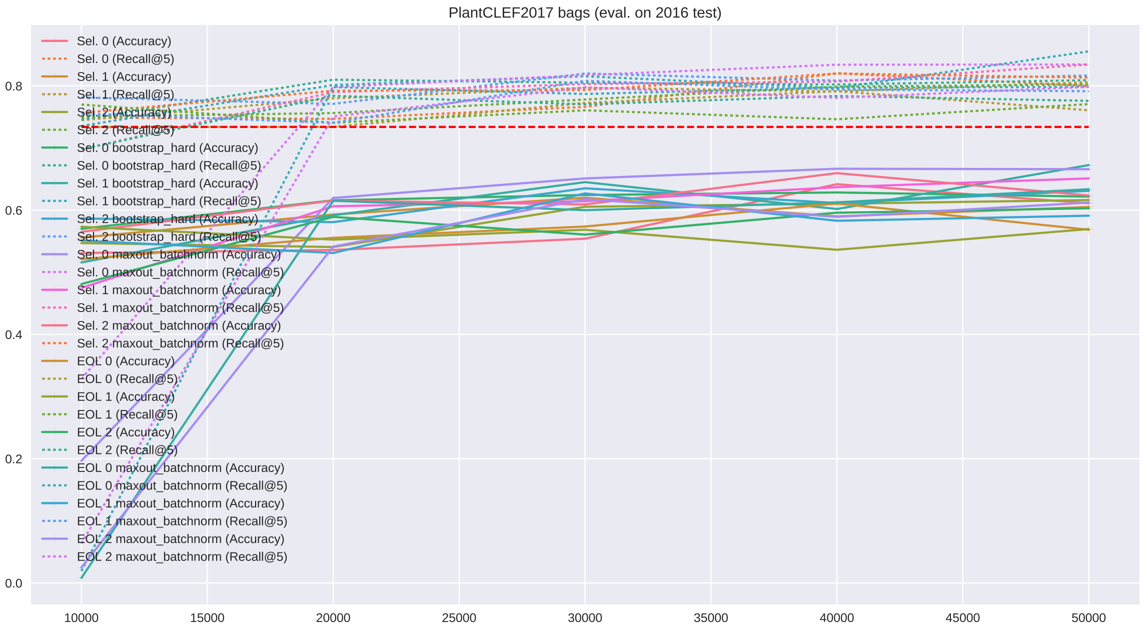


Figure 4.5: Accuracy (solid) and recal@5 (dotted) for further fine-tuning using different settings.

Post Processing on the Test Set

Averaging Predictions per Observation. As shown by the previous year's challenge winner [74], averaging the predictions over images of the same observation (specimen) increases accuracy significantly. Therefore we average scores per observations in all submitted runfiles.

Adjusting Predictions by Categorical Distribution. In PlantCLEF 2017, we first decided to experiment with adjusting the predictions, given the fact that we are evaluating the whole test set of images and assuming a change in categorical distribution: The training sets and the test set came from a a different source and therefore the species in the test set might not follow the same distribution as the species in the training set. We used the frequency of each class k among the observations in the "trusted" dataset as prior $p_Y(k)$, and estimated the test prior $p_Y^e(k)$ as the average predictions (per observation) on the test set. In order to make the adjustment of predictions softer, the ratio of priors was replaced by its square root.

The predictions $f_{\text{CNN}}(k|\mathbf{x})$ would be adjusted as follows:

$$q^*(k|\mathbf{x}) \propto f_{\text{CNN}}(k|\mathbf{x})\sqrt{\frac{p_Y^e(k)}{p_Y(k)}}, \tag{4.3}$$

Unfortunately, because of a mistake we adjusted the predictions for the challenge submissions wrongly:

$$q^\dagger(k|\mathbf{x}) \propto f_{\text{CNN}}(k|\mathbf{x})\sqrt{\frac{p_Y(k)}{p_Y^e(k)}}. \tag{4.4}$$

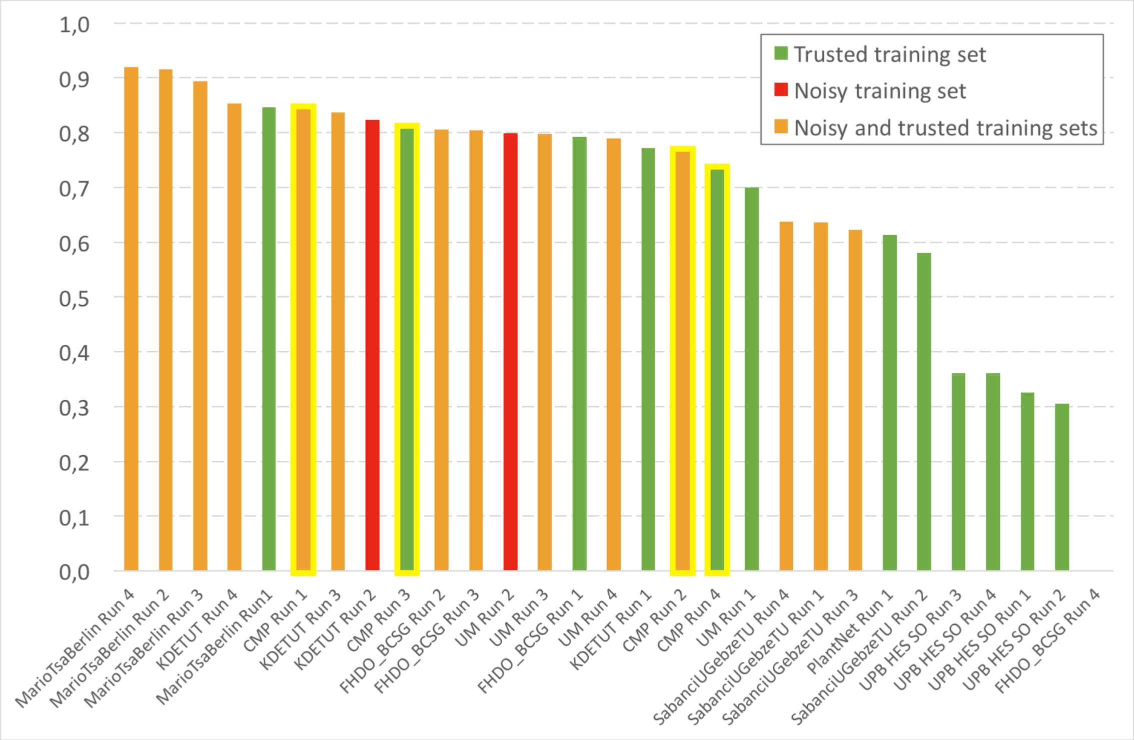


Figure 4.6: Results of the PlantCLEF 2017 [61] challenge.

4.2.4 Description of the Submitted Runfiles

In PlantCLEF 2017, each participant was 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 (wrong) prediction distribution adjustment from Section 4.2.3 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 (wrong) prediction distribution adjustment on top of results from the third runfile.

The challenge results are plotted in Figure 4.6: with CMP Run 1 we scored 3rd in the challenge (after MarioTsaBerlin and KDE TUT submissions).

4.2.5 Post Challenge Evaluation with Correct Prediction Adjustment

As discussed in Section 4.2.3, the prior adjustment in our submissions to the 2017 challenge was wrong. We noticed the error after the challenge has ended and our technical report [180] was published. Because the test set ground truth annotations and evaluation tools have been published after the challenge, we can compute the scores with the correct prior adjustment:

$$q(k|\mathbf{x}) \propto f_{\mathrm{CNN}}(k|\mathbf{x}) \frac{p^e_Y(k)}{p_Y(k)}, \tag{4.5}$$

and the “softer” version with square-root of the ratio of priors,

$$q^*(k|\mathbf{x}) \propto f_{\text{CNN}}(k|\mathbf{x})\sqrt{\frac{p_Y^e(k)}{p_Y(k)}}. \tag{4.6}$$

The results in Table 4.4 show that correcting the predictions with the correct ratio of priors, whether square-rooted or not, would noticeably improve the classification results.

Table 4.4: Post-challenge evaluation on the PlantCLEF 2017 test set with correct adjustment of predictions: Mean Reciprocal Rank (MRR), Top1 accuracy and Top5 accuracy.

		Accuracy (%)	
Run	MRR (%)	Top1	Top3
CMP Run 1: all data (trusted + filtered noisy)	84.3	78.6	91.3
CMR Run 2: all data, wrong adjustment with $\sqrt{}$	76.5	68.0	87.0
Post challenge: all data, adjustment with $\sqrt{}$ (Eq. 4.6)	86.6	81.7	92.8
Post challenge: all data, adjustment (Eq. 4.5)	**86.7**	81.5	93.0
CMP Run 3: only trusted data	80.7	74.1	88.7
CMR Run 4: trusted data, wrong adjustment with $\sqrt{}$	73.3	64.1	84.9
Post challenge: trusted data, adjustment with $\sqrt{}$ (Eq. 4.6)	83.2	77.5	90.3
Post challenge: trusted data, adjustment (Eq. 4.5)	**83.3**	77.4	90.6

4.2.6 Discussion

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 4.6. Based on our evaluation, the following steps increase the classification accuracy:

- Maxout [68] with batch normalisation [84] 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.

As shown in Section 4.2.5, adjusting the species distribution on the test set with the correct ratio of priors would noticeably increase the recognition scores.

4.3 Experts vs. Machines in Plant Identification 2017

The relatively high accuracy of computer vision / machine learning based methods for fine-grained plant species recognition in the PlantCLEF 2017 challenge raised questions about comparison of automated plant species recognition with human experts. While an experiment comparing "Man vs. Machine" in plant identification [18] was made already in 2014, the performance of machine-learned systems has increased significantly with the application of deep learning and convolutional neural networks. In order to compare the accuracy of human experts in the field of plant identification with the accuracy of machine learning systems for plant recognition, we contributed to the Experts vs. Machines experiments in 2017 [17].

4.3.1 Training Data and Method

The Experts vs. Machines (2017) experiments considered the same 10 000 plant species and were provided with the same training data as the PlantCLEF 2017 challenge described in Section 4.2.1, consisting from the "trusted" training set downloaded from EoL and the "noisy" training set obtained using web search engines. The main reason for providing both datasets was to evaluate to what extent can the training of computer vision models benefit from noisy data compared to training from trusted data only (as usually done in supervised classification). Keeping the same training datasets had another advantage: participants of PlantCLEF 2017 could easily contribute to this comparison with the models trained for the challenge. Therefore, we contributed with the results of our CNN ensemble trained for the PlantCLEF 2017 challenge, i.e. with the same method and models as described in Section 4.2.2. More specifically, we used the ensemble of 17 networks (averaging their results) denoted as CMP Run 1 in PlantCLEF 2017 in Section 4.2.4.

4.3.2 Test Data and Evaluation Protocols

Two experiments with different sets of test images and evaluation protocols have been performed. Both experiments use the Mean Reciprocal Rank (MRR) score, i.e. the mean of the inverse of the rank of the correct species in the predictions:

$$\mathrm{MRR} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{\mathrm{rank}_i}, \tag{4.7}$$

where $|Q|$ is the number of queries in the test set, and rank_i is the rank of the correct species for the i-th query.

Extending the 2014 Experiment

The first part extends the results of the Man vs. Machine experiments conducted in 2014 [18]. In order to allow a direct comparison, the experiments were performed on the same test images as in the [18], which in 2014 were selected at random from the whole set of PlantCLEF 2014 observations and shared with a large audience of potential volunteers composed of four target groups:

1. **“expert of the flora”**: highly skilled people such as taxonomists, expert botanists of the considered flora.
2. **“expert”**: skilled people like botanists, naturalists, teachers, but not necessarily specialized on the considered flora.
3. **“amateur”**: people interested by plants in parallel with their professional activity, having knowledge at different expertise levels.
4. **“novice”** : inexperienced users.

The human predictions were collected through a user interface presenting the 100 observations one by one (with one or more pictures of different plant organs) and allowing the user to select up to three species for each observation using a drop-down menu covering the 500 species of the PlantCLEF 2014 dataset. In order to facilitate the participation of amateurs and novices, the most popular common names were displayed next to the scientific name of the taxon. If the user did not provide any species proposition for an observation, the rank of the correct species was considered infinite in the evaluation metric. The evaluation has been restricted to the knowledge-based identification of plants, without any additional sources of information or identification tools allowed during the test. Concretely, the **participants were not allowed to use external resources** like field guides or flora books. Only 20 volunteers finally accepted to participate: 1 “expert of the French flora”, 7 “experts”, 7 “amateurs” and 5 “novices”. For a fair comparison with human-powered identifications, the only the top 3 predictions of machine learning models were taken into account. While [18] included the results of ten participants of LifeCLEF 2014 (with 27 runs in total), only three participants of LifeCLEF 2017 accepted to join the experiments with their competition models, and only two of us (KDE TUT, CMP = our submission) were actually eligible for the experiment, since the third participant (Mario TSA/MNB) trained on a dataset containing the tested 100 observations. Note that while the 2014 test set was limited to the 500 species used in this experiment, the additional 2017 predictions from CMP and KDE TUT were provided by networks trained for the recognition of the 10 000 species in PlantCLEF 2017 (including the 500 tested species).

Experts vs. Machines 2017 Experiment

With the aim to evaluate more precisely the capacities of state-of-the-art plant identification systems compared to human experts, the 2017 experiment was set up with:

1. a more difficult test set,
2. a group of highly skilled experts composed of the most renowned botanists of the considered flora.

The new test set was created by following procedure: First, 125 plants were photographed between May and June 2017, a suitable period for the observation of flowers in Europe, in a botanical garden called the ”Parc floral de Paris”, and in a natural area located in the north of Montpellier (southern part of France, close to the Mediterranean sea). The photos have been acquired using smartphone cameras, namely an iPhone 5 and a Samsung S5 G930F. The selection of species in the test set followed several criteria, including:

1. difficulty, i.e. commonly confused species,
2. the availability of well developed specimens with well visible organs, and
3. the diversity of the selected set of species in terms of taxonomy and morphology.

About 15 pictures of each specimen were acquired in order to cover all the informative parts of the plant. However, all pictures were not included in the final test set in order to deliberately hide a part of the information and increase the difficulty of the identification. Therefore, a random selection of only 1 to 5 pictures was made for each specimen. In the end, a subset of 75 plants illustrated by a total of 216 images related to 33 families and 58 genera was selected. This test set is available online[3] under an open data license (CC0) in order to foster further evaluations by other research teams.

The test set was sent to 20 expert botanists - taxonomists, botanists, research scientists specialising on the considered flora, and a few non-professional expert botanists. Most of them are or were involved in the conception of renowned books or tools dedicated to the French flora or in the study of large plant groups. In addition to the test set, the experts were provided an exhaustive list of 2 567 possible species, which is basically the subpart of the 10 000 species used in PlantCLEF2017 related to the French flora exclusively. Regarding the difficulty of the task and contrary to the previous human vs. machine experiment done in 2014, each participant was **allowed to use any external resource** (book, herbarium material, computational tool, web app, etc.) except automated plant identification tools (such as the Pl@ntNet app). For each plant, the experts were allowed to propose up to 3 species names ranked by decreasing confidence. 9 of 20 contacted experts finally completed the task on time and returned their propositions. In parallel, the research groups participating in LifeCLEF 2017 were asked to run their system on the same test set as the one sent to the experts. The three research groups who developed the top three performing systems in the challenge (Mario TSA/MNB, KDE TUT, and CMP = our submission) joined the effort and provided a total of 9 run files containing the species predictions.

4.3.3 Results

Progress Since 2014

Figure 4.7 reports the Mean Reciprocal Rank scores obtained by all human participants and all automated identification systems ("machines"). The description of the systems that were evaluated in 2014 ("Machine 2014") can be found in [18]). The main outcome of Figure 4.7 is the impressive progress that was made by machines between 2014 and 2017. This progress is mostly allowed by the use of recent deep convolutional neural network architectures, but also by using much larger training datasets: The systems from 2014 were trained on 60 962 images, while the systems from 2017 were trained on more than 250 000 images (for models using only the "trusted" data) and more than 1.1M images (for models using "noisy" data). Interestingly, the fact that the 2017 systems were trained on 10K species rather than 500 species did not affect their performance to much. In fact, this might even have increased the performance by allowing to learn better visual representations. One can notice that the quality of the identifications made by the best

---

[3] `http://otmedia.lirmm.fr/LifeCLEF/mvsm2017/` Last accessed 2nd Apr 2020.

evaluated system is very close to the one of the only highly skilled botanist (qualified as ”Expert of the flora” in Figure 4.7. Other participants, including the botanists who were not directly specialists on the targeted flora, were outperformed by the five machine learning submissions experimented in 2017.

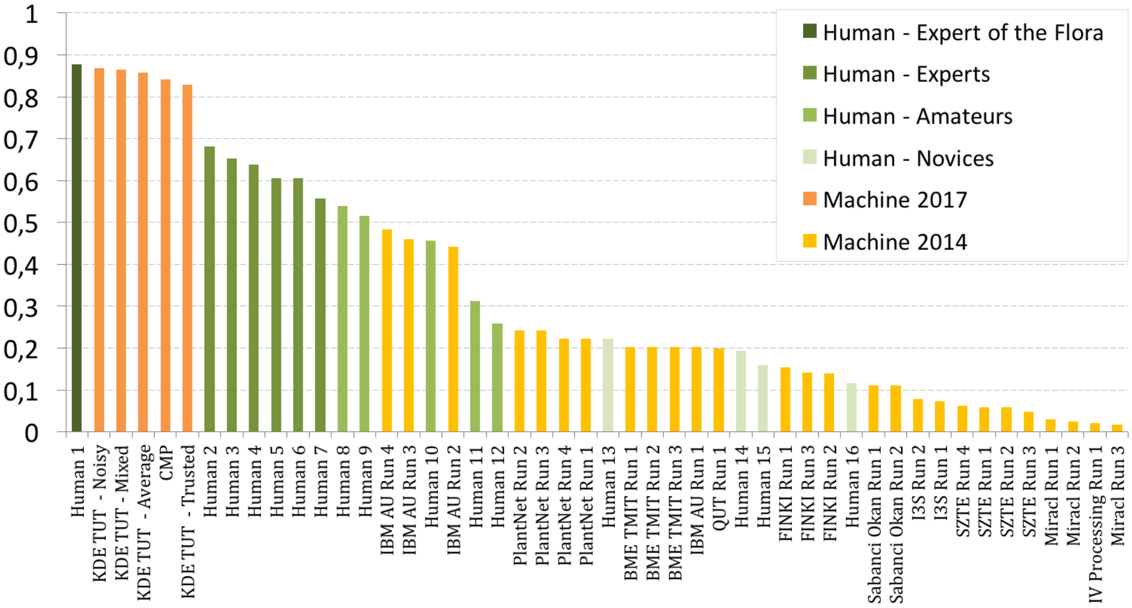


Figure 4.7: Identification performance of automated systems (2014 and 2017) and humans of various expertise on the 2014-th test set.

Experts vs. Machines in 2017

Figure 4.8 displays the top-1 identification accuracy achieved by both the experts and the automated systems. Table 4.5 reports additional evaluation metrics – namely the Mean Reciprocal Rank score, the top-2 accuracy and the top-3 accuracy. None of the botanists identified all observations correctly. The top-1 accuracy of the experts is in the range from 61% to 96%, with a median value of 80%. This illustrates the high difficulty of the task, especially when taking into account that the experts were allowed to use any external resource to complete the task, flora books in particular. It shows that a large part of the observations in the test set did not contain enough information to be surely identified when using classical identification keys. Only the three experts with an exceptional field expertise were able to correctly identify more than 80% of the observations. Figure 4.8 also shows that the top-1 accuracy of the evaluated machine learning systems is in the range from 56% to 74% with a median value of 66%. While this is lower than the median of experts, the best systems were able to perform similarly or slightly better than three of the highly skilled participating experts.

If we look at the top-3 accuracy values provided in Table 4.5, we can see that the best evaluated system returned the correct species within its top-3 predictions for more than 89% of the test observations. Only the two best experts obtained a higher top-3 accuracy. This illustrates one of the strengths of the automated identification systems: They can return an exhaustive ranked list of the most probable predictions over all species whereas this is a very difficult and painful task for human experts. Figure 4.9 displays the further

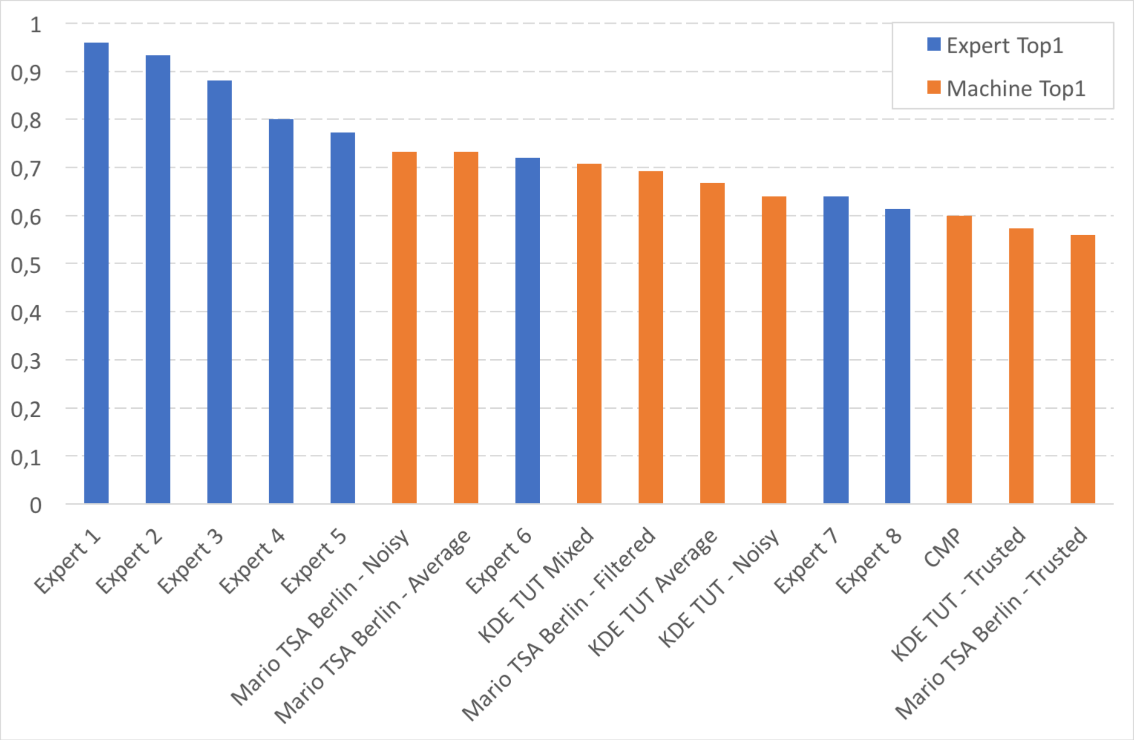


Figure 4.8: Identification performance achieved by machines and human experts for the Experts vs. Machines 2017 experiments.

Table 4.5: Results of the Experts vs. Machines 2017 experiments, ordered by the top 1 accuracy

Run	RunType	MRR (%)	Top1 (%)	Top2 (%)	Top3 (%)
Expert 1	man	96.7	96.0	97.3	97.3
Expert 2	man	94.7	93.3	96.0	96.0
Expert 3	man	88.0	88.0	88.0	88.0
Expert 4	man	80.0	80.0	80.0	80.0
Expert 5	man	78.0	77.3	78.7	78.7
Mario TSA Berlin - Noisy	machine	81.9	73.3	82.7	89.3
Mario TSA Berlin - Average	machine	80.5	73.3	81.3	85.3
Expert 6	man	74.0	72.0	76.0	76.0
KDE TUT Mixed	machine	78.6	70.7	80.0	82.7
Mario TSA Berlin - Filtered	machine	75.1	69.3	74.7	78.7
KDE TUT Average	machine	75.3	66.7	76.0	78.7
Expert 7	man	64.0	64.0	64.0	64.0
KDE TUT - Noisy	machine	75.0	64.0	80.0	81.3
Expert 8	man	62.0	61.3	62.7	62.7
CMP	machine	67.9	60.0	66.7	72.0
KDE TUT - Trusted	machine	65.6	57.3	61.3	72.0
Mario TSA Berlin - Trusted	machine	64.6	56.0	64.0	68.0

top-K accuracy values as a function of K for all the evaluated systems. It shows that the performance of all systems continues to increase significantly for values of K higher than 3 and then becomes more stable for values of K in the range from 20 to 50. The best system reaches a top-11 accuracy of 97.3%, *i.e.* the same value of the top-1 accuracy of the best expert, and a 100% top-K accuracy for $K = 39$. In view of the thousands of species in the whole check list, it is likely that such a system would be very useful even for the experts themselves. By providing an exhaustive short list of all the possible species, it would help them to not exclude any candidate species that they might have missed otherwise. Our (CMP) submission to the Experts vs. Machines experiment achieved 60% top-1 accuracy, i.e. lower than the median of the three best performing methods from PlantCLEF 2017, which is consistent with the challenge, where the CMP submissions scored 3rd.

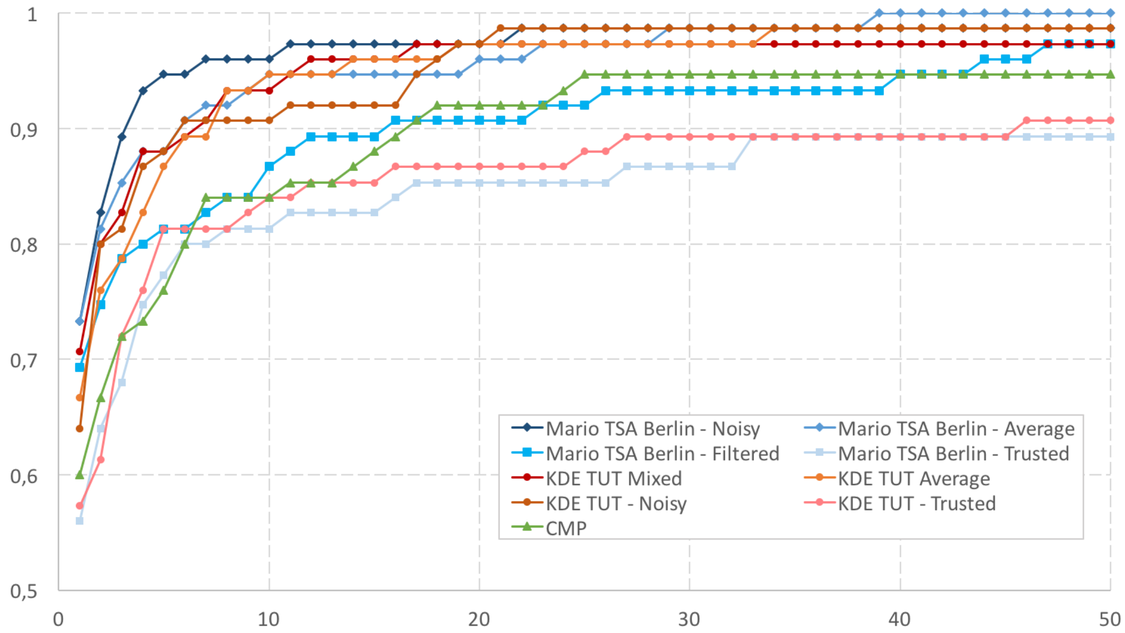


Figure 4.9: Top-K accuracy of the systems evaluated in the Experts vs. Machines 2017 experiments.

Table 4.6: Maximal top-1 and top-3 accuracy (%) across all human propositions, all machine predictions, or all of them together.

Accuracy	All humans	All machines	All humans & machines
Top 1	97.5	87.3	97.5
Top 3	98.7	93.7	**100.0**

To illustrate the possible synergy between experts and automated identification systems, Table 4.6 shows the top-1 and top-3 accuracy when considering the minimal rank of each test sample across either all human propositions, all machine predictions, or all of them together. The correct species is retrieved in the top-3 propositions of at least one expert in 98.7% of the cases, in the top-3 propositions of at least one system in 93.7% of the cases, and in the top-3 propositions of at least one of them all in 100% of the cases.

4.4 ExpertLifeCLEF 2018

Similarly to the previous LifeCLEF challenges, the goal of the ExpertLifeCLEF 2018 challenge was to assess the quality of automatic, machine-learned recognition systems for plant identification. This time the challenge was organized with the intention to directly compare the accuracy of the automatic systems with human experts in plant sciences, continuing with the comparison efforts described in Section 4.3. For practical reasons, the experts were evaluated on a small subset of the test data.

The data provided for the challenge cover 10 000 species of plants – herbs, trees and ferns – and consist from:

- PlantCLEF 2017 EoL: 256K images from the Encyclopedia of Life (EoL) [2] provided in the 2017 challenge [61] as the "trusted" training set.
- PlantCLEF 2017 web: 1.4M images automatically retrieved by web search engines, provided in the 2017 challenge [61] as the "noisy" training set.
- PlantCLEF 2017 test set: 25K test images from the 2017 challenge [61], now available with ground truth label annotations.
- PlantCLEF 2016 subset: 64K images from the PlantCLEF 2016 [60] challenge training- and test sets, covering only 717 of the 10k species. The remaining classes from the 2016 challenge do not exactly taxonomically match the 2017/2018 list of species.
- ExpertLifeCLEF 2018 test set: 6 892 unlabeled images used for evaluation of the submitted methods. Examples from the set are displayed in Figure 4.10.

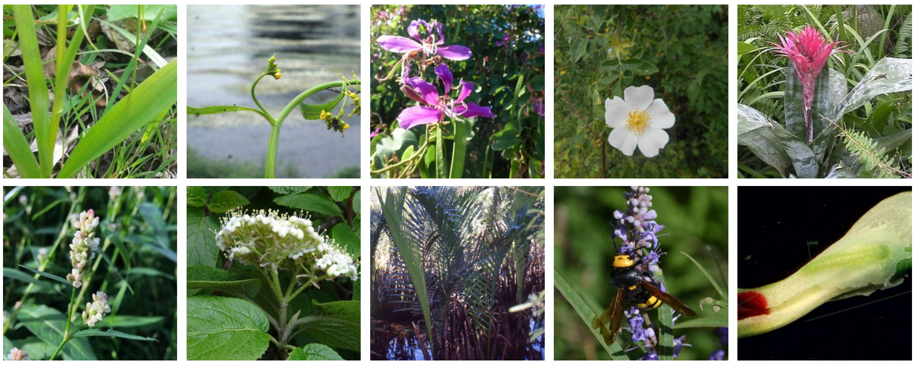


Figure 4.10: ExpertLifeCLEF 2018 test set - randomly selected samples.

The proposed classification system builds upon the state-of-the-art Convolutional Neural Network (CNN) architectures, described in Section 4.4.1. Section 4.4.1 discusses the use of running averages of the trained network parameters instead of values from the last training step which noticeably increased the accuracy of our models.

The class frequencies in the training data follow a long-tailed distribution. It is reasonable to expect that the training data, whose significant majority was downloaded from the web, have different class prior probabilities than the test set. In this challenge, we consider the problem of different class prior probability distributions described in Section

Table 4.7: Optimizer hyper-parameters, common to all our ExpertLifeCLEF 2018 experiments.

Parameter	Value
Optimizer	rmsprop
RMSProp momentum	0.9
RMSProp decay	0.9
Initial learning rate	0.01
Learning rate decay type	Exponential
Learning rate decay factor	0.94

3.3 and use the existing EM algorithm [161] for Maximum Likelihood estimate of the new class priors, as described in Section 3.3.2.

Section 4.4.2 describes the 5 submissions we made. Results of the challenge are presented in Section 4.4.3. One of the submitted plant recognition methods achieved the best accuracy among automated systems, and thus placed 1st in the challenge. It outperformed 5 of 9 human experts.

4.4.1 Methodology

Convolutional Neural Networks

The proposed method is based on two architectures – Inception-ResNet-v2 and Inception-v4 [186] – and their ensembles described in Section 4.4.2. The TensorFlow-Slim API was used to adjust and fine-tune the networks from the publicly available ImageNet-pretrained checkpoints[4]. All networks in our experiments shared the optimizer settings listed in Table 4.7. The batch size, input resolution and random crop area range were set differently for each network listed in Table 4.8.

The following image pre-processing was used for training:

- Random crop, with aspect ratio range $(0.75, 1.33)$ and with different area ranges listed in Table 4.8,
- Random left-right flip,
- Brightness and Saturation distortion.

At test-time, 14 predictions per image are generated by using 7 crops and their mirrored versions:

- 1x Full image,
- 1x Central crop covering 80% of the original image,
- 1x Central crop covering 60% of the original image,
- 4x corner crops covering 60% of the original image.

---

[4] https://github.com/tensorflow/models/tree/master/research/slim/#pretrained Last accessed 2nd Apr 2020.

Table 4.8: Networks and hyper-parameters used in the experiments:

#	Net architecture	Batch size	Input Resolution	Random crop area
1	Inception-ResNet v2	32	299×299	5% - 100%
2	Inception-ResNet v2	16	498×498	25% - 100%
3	Inception-ResNet v2	16	498×498	5% - 100%
4	Inception v4	32	299×299	5% - 100%
5	Inception v4	32	598×598	5% - 100%
6	Inception v4	32	299×299	50% - 100%

Fine-tuning and Data Splits

Networks $\#1, \ldots, \#6$, initialized from the ImageNet pre-trained checkpoints, were first trained on PlantCLEF data from previous years (PlantCLEF 2017 EoL + PlantCLEF 2017 web + PlantCLEF 2016 subset). PlantCLEF 2017 test set was used for validation.

Another set of networks, denoted as $\#1^{\text{clean}}, \ldots, \#6^{\text{clean}}$, was fine-tuned from models $\#1, \ldots, \#6$ without using the noisy PlantCLEF 2017 web set. For this fine-tuning, we added most of the PlantCLEF 2017 test set, keeping only 1 000 observations (1 403 images) as a min-val set.

Running Averages

Preliminary experiments, using the 2017 test set for validation, showed a significant improvement in accuracy when using Polyak averaging [148], i.e. using running averages of the trained variables instead of the values from the last training step. Namely we used an exponential decay with decay rate of 0.999.

In this task where majority of the training data is noisy, we interpret this as keeping a stable version of the variables, since mini-batches with noisy samples may produce large gradients pointing outside of the local optima. Another possible interpretation is that the learning rate was still too high. Unfortunately, we did not have the computational time to experiment with different learning rate schedules.

Class Prior Estimation

In many computer vision tasks, the class prior probabilities are assumed to be the same for the training data and test data. In ExpertLifeCLEF, however, it is reasonable to assume that class priors change: The largest part of the training set comes from the web, where the class frequencies may not correspond with the test-time priors (depending on the species incidence, the interest of users, etc.). In Section 3.3.1, we discussed the problem of adjusting CNN outputs to the change in class prior probabilities and proposed to recompute the posterior probabilities (predictions) $p(k|\mathbf{x})$ by Equation 3.20.

For the estimation of the new (test set) priors, we used the maximization of the likelihood of the test observations as discussed in Section 3.3.2. Specifically, we used the EM algorithm of Saerens et al. [161], where the E and M steps were described in Equations

3.23 and 3.24 respectively, i.e.:

$$p^{(s)}(k|\mathbf{x}) = \frac{p(k|\mathbf{x})\dfrac{\hat{p}_k^{(s)}}{p_Y(k)}}{\sum\limits_{j=1}^{K} p(j|\mathbf{x})\dfrac{\hat{p}_j^{(s)}}{p_Y(j)}}$$

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

In our submissions, we estimated the class prior probabilities for the whole test set. However, one may consider estimating different class priors for different locations, based on the GPS-coordinates of the observations. Moreover, as discussed later in Section 6.4, one may use this procedure even in the cases where the new test samples come sequentially.

4.4.2 Submissions

In the challenge, each team was allowed to submit up to 5 different run-files with predictions. We used this opportunity to evaluate the following 5 submissions:

CMP Run 1 is an ensemble of 6 CNNs: $\#1^{\text{clean}}, \ldots, \#6^{\text{clean}}$ described in Section 4.4.1. This submission used the automatic test set class-prior estimation from the CNN outputs, discussed in Section 4.4.1.

CMP Run 2 is an ensemble of the same 6 CNNs as in CMP Run 1, but without the class prior estimation on the test data.

CMP Run 3 is an ensemble of 12 CNNs: $\#1, \ldots, \#6$ and $\#1^{\text{clean}}, \ldots, \#6^{\text{clean}}$ described in Section 4.4.1. This submission used the automatic test set class-prior estimation.

CMP Run 4 is an ensemble of 6 CNNs: $\#1, \ldots, \#6$ described in Section 4.4.1. This submission used the automatic test set class-prior estimation.

CMP Run 5 is a single Inception-v4 model, denoted as CNN $\#4^{\text{clean}}$, using the automatic test set class-prior estimation.

In all runs, the predictions (optionally improved by the class prior estimation) for all crops of the test image are averaged to compute the final image prediction. Moreover, for observations with several images (connected by the ObservationID values in the provided data), the final classification decision is taken based on the average of all corresponding image predictions.

4.4.3 Results

The official results of the challenge are displayed in Figure 4.11. Our system achieved the best results among automatic methods: 88.4% accuracy on the full test set. The best scoring submission was the largest ensemble - CMP Run 3 - using all 12 models. Results of all CMP submissions are listed in Table 4.9.

When evaluated against human experts in plant sciences, the system (both CMP Run 3 and CMP Run 4) outperformed 5 of 9 tested human experts. That means that in the

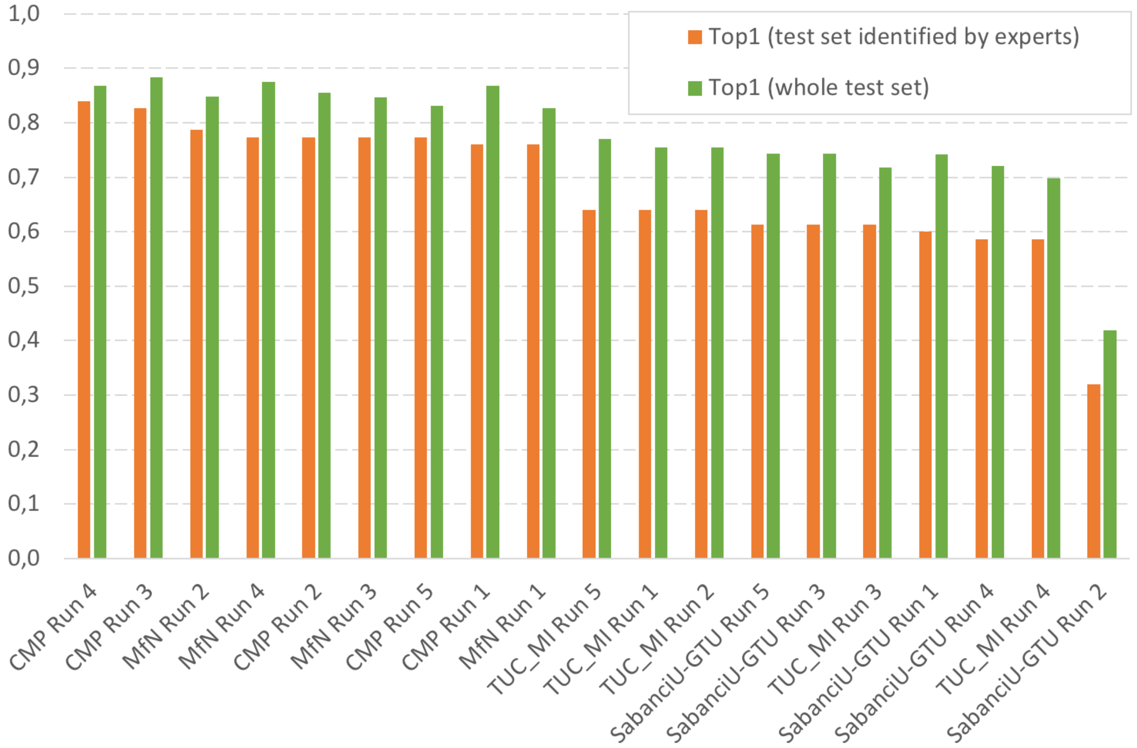


Figure 4.11: Results of runs submitted by the challenge participants.

task of plant recognition from images, machine learning systems reached human expert performance - achieving better accuracy than the median of human experts. The detailed results are displayed in Figure 4.12.

Interestingly, while fine-tuning on "clean" data slightly improved the recognition accuracy on the full test set, it significantly decreased the accuracy on the test subset for human experts. Similarly, test-time prior estimation on the full test set noticeably improved the accuracy, but had an opposite effect on the subset. We assume that the test subset selected for human experts was too small to provide a representative, identically distributed, sample of the full test set. Therefore the results on the test subset for human experts may be biased towards a small number of species contained in it.

4.4.4 Discussion

The proposed machine-learning system for recognition of 10 000 plant species achieved an excellent accuracy of 88.4% in the ExpertLifeCLEF 2018 challenge, scoring 1st among automated systems.

The ensemble of Convolutional Neural Networks benefited from the following improvements:

Table 4.9: Results of CMP submissions on the full test set and its subset for human experts.

CMP Run	1	2	3	4	5
Accuracy (full test set)	86.8%	85.6%	**88.4%**	86.7%	83.2%
Accuracy (smaller test set)	76.0%	77.3%	82.7%	**84.0%**	77.3%

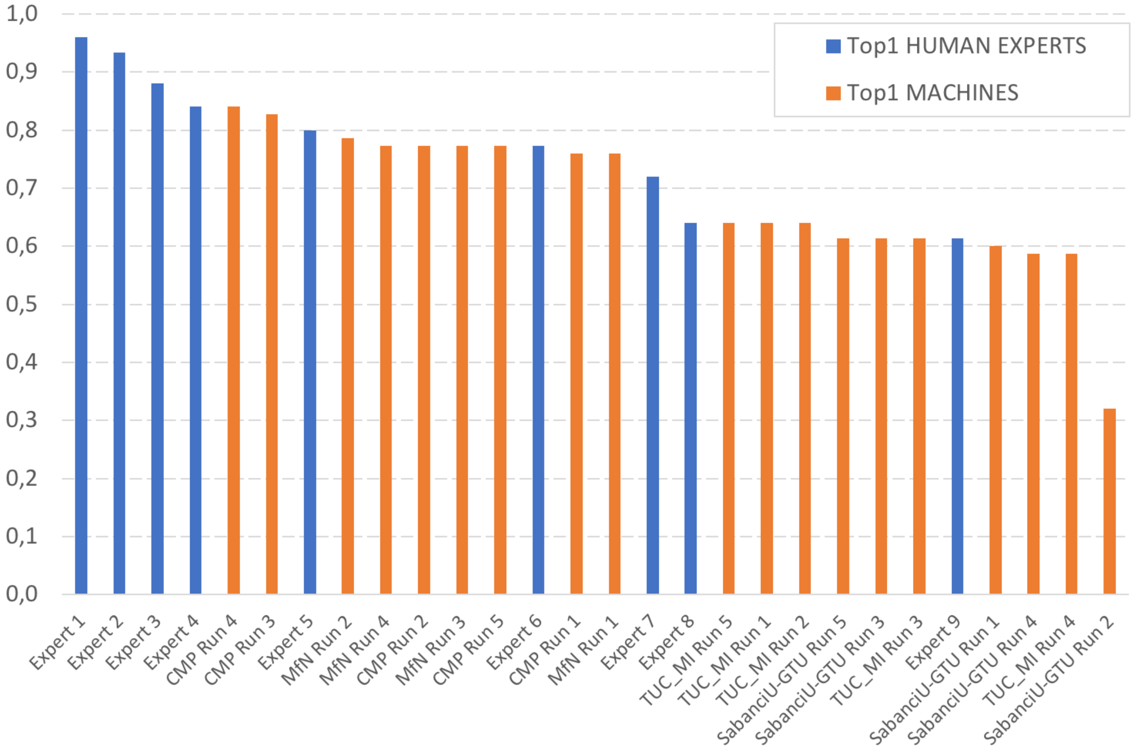


Figure 4.12: Results of the "Experts vs Machines" experiment.

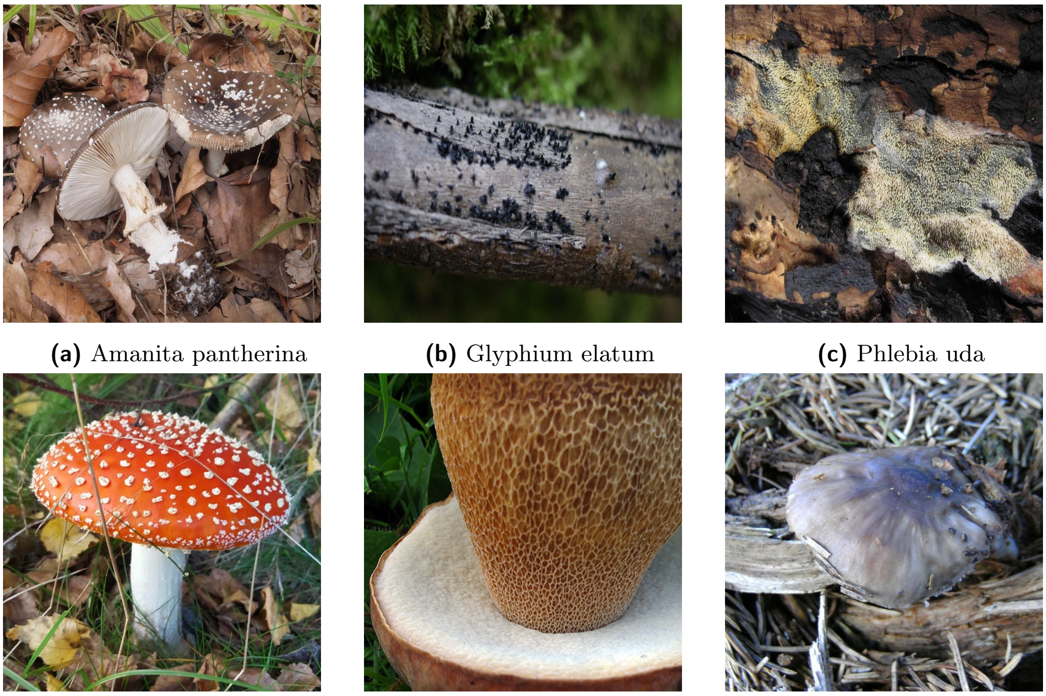
1. Adjusting the CNN predictions according to the estimated change of the class prior probabilities.
2. Replacing network parameters by their running averages with exponential decay.
3. Test-time data augmentation.

The experiment with human experts shows that machine learning reached the expert knowledge in plant recognition: our system scored better than an average (median) human expert in plant recognition, achieving better recognition rate than 5 of the 9 evaluated human experts. However, it is important to note that human experts are usually specialized in a more active recognition approach, such as studying a specimen from different views, and - unlike our models - the experts are able to describe the reasoning for the species identification.

4.5 Fine-Grained Visual Categorization Challenges 2018: FGVCx Fungi, and FGVCx Flowers, iNaturalist

The 5th Fine-Grained Visual Categorization (FGVC) workshop at CVPR 2018 organized several computer vision challenges focusing on different applications of fine-grained recognition:

- classification of product attributes in the iMaterialist 2018 Fashion Challenge[5] and iMaterialist 2018 Furniture Challenge[6],
- identification of food items in an image in the iFood 2018 Challenge[7],
- analyzing if images from camera traps ("wild cams") captured an animal in the iWildCam 2018 Challenge[8],
- fine-grained classification of almost 1 400 fungi species in the FGVCx Fungi Classification Challenge[9],
- fine-grained classification of almost 1 000 plant species in the FGVCx Flower Classification Challenge[10],
- large scale classification of over 8 000 species in the iNaturalist 2018 Challenge[11].



(a) Amanita pantherina **(b)** Glyphium elatum **(c)** Phlebia uda

(d) Amanita muscaria **(e)** Boletus reticulatus **(f)** Pluteus pouzarianus

Figure 4.13: Examples from the FGVCx Fungi training set.

---

[5] `http://www.kaggle.com/c/imaterialist-challenge-fashion-2018` Last accessed 2nd Apr 2020.

[6] `http://www.kaggle.com/c/imaterialist-challenge-furniture-2018` Last accessed 2nd Apr 2020.

[7] `http://www.kaggle.com/c/ifood2018/` Last accessed 2nd Apr 2020.

[8] `http://www.kaggle.com/c/iwildcam2018` Last accessed 2nd Apr 2020.

[9] `http://www.kaggle.com/c/fungi-challenge-fgvc-2018` Last accessed 2nd Apr 2020.

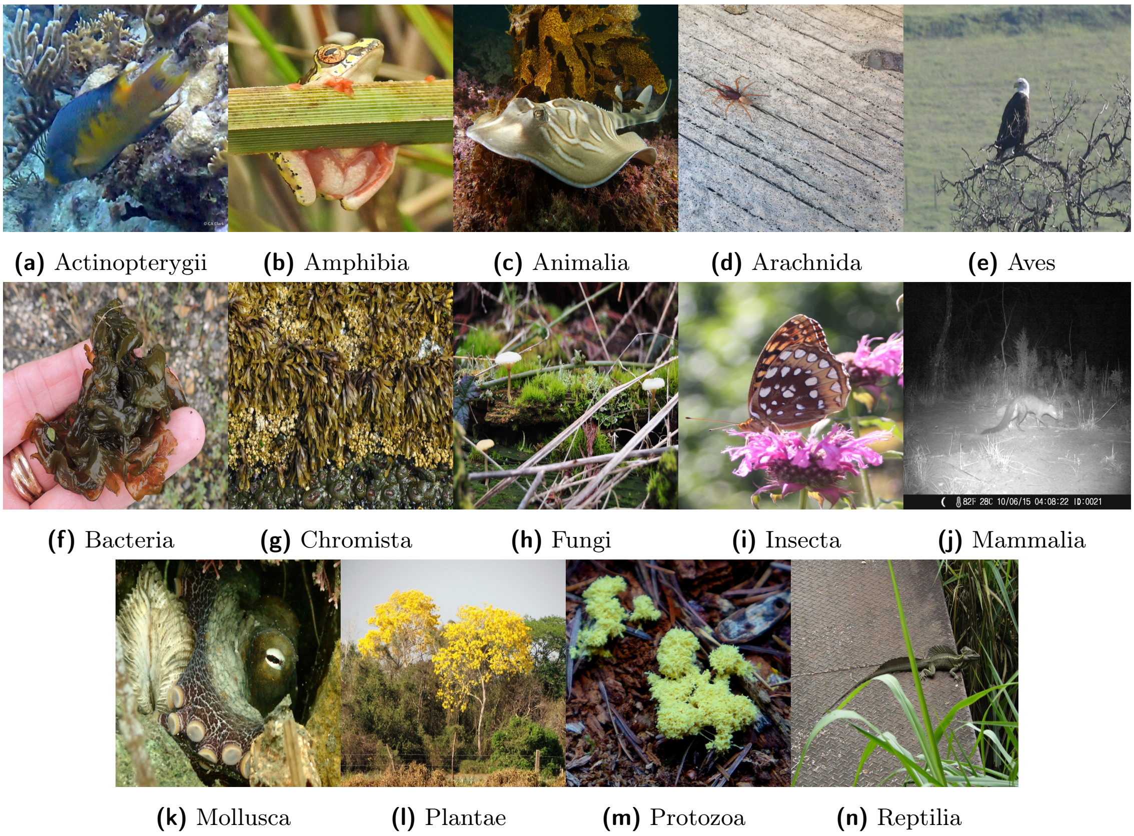
[10] `http://www.kaggle.com/c/fgvc2018-flower` Last accessed 2nd Apr 2020.

[11] `http://www.kaggle.com/c/inaturalist-2018` Last accessed 2nd Apr 2020.

We joined the three species identification tasks. The datasets provided for the FGVCx Fungi, FGVCx Flowers and iNaturalist challenges are described in Sections 4.5.1, 4.5.2 and 4.5.3 respectively. The methodology and submissions to all three challenges are described in Section 4.5.4 and the challenge results are summarized in Section 4.5.5.

4.5.1 FGVCx Fungi Dataset

The FGVCx Fungi Classification Challenge provided an image dataset, that covers 1394 fungal species and is split into a training set with 85 578 images, a validation set with 4182 images and a a competition test set with 9758 images without publicly available labels. Examples from the FGVCx Fungi training set are shown in Figure 4.13. There is a substantial change of categorical priors $p_Y(k)$ between the training set and the validation set: The distribution of images per class is highly unbalanced in the training set, while the validation set distribution is uniform.



(a) Actinopterygii **(b)** Amphibia **(c)** Animalia **(d)** Arachnida **(e)** Aves

(f) Bacteria **(g)** Chromista **(h)** Fungi **(i)** Insecta **(j)** Mammalia

(k) Mollusca **(l)** Plantae **(m)** Protozoa **(n)** Reptilia

Figure 4.14: The iNaturalist training set: one example from each super-category.

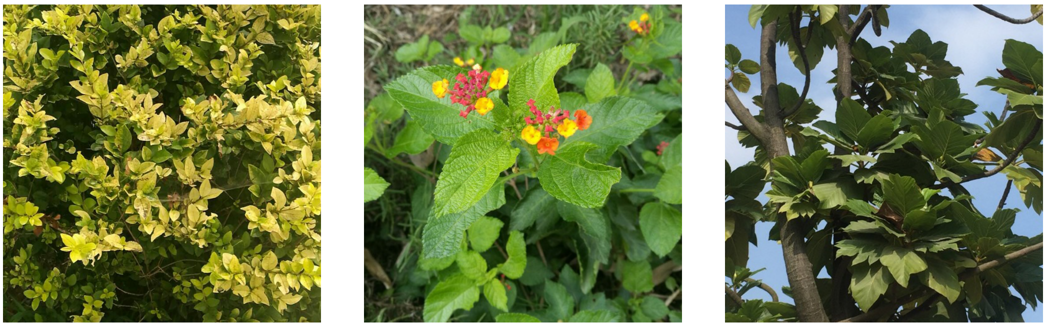
4.5.2 iNaturalist 2018 Dataset

The iNaturalist 2018 challenge was a large scale fine-grained species recognition competition, providing a dataset of 8142 species from 14 super-categories: *Plantae*, *Aves*, *Reptilia*, *Amphibia*, *Mammalia*, *Fungi*, *Actinopterygii*, *Chromista*, *Protozoa*, *Mollusca*, *Insecta*, *Arachnida*, *Bacteria* and *Other*. However, the species names in the dataset were replaced with their other unique identifiers in order to prevent competitors from downloading additional images, as additional data sources were not allowed in this competition. The labeled dataset was split into a training set of 437 513 images and a validation set of 24 426 images. Figure 4.14 displays examples from the training set.

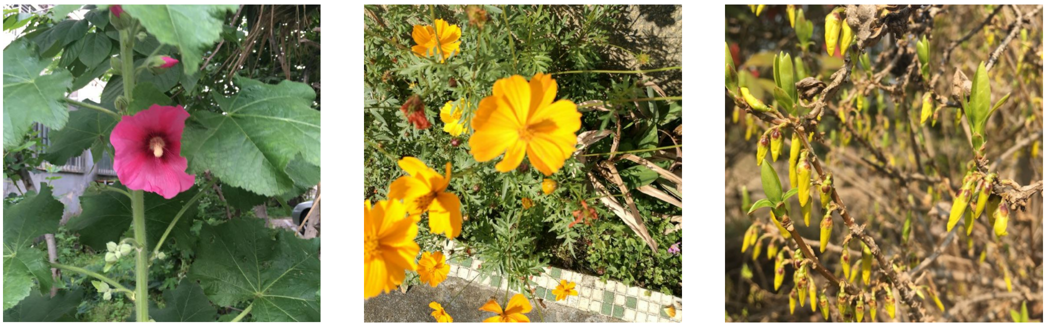
Similarly to FGVCx Fungi dataset described in Section 4.5.1, there was a change in the categorical distribution in the iNaturalist dataset: The training set had a long-tailed species distribution, while the species in the validation set was uniform (3 images per species). The challenge was evaluated on a test set of 149 394 images without publicly available labels.

4.5.3 FGVCx Flowers Dataset

The FGVCx Flower Classification Challenge dataset covers 997 species of flowering plants. The dataset provided by Xingse[12] and PictureThis[13] consists of 669 304 training images and 12 961 test images without publicly available labels. Note that no validation set was provided for this challenge. Examples from the FGVCx Flowers training set are shown in Figure 4.15.



(a) Ligustrum vicaryi **(b)** Lantana camara **(c)** Artocarpus communis



(d) Alcea rosea.jpg **(e)** Cosmos sulphureus **(f)** Forsythia viridissima

Figure 4.15: Examples from the FGVCx Flowers training set.

4.5.4 Method

Our submissions for all the FGVC challenges were based on the Inception-v4 and Inception-ResNet-v2 architectures [186], inspired by the winning submission to ExpertLifeCLEF 2018

---

[12] https://www.xingseapp.com/ Last accessed 2nd Apr 2020.

[13] https://www.picturethisai.com/ Last accessed 2nd Apr 2020.

described in Section 4.4.

All networks were trained using the Tensorflow Slim[14] framework. We used Polyak averaging [148], keeping shadow variables with exponential moving averages of the trained variables. The following hyper-parameters were used for the training of all models in this Section:

- Optimizer: RMSprop
- Batch size: 32
- Initial learning rate: 0.01
- Learning rate decay type: exponential/staircase
- Learning rate decay factor: 0.94
- Weight decay: 0.00004
- Moving average decay: 0.999

Adjusting Predictions by Class Priors

Let us assume that the classifier trained by cross-entropy minimization learns to estimate the posterior probabilities, i.e. $f_{\text{CNN}}(k|x) \approx p(k|x)$, as discussed in Section 3.2. If the class prior probabilities $p_Y(k)$ change, the posterior probabilities should change as well. The topic of adjusting CNN predictions to new priors was discussed in Section 3.3.1: in the case when the new class priors $p_Y^e(k)$ are known, the new posterior $p^e(k|x)$ can be computed from Equation 3.20 as:

$$p^e(k|\mathbf{x}) \propto p(k|\mathbf{x}) \frac{p_Y^e(k)}{p_Y(k)},$$

We assume that the uniform distribution $p_Y^e(k) = \frac{1}{K}$ is given, as it is the case of the FGVCx Fungi and iNaturalist validation sets described in Sections 4.5.1 and 4.5.2 respectively. Then:

$$p^e(k|\mathbf{x}_i) \propto \frac{p(k|\mathbf{x}_i)}{p_Y(k)}. \tag{4.8}$$

Test-time Image Augmentation

We considered the following 14 image augmentations at test time: The original image; additional 6 crops of the original image with 80% (central crop) and 60% (central crop + 4 corner crops) of the original image width/height; and the mirrored versions of the 7 foregoing augmentations. All augmentations are then resized to square inputs using bilinear interpolation.

Predictions from all augmentations are then combined by averaging (sum) or mode (i.e. the most frequent prediction) of the predicted species.

---

[14] https://github.com/tensorflow/models/tree/master/research/slim/
Last accessed 2nd Apr 2020.

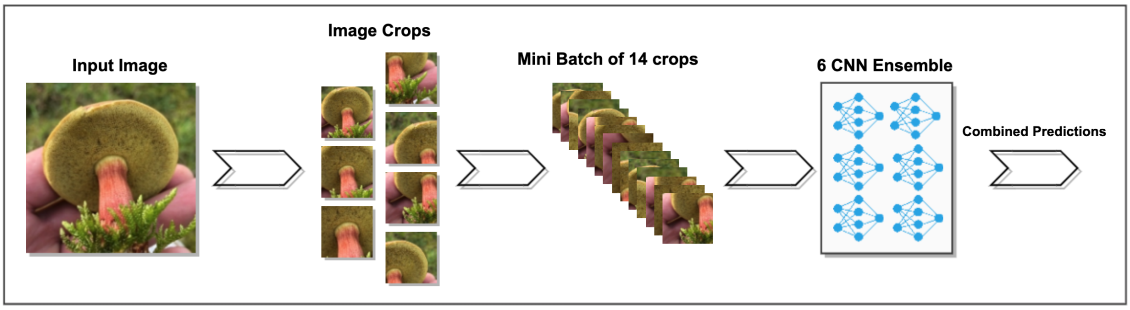


Figure 4.16: FGVCx Fungi 2018: Predictions combined from an ensemble of 6 CNNs with test-time image augmentation (crops, mirrors).

Ensembles

For the FGVCx Fungi recognition challenge, we trained an ensemble of 6 CNNs listed in Table 4.10. The predictions of all ensemble models and test-time image augmentations were combined by mode, i.e. the final prediction was the species appearing most often as the top-1 result among the 84 predictions (6 models × 7 crops × 2 mirror). The pipeline is illustrated in Figure 4.16.

Table 4.10: Models trained for the FGVCx Fungi classification competition.

CNN	Architecture	Input Size	Fine-tuned from
#1	Inception-v4	299x299	ImageNet 2012
#2	Inception-v4	299x299	LifeCLEF 2018
#3	Inception-v4 “x2”	598x598	ImageNet 2012
#4	Inception-v4 “x2”	598x598	LifeCLEF 2018
#5	Inc.-ResNet-v2	299x299	ImageNet 2012
#6	Inc.-ResNet-v2	299x299	LifeCLEF 2018

Table 4.11: Models trained for the FGVCx Flowers classification competition.

CNN	Architecture	Input Size	Fine-tuned from
#1	Inception-v4	299x299	ImageNet 2012
#2	Inception-v4	299x299	LifeCLEF 2018
#3	Inception-v4	299x299	iNaturalist 2018
#4	Inception-v4 “x2”	598x598	LifeCLEF 2018
#5	Inc.-ResNet-v2	299x299	LifeCLEF 2018

The ensemble for the FGVCx Flower classification challenge consisted of 5 CNNs listed in Table 4.11. Predictions of the 5 ensemble models and 14 test-time image augmentations were combined by averaging.

The ensemble for the iNaturalist classification challenge consisted of 11 CNNs listed in Table 4.12. Same as in the FGVCx Flower recognition challenge, the predictions of all ensemble models and test-time image augmentations were combined by averaging.

Table 4.12: Models trained for the iNaturalist classification competition.

CNN	Architecture	Input Size	Fine-tuned from	Trained on iNaturalist
#1	Inception-v4	299x299	ImageNet 2012	training set
#2	Inception-v4 “x2”	598x598	ImageNet 2012	training set
#3	Inc.-ResNet-v2	299x299	ImageNet 2012	training set
#4	Inception-v4	299x299	LifeCLEF 2018	training set
#5	Inception-v4 “x2”	598x598	LifeCLEF 2018	training set
#6	Inc.-ResNet-v2	299x299	LifeCLEF 2018	training set
#7	Inception-v4	299x299	CNN #1	training + validation set
#8	Inception-v4 “x2”	598x598	CNN #2	training + validation set
#9	Inc.-ResNet-v2	299x299	CNN #3	training + validation set
#10	Inception-v4	299x299	CNN #4	training + validation set
#11	Inc.-ResNet-v2	299x299	CNN #6	training + validation set

4.5.5 Results

First, we evaluate the accuracy of our models on the validation set before and after applying techniques like test-time augmentation, ensembling, or adjusting predictions to new class priors. Second, the official challenge results are summarized.

FGVCx Fungi Validation Dataset

Let us first validate the CNNs trained for the FGVCx Fungi Classification challenge on the FGVCx Fungi validation set. Table 4.13 compares the six trained CNN models before applying additional techniques, with 1 forward pass (central crop, 80%) per image. We will continue the validation experiments with CNN 1, i.e. Inception-v4 pre-trained from an ImageNet checkpoint, which achieved the best validation accuracy.

The test-time pre-processing of the image input makes a noticeable difference. Table 4.14 shows the difference in accuracy for different sizes of central crop of the original image. Table 4.15 compares the validation scores of the best performing CNN #1 against the ensemble of all 6 networks, and measures the effect of the proposed multi-crop evaluation.

The advantage of adjusting the predictions with the new categorical prior is shown in Figure 4.17: at the end of training the accuracy increases by 3.8%, from 48.8% to 52.6%.

Table 4.13: Accuracy and Recall@5 of individual networks (central crop, 80%) on the FGVCx Fungi validation set.

CNN	Acc. (%)	R@5 (%)
#1 Inception-v4 (ImageNet)	48.8	77.0
#2 Inception-v4 (LifeCLEF)	48.5	75.8
#3 Inception-v4 “x2” (ImageNet)	48.6	76.6
#4 Inception-v4 “x2” (LifeCLEF)	48.8	76.2
#5 Inc.-ResNet-v2 (ImageNet)	47.7	76.0
#6 Inc.-ResNet-v2 (LifeCLEF)	47.4	75.8
Inception-v4 [43]	44.7	73.5

Table 4.14: Inception-v4 (fine-tuned from the ImageNet checkpoint) with differently sized central crops. Top-1 Accuracy and Recall@5 on the FGVCx Fungi validation set.

Central crop	Accuracy (%)	Recall@5 (%)
100%	45.9	75.1
80%	**48.8**	**77.0**
60%	48.6	76.3
40%	43.1	69.3

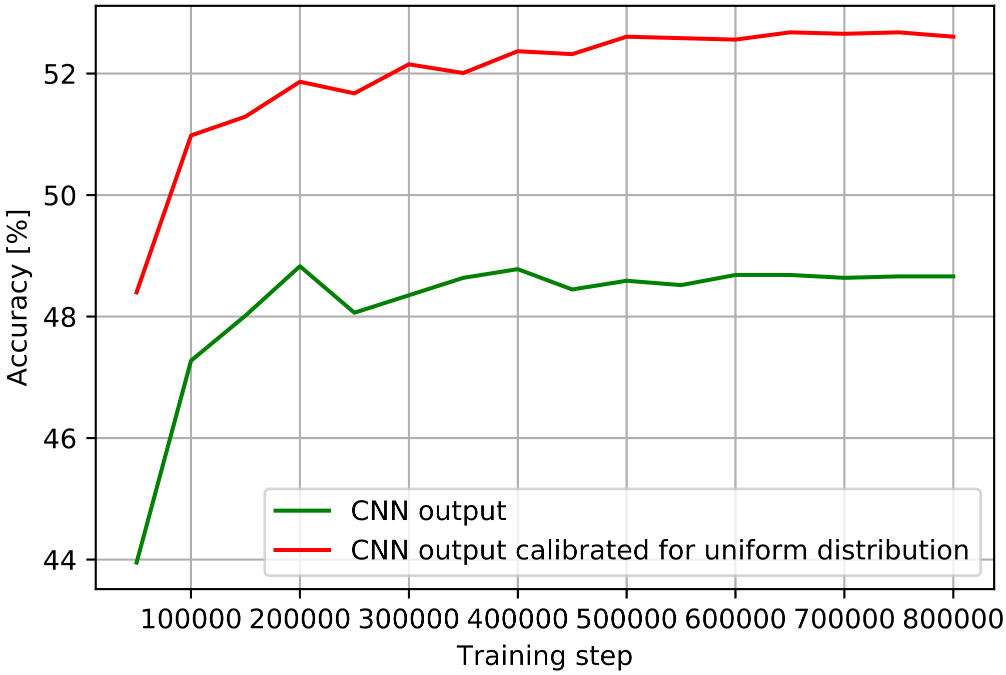


Figure 4.17: Accuracy of Inception-v4 (fine-tuned from ImageNet checkpoint) on the FGVCx Fungi validation set, before (green) and after (red) adjusting the predictions by $p_Y^e(k)$.

Table 4.15: Top-1 recognition accuracy on the FGVCx Fungi validation set: single CNN (#1) vs. ensemble (#1, . . . , #6) and single central crop (1) vs. multiple crops (14). Predictions from ensembles and crops were combined by averaging (sum) or by choosing the most common top prediction (mode). Results are shown both before and after adjusting the predictions by known $p_Y^e(k)$.

			Accuracy (%)	
#CNNs	Crops	Pool	Baseline	Known $p_Y^e(k)$
1	1	–	48.8	52.6
1	14	sum	51.8	56.0
6	1	sum	54.1	58.5
6	14	sum	**54.2**	**60.3**
6	14	mode	**54.2**	59.1

FGVCx Fungi Competition

The test dataset for the FGVCx Fungi Classificatin competition on Kaggle was divided into two parts - public and private. Public results were calculated with approximately 70% of the test data and results were visible to all participants. The rest of the data was used for final competition evaluation to avoid any possible bias towards performance on the test images.

Table 4.16: Results of the top ten teams in the FGVCx Fungi classification challenge. Source: `http://kaggle.com/c/fungi-challenge-fgvc-2018/leaderboard` Last accessed 2nd Apr 2020.

		Recal@3 Error (%)	
#	Team Name	Private Score	Public Score
1	CMP (ours)	**21.197**	**20.772**
2	digitalspecialists	23.188	23.471
3	Val An	25.091	25.213
4	DL Analytics	28.341	26.853
5	Invincibles	28.751	28.493
6	Tian Xi	32.235	31.636
7	Igor Krashenyi	32.616	34.164
8	wakaka	42.219	41.339
9	George Yu	47.621	47.113
10	Xinshao	67.837	67.509

We chose our best performing system, i.e. the ensemble of the 6 fine-tuned CNNs with 14 crops per test image and with predictions adjusted to new class priors, for the final submission to Kaggle. The accumulation of predictions was done by the mode from top species per prediction, as it had better preliminary scores on the public Kaggle test set.

Our submission to the challenge achieved the best scores in terms of Recall@3 error both in the public and private leaderboard. The Recall@3 error is defined as follows: for each image, if the ground truth label is found among the top 3 predicted labels, the error is 0, otherwise it is 1. The final score is the error averaged across all images. The results of the top 10 teams are listed in Table 4.16.

iNaturalist Competition

The test dataset for the iNaturalist competition on Kaggle was also divided into two parts - public and private. Public results were calculated with approximately 70% of the test data, while rest of the data was used for final competition evaluation.

For the final submission, we used the ensemble of the 11 fine-tuned CNNs from Table 4.12 with 14 crops per test image and with predictions adjusted to new class priors. The accumulation of predictions was done by the mode from top species per prediction, as it had better preliminary scores on the public part of iNaturalist test set on Kaggle.

Similarly to the FGVCx Fungi classification challenge, the iNaturalist competition also used the Recall@3 error. The results of the top 10 teams in the challenge are listed in Table 4.17. Our submission scored fourth in the competition (i.e. had the fourth best top3 accuracy), and had the third best top1 accuracy, as displayed in Figure 4.18.

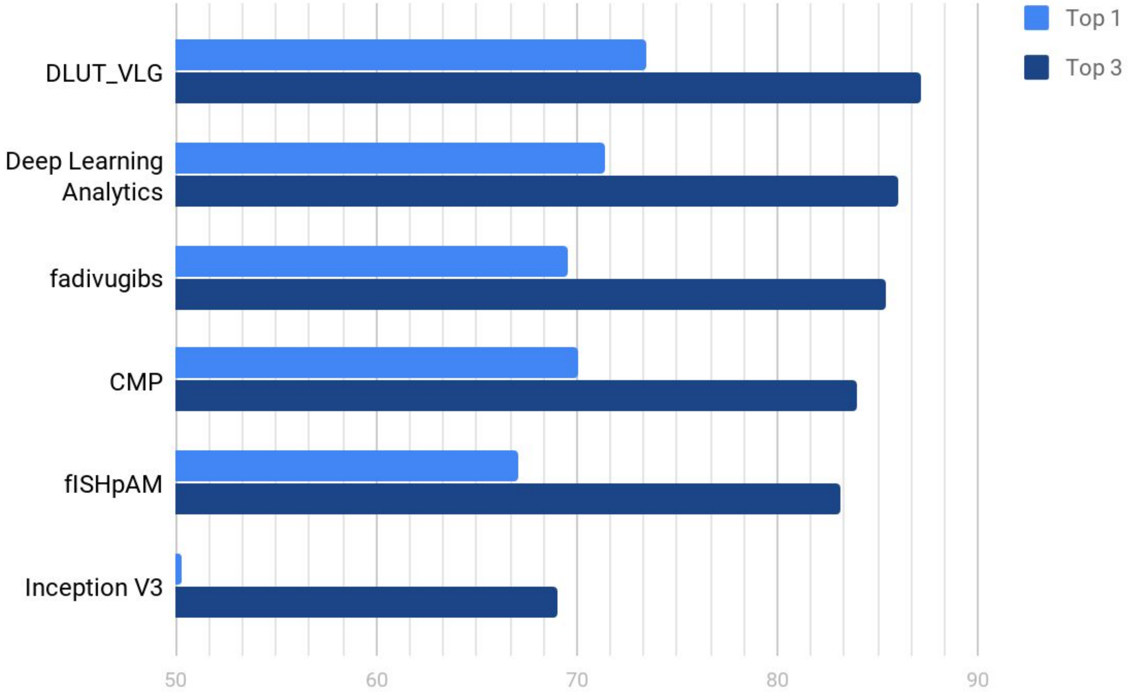


Figure 4.18: Top-1 and top-3 accuracy on the private test set of iNaturalist 2018. Source: Competition presentation https://www.dropbox.com/s/52nz6qc3zcwqhoa/iNaturalist_Competition_FGVC_2018.pdf (Last accessed 2nd Apr 2020.)

Table 4.17: Results of the top ten teams in the iNaturalist 2018 species classification challenge. Source: `https://www.kaggle.com/c/inaturalist-2018/leaderboard` (Last accessed 2nd Apr 2020.)

		Recal@3 Error (%)	
#	Team Name	Private Score	Public Score
1	DLUT VLG	12.858	13.068
2	Deep Learning Analytics	13.981	14.214
3	fadivugibs	14.618	14.914
4	CMP (ours)	16.076	16.360
5	fISHpAM	16.892	17.149
6	traveler	16.988	17.235
7	yen	17.201	17.412
8	Shuang	18.357	18.549
9	Mr.M	20.092	20.291
10	Dequan Wang	20.814	21.157

In the competition presentation [15], the winner of iNaturalist 2018 - DLUT VLG, a team from the Dalian University of Technology - mentioned the following techniques increasing the classification accuracy:

- CNN with second-order pooling denoted Matrix Power Normalized Covariance (MPN-COV) [115, 201],
- exploiting higher resolution images by increasing the network input size, and performing dense crops at multiple scales on test images for inference,
- pre-training ResNet-152 on ImageNet-11k, then fine-tuning on iNaturalist 2017, then two-stage training of MPN-COV on iNaturalist 2018,
- dealing with the class imbalance on the training set by fine-tuning on the validation set with uniformly distributed classes.

Note that the winner of iNaturalist 2018 also participated in the FGVCx Flowers competition described below.

FGVCx Flowers Competition

Same as in the previous two competitions, the test dataset of the FGVCx Flowers classification competition on Kaggle was divided into two parts - public and private. Public results were calculated with approximately 60% of the test data, the rest was used for final competition evaluation. Unlike the previous FGVC challenges, the FGVCx Flowers competition used the top1 accuracy as the main score.

---

[15] `https://www.dropbox.com/s/52nz6qc3zcwqhoa/iNaturalist_Competition_FGVC_2018.pdf` Last accessed 2nd Apr 2020.

Table 4.18: Results of the top ten teams in the FGVCx Flowers classification challenge. Source: `https://www.kaggle.com/c/fgvc2018-flower/leaderboard` (Last accessed 2nd Apr 2020.)

		Top 1 Error (%)	
#	Team Name	Private Score	Public Score
1	CMP (ours)	7.599	6.828
2	fadivugibs	8.177	7.638
3	DLUT VLG	8.242	7.677
4	yen	8.396	7.716
5	xiaoxiao	9.579	8.641
6	NDer MJU	11.636	10.976
7	thesouthfrog	15.211	13.618
8	nimahai	16.368	15.258
9	Miroslav Štola	20.187	19.637
10	lmao	20.342	20.177

Our final submission, averaging the predictions of the 5 fine-tuned CNNs from Table 4.11 with 14 crops per test image and with predictions adjusted to new class priors, scored first in the competition. The results of the top 10 teams in the challenge are listed in Table 4.18.

4.5.6 Discussion

Our submissions to the FGVC challenges, based on our winning submission to ExpertLifeCLEF 2018 described in 4.4, achieved excelent results in the challenges: 1st place in the FGVCx Fungi recognition challenge and in the FGVCx Flower recognition challenge, and 4th place in the iNaturalist species recognition challenge. The results confirm the suitability of our deep learning approach to species identification.

The FGVCx Fungi recognition challenge helped us to get in touch with the challenge sponsor, the Danish Mycological Society, and discuss further application of our fungi recogniton system, described later in Chapter 5.

4.6 PlantCLEF 2019

Compared to previous PlantCLEF challenges [60, 61, 63], which contained mainly species living in Europe and North America, the 2019 task is focused on the recognition of species from "data deficient regions" - mainly the Guiana shield and the Amazon rain forest. The proposed approach is based on CMP's winning submission to ExpertLifeCLEF 2018, described in Section 4.4. Checkpoints of our models from ExpertLifeCLEF 2018 have been shared with other participants of PlantCLEF 2019 in order to provide a good starting point to all participants.

4.6.1 Dataset

The PlantCLEF 2019 training set covers 10 000 species and consists of:

- PlantCLEF 2019 EoL: 72 260 images covering 4 197 classes from the Encyclopedia of Life [2].
- PlantCLEF 2019 Google: 68 254 images covering 6 262 classes automatically retrieved by web search engines.
- PlantCLEF 2019 Bing: 307 557 images covering 8 666 classes automatically retrieved by web search engines.

The average number of images per species decreased dramatically from ExpertLifeCLEF 2018. One fifth of species contains less than 10 images and some of them contains only 1 image. Examples from the training set and test set are displayed in Figure 4.19.

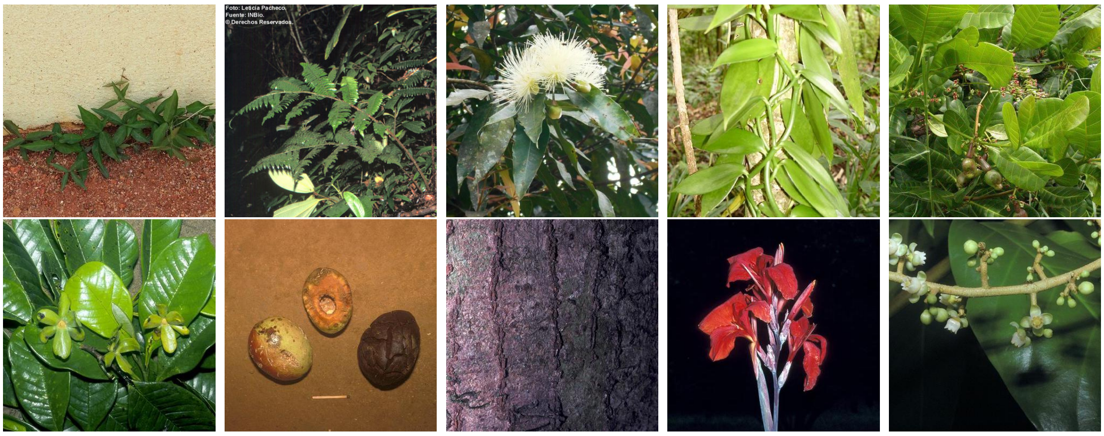


Figure 4.19: Randomly selected images from the PlantCLEF 2019 training set (top) and test set(bottom).

The challenge test set contained 2 974 images of covered 742 plant observations. Human experts were tested on its subset counting 117 plant observations.

4.6.2 Methodology

Cleaning and Extending the Training Dataset

A brief manual inspection showed that the provided training set contains noisy samples - wrongly labeled images, including images of non-flora objects. Examples of noisy samples are in Figure 4.20. We therefore decided to detect non-flora images by a pre-trained Darknet53 448x448 [154] classifier. Out of 428 702 images from the official training set, we removed 6 181 images detected as non-flora. As many species only had one or two images in the training set, the removal of untrusted images from the training data left approximately 2 000 classes without training samples. We had to gather additional training images to fill that gap. We created a new training set[16] including external training data downloaded from the Global Biodiversity Information Facility (GBIF) [4], described in Table 4.19. Changes in the dataset statistics are visualized in Figure 4.21.

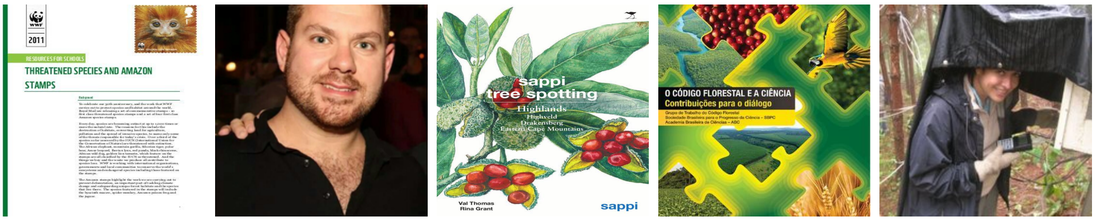


Figure 4.20: Randomly selected noisy images from the PlantCLEF 2019 training set.

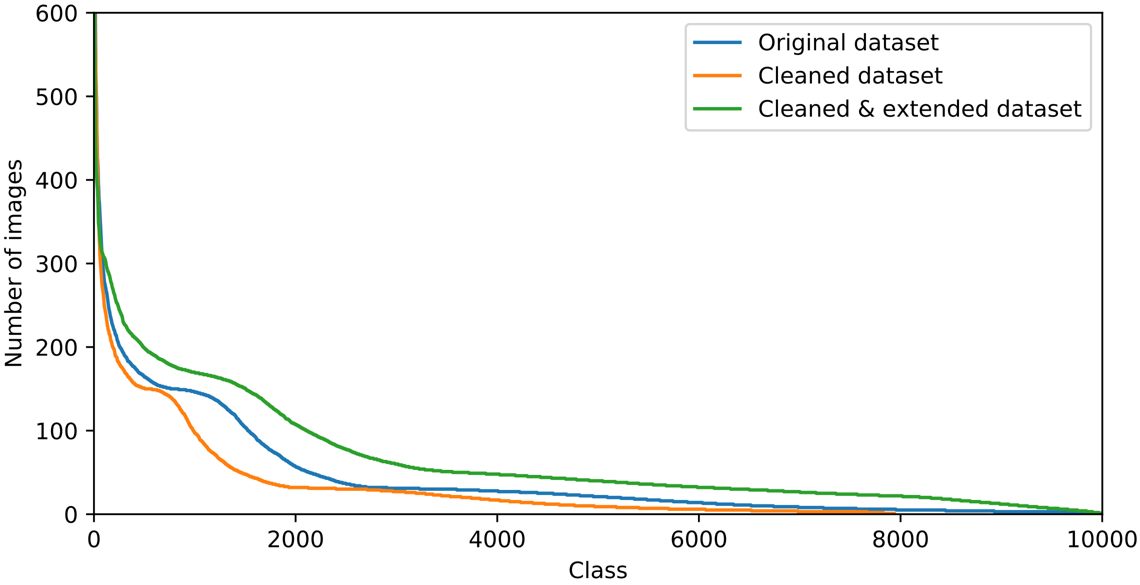


Figure 4.21: Numbers of training images per class in the original dataset (blue), cleaned dataset (orange) and cleaned and extended (green), sorted for each dataset separately.

---

[16] For full reproducibility, a list of removed samples as well as an archive with additional training images are shared at `http://cmp.felk.cvut.cz/~sulcmila/LifeCLEF2019/`. Last accessed 2nd Apr 2020.

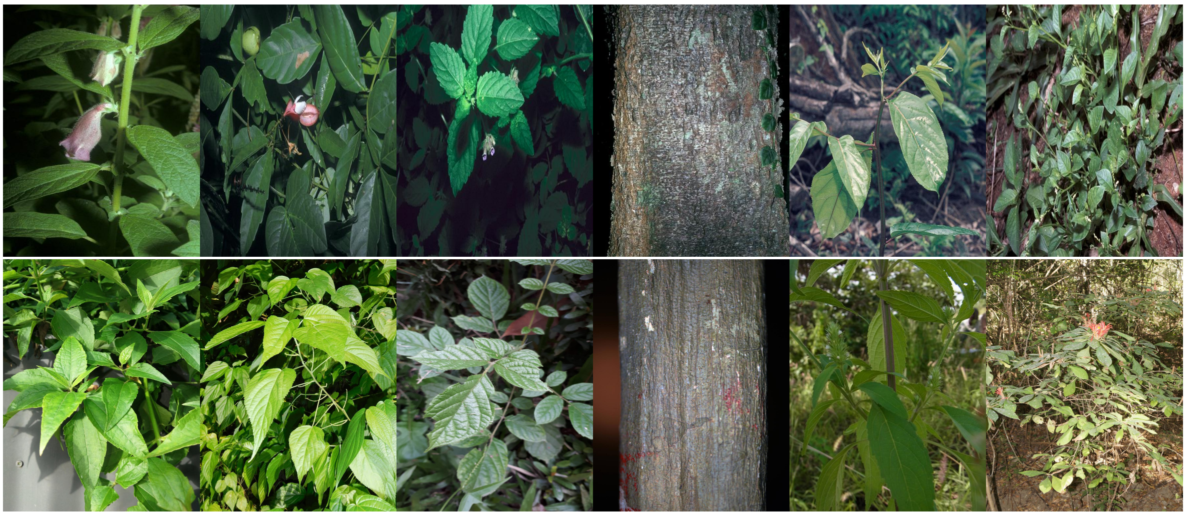


Figure 4.22: Six nearest couples of test set images (top) and GBIF images (bottom).

Table 4.19: Training data (after cleaning and extending the provided training set) used in the experiments.

Data Source	Classes	Non EoL classes	Number of Images
EoL	4 197	0	58 548
Noisy Google	6 262	3 800	64 863
Noisy Bing	8 666	5 069	305 291
GBIF (additional)	9 402	5 734	238 009
All	9 998	5 801	666 711

To make sure that none of the additional training images (or its resized or cropped versions) downloaded from GBIF appear in the test set, we used the image retrieval pipeline of Radenovic et al. [153] with VGG-16 and whitening. The nearest neighbours of test images among the downloaded images are vizualized in Figure 4.22.

Convolutional Neural Networks

The proposed system is based on two CNN architectures – Inception ResNet v2 and Inception v4 [186]. The TensorFlow-Slim API was used to adjust and fine-tune the networks from the publicly available[17] ExpertLifeCLEF 2018 winning checkpoints.

All networks in our experiments shared the optimizer settings enumerated in Table 4.20. The networks and their input resolutions are listed in Table 4.21.

---

[17] http://cmp.felk.cvut.cz/~sulcmila/LifeCLEF2018/ Last accessed 2nd Apr 2020.

Table 4.20: Optimizer hyper-parameters, common to all networks in the experiments.

Parameter	Value
Batch size	32
Optimizer	RMSProp
RMSProp momentum	0.9
RMSProp decay	0.9
Initial learning rate	0.0075
Learning rate decay type	Exponential (stairs)
Learning rate decay factor	0.975
Moving average (Polyak [148]) decay:	0.999

The following image pre-processing techniques were used for training:

- Random image crop with aspect ratio range $(0.75, 1.33)$ and content at least 80% of origin image.
- Random left-right flip.
- Brightness and saturation distortion.

Table 4.21: Network input resolutions.

#	Net architecture	Input Resolution
1	Inception v4	299×299
2	Inception v4 (second)	299×299
3	Inception v4	598×598
4	Inception ResNet v2	299×299
5	Inception ResNet v2 (second)	299×299

Test-time Data Augmentation

At test-time, 3 predictions per image are generated by using 3 crops:

- 1x Full image,
- 1x Central crop covering 80% of the original image,
- 1x Central crop covering 60% of the original image.

In some of our challenge submissions described later, the mirrored versions of all three crops were also evaluated.

Adjusting Class Priors at Test-time

The training set data distribution is highly unbalanced and we can not guarantee that the test images were drawn from the same distribution: as described in Section 4.6.1, the training set comes from different sources, where the class frequencies may not correspond with the test-time priors.

Following the notation from Section 3.3.1, the predictions $p(k|\mathbf{x})$ of a network trained on a dataset with class prior probabilities $p_Y(k)$ should be corrected in case of evaluation on a test set with different class priors $p^e_Y(k)$. From Equation 3.20 (Section 3.3.1) we know that

$$p^e(k|\mathbf{x}) \propto p(k|\mathbf{x})\frac{p^e_Y(k)}{p_Y(k)}.$$

Since the test-time priors $p^e_Y(k)$ are unknown, we use three different estimates of adjusting the predictions:

UNIFORM: As the simplest option, we adjust the test predictions by assuming a uniform prior for all classes.

MLE: As the second option, we compute a Maximum Likelihood Estimate of the test time prior $p^e_Y(k)$ using the EM algorithm of Saerens et al. [161] described in Section 3.3.2. Let us recall the two steps from Equations 3.23 and 3.24:

$$\begin{aligned} p^{(s)}(k|\mathbf{x}) &= \frac{p(k|\mathbf{x})\frac{\hat{p}^{(s)}_k}{p_Y(k)}}{\sum\limits_{j=1}^{K} p(j|\mathbf{x})\frac{\hat{p}^{(s)}_j}{p_Y(j)}} \\ \hat{p}^{(s+1)}_k &= \frac{1}{N^e}\sum_{\mathbf{x}\in\mathcal{E}} p^{(s)}(k|\mathbf{x}) \end{aligned}$$

MAP: As the third option, we use the Maximum a Posteriori estimate proposed in Section 3.3.3. Recall the objective from Equation 3.30:

$$\begin{aligned} \hat{\mathbf{p}}^{\mathrm{MAP}} &= \arg\max_{\mathbf{p}} p(\mathbf{p}|\mathcal{E}) \\ &= \arg\max_{\mathbf{p}} p(\mathbf{p})\prod_{\mathbf{x}\in\mathcal{E}} p(\mathbf{x}|\mathbf{p}) \\ &= \arg\max_{\mathbf{p}} \left[\log p(\mathbf{p}) + \sum_{\mathbf{x}\in\mathcal{E}} \log p(\mathbf{x}|\mathbf{p})\right] \\ \text{s.t. } &\sum_{k=1}^{K} p_k = 1;\ \forall k : p_k \geq 0 \end{aligned}$$

We model the prior knowledge about the categorical distribution by the symmetric Dirichlet distribution:

$$p(\mathbf{p}) = \frac{1}{B(\alpha)}\prod_{k=1}^{K} p_k^{\alpha-1} \tag{4.9}$$

where the normalization factor for the symmetric case is $B(\alpha) = \frac{\Gamma(\alpha)^K}{\Gamma(\alpha K)}$. We use $\alpha = 3$.

Table 4.22: Description of our (corrected, post-challenge) submissions.

Run description			Test accuracy (%)			
Name	Test-time augm.	Prior est.	Top1	Top1 Exp.	Top5 All	Top5 Exp.
CMP Run 2	3×scale	(none)	**31.9**	40.2	46.8	58.1
CMP Run 3	3×scale	uniform	30.7	40.2	45.1	57.3
CMP Run 4	3×scale	MAP	31.1	40.2	45.4	53.8
CMP Run 5	3×scale	MLE	24.4	32.5	35.6	41.0
CMP Run 6	3×scale + mirrors	(none)	**31.9**	**41.0**	**47.0**	**58.1**
CMP Run 7	3×scale + mirrors	uniform	30.1	40.2	45.3	57.3
CMP Run 4*	3×scale + mirrors	MAP	31.1	**41.0**	46.1	56.4
CMP Run 5*	3×scale + mirrors	MLE	24.7	31.6	36.0	41.9

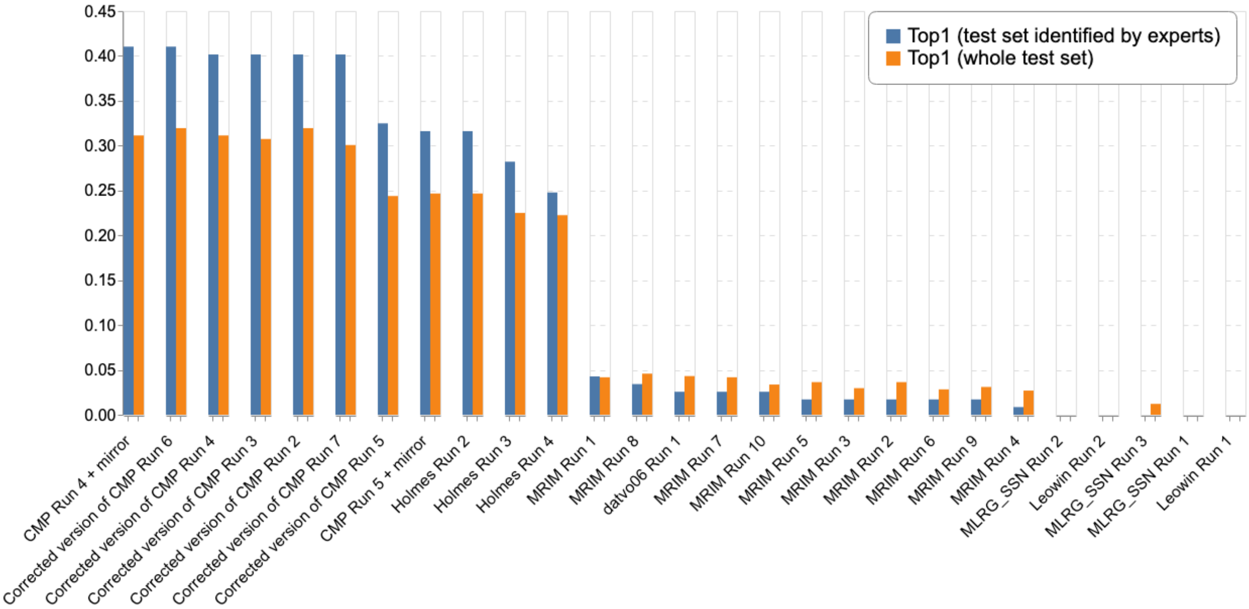


Figure 4.23: Comparison of automatic plant recognition methods on the PlantCLEF 2019 test set. (Note: The plot displays our post-challenge submissions).

4.6.3 Results

Table 4.22 describes eight final runs used for the evaluation. An ensemble of all five networks from Section 4.6.2 was used in all runs and predictions were averaged over all networks and all test image augmentations from Section 4.6.2.

The evaluation results are shown in Figures 4.23 and 4.24. From the class prior estimation methods, MAP estimation with the Dirichlet hyperprior achieves noticeably better results than the MLE. This is in accordance with the results presented later in Chapter 6, where adding the hyperprior brings noticeable improvement over the MLE estimation, which may have a tendency to overfit. The best results were achieved when not adjusting the predictions to a new prior. The weaker performance of the prior estimation meth-

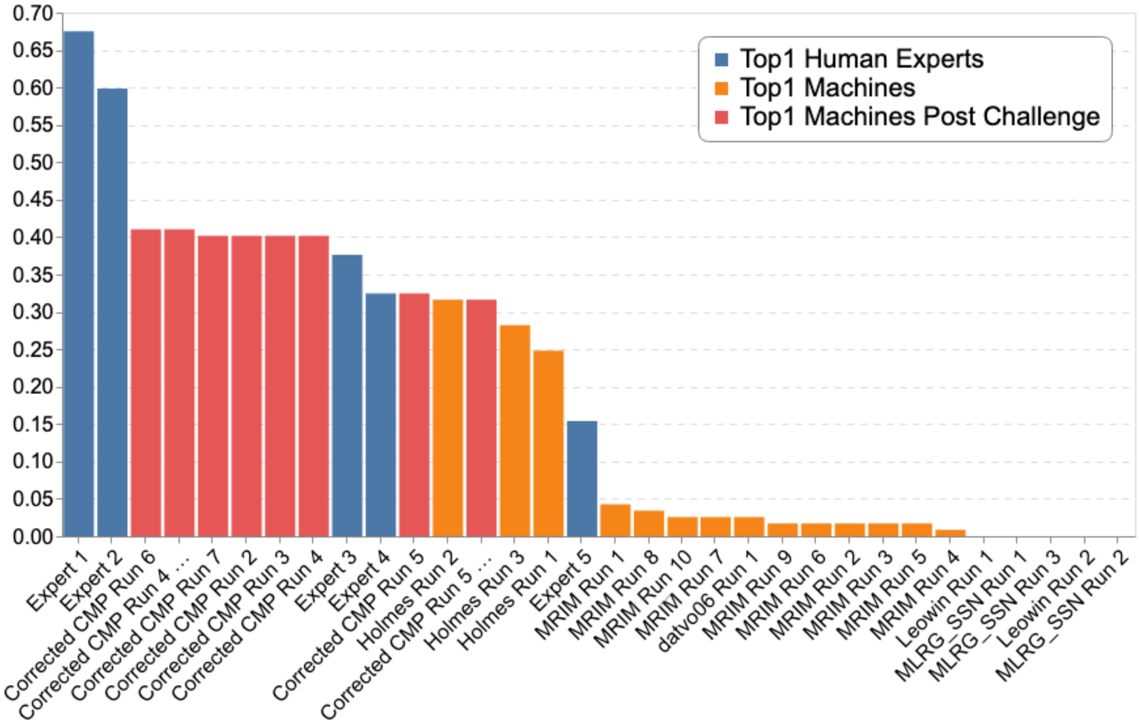


Figure 4.24: Comparison of automatic plant recognition methods against human experts. The results of our method are shown in red as "Post Challenge" (our results submitted at the challenge deadline, shown in orange, were wrongly exported).

ods may be related to the small number of training examples per class – insufficient to train the CNN classifiers well enough – and the relatively small size of the test set – only 742 observations while considering 10 000 species. Note that the results from Table 4.22 are the official post-challenge evaluation not included in the challenge leaderboard, as our predictions were wrongly exported into the challenge run-files.

4.6.4 Discussion

The proposed system achieves the best accuracy on the PlantCLEF 2019 test set - 31.9% on the full set and 41.0% on the test subset for plant identification experts. The results show that even for "data-deficient" plant species, automatic image recognition systems achieve human expert accuracy in visual recognition of plants: The proposed method performed better than 3 of the 5 participating experts in plant recognition.

4.7 Summary of the Challenge Results

The fine-grained species recognition challenges presented in this Chapter provided valuable large scale datasets, which – in addition to presenting difficult fine-grained recognition tasks – reveal additional problems such as learning with noisy labels and highly unbalanced training data, change of categorical priors between the training and test data, etc. Moreover, the competitions provide benchmarks of the best performing machine learning and computer vision algorithms.

The best results in all large scale recognition challenges presented in this chapter were achieved with deep Convolutional Neural Networks, including our winning submissions to the ExpertLifeCLEF 2018 plant identification challenge in Section 4.4, the FGVCx Fungi Classification Challenge and FGVCx Flower Classification Challenge in Section 4.5 and the best results on the PlantCLEF 2019 test set achieved by our post-challenge submission in Section 4.6. In all challenges, the top-performing submission were based on an ensemble of several CNN models, consistently achieving better results then single-model submissions. Our results suggest that the Polyak averaging [148] technique, where running averages of the trained variables are used instead of the values from the last training step, improves the recognition accuracy.

The problem of different categorical priors in the training and test data of the PlantCLEF challenges has motivated us to study the problem as presented in Section 3.3. The methods for adjusting CNN predictions to new categorical priors are experimented in more detail in Chapter 6.

Winning the FGVCx Fungi Classification Challenge started our communication with the mycologists from the Danish Mycological Society, with whom we integrated our classification models into a citizen science project for collection of fungi observations, as described in Chapter 5.

CHAPTER 5

Automatic Fungi Recognition as a Tool for Citizen-Science

This chapter presents a computer vision system for recognition of fungi “in the wild”, based on our winning submission to Kaggle competition organized with the Fine-Grained Categorization Workshop at CVPR 2018 from Section 4.5.5, and the application of this system to assist a citizen-science community and help mycologists increase the involvement of citizens in data collection.

Existing applications for image-based mushroom recognition are reviewed in Section 5.1.1. To the best of our knowledge, our system recognizes the largest number of species, and it is the first image-based fungi recognition system to assist citizen-scientists and mycologists in identification and collection of observations.

From the computer vision perspective, the application of the system to citizen-science data collection creates a valuable continuous stream of labeled examples for a challenging fine-grained visual classification task. The increasing amount of labeled data will allow to improve the classification baselines in the future and to study other interesting problems, such as fungi phenotyping, location-based estimation of categorical prior, etc. By linking the system to an existing mycological platform involving validation by the community, as is the case in the Atlas of Danish Fungi [42, 55, 80], a supervised machine learning system with human in the loop is created.

5.1 Related Work

5.1.1 Fungi Recognition

Several mobile applications for fungi identification include a computer vision classification system. Only few have positive user reviews on the identification results. Examples of apps with positive user reviews are:

- Mushroom Identificator[1] with 1M+ downloads and a review score of 4.0/5, recognizing more than 900 mushroom species,

---

[1] https://play.google.com/store/apps/details?id=com.pingou.champignouf Last accessed 2nd Apr 2020.

- Mushrooms App[2] with 0.5M+ downloads and a review score of 4.4/5, recognizing 210 mushroom species.

De Vooren et al. [195] published an image analysis tool for mushroom cultivars identification in 1992, analyzing morphological characters like length, width and other shape descriptors.

Computer vision may be used for classification of microscopy images of fungal spores. Tahir et al. [189] and Zielinski et al. [212] introduce datasets of microscopy images of fungal infections and propose methods to speed up medical diagnosis, allowing to avoid additional expensive biochemical tests.

5.1.2 Crowd-based Image Collection and Identification

The **Global Biodiversity Information Facility (GBIF)** [4] is the largest index of biodiversity data in the world. GBIF is organized as a network involving 58 participating countries and 38 organisations (mainly international) publishing more than 45 000 biodiversity datasets under open source licenses. The index contains more than 1.3 billion species occurrence records of which more than 47 million include images. With the recent advances in the use of machine vision in biodiversity related technology, GBIF intends to facilitate collaborations in this field, promote responsible data use and good citation practices. GBIF has the potential to play an active role in preparing training datasets and make them accessible under open source licenses [158].

iNaturalist [5] is a pioneering crowd-based platform allowing citizens and experts to upload and categorize observations of the world fauna, flora and fungi. All annotated data are directly uploaded to GBIF once verified by three independent users. iNaturalist covers more than 238 000 species through almost 28 million observations.

Wild Me is a non-profit organization that aims to combat extinction with citizen-science and artificial intelligence. Their projects using computer vision [141] to boost detection and identification include: **Flukebook**, a collaboration system to collect citizen observations of dolphins and whales and to identify individuals, and **GiraffeSpotter**, a photo-identification database of giraffe encounters.

The **Atlas of Danish Fungi (SvampeAtlas)** [42, 55, 80] involves more than 1000 volunteers who have contributed approximately 500 000 quality-checked observations of fungi. More than 270 000 old fungal records were imported into the project database which now contains more than 800 000 quality-checked fungal records. The project has resulted in a greatly improved knowledge of Denmark's fungi. More than 180 basidiomycetes[3] have been added to the list of known Danish species, and several species that were considered extinct have been re-discovered. At the same time, a number of search and assistance functions have been developed that present common knowledge about the individual species of fungi, which makes it much easier to include knowledge of endangered species in the nature management and decision making.

All validated records are published to the Global Biodiversity Information Facility [4] on a weekly basis. Since 2017, the Atlas of Danish Fungi has had interactive validation

---

[2] `https://play.google.com/store/apps/details?id=bazinac.aplikacenahouby` Last accessed 2nd Apr 2020.

[3] Microscopic spore-producing structure found on the hymenophore of fruiting bodies.

of fungal records. When a user submits a record, a probability score is calculated for the accuracy of the identification. This score ranges from 1 to 100. The calculation includes:

1. The rarity of the species (# approved records).
2. The geographical distribution of the species.
3. Phenology of the species (e.g. many mycorrhizal fungi have a low probability score in spring).
4. User's previous approved identifications of the same species.
5. Nr. of species within the morphological group the user has correctly identified in the past.
6. Confidence indicated by the user: Certain: 100%, Probable: 50%, Possible: 10%.

Subsequently, other users may agree on the identification, increasing the identification score in accordance with the principles 4–6, or propose alternative identifications. The identification with the highest score is highlighted, alternative identifications and their scores are also visible to logged-in users. In the search results, the probability score is displayed in three general categories:

1. Approved (score above 80) with 3 stars.
2. Likely (score between 50 and 80) with 2 stars.
3. Suggestion (score below 50) with 1 star.

A group of taxonomic experts (validators) are monitoring data in the Atlas of Danish Fungi. These have the power to approve findings regardless of the score in the interactive validation. This can be relevant for discoveries of new species, for very rare species and for records of species where special experience or sequencing of genetic material (DNA) is required for a safe identification. Expert-validated findings are marked with a small microscope icon.

5.2 Online Fungi Classification Service

The recognition system is based on the dataset provided by the FGVCx Fungi Classification Challenge described in Section 4.5.1. The pipeline used for the FGVCx Fungi Classification challenge was described in Section 4.5.4. All six fine-tuned networks from our ensemble are publicly available[4]. The predictions of the CNNs were adjusted to new priors was discussed in Sections 3.3.1 and 4.5.4.

In order to provide a flexible and scalable image-based fungi identification service for the Atlas of Danish Fungi, we created a recognition server based on the open-source TensorFlow Serving [137] framework. The server currently uses one of our pretrained models from Section 4.5.4, the framework allows to deploy several models at the same time. No test-time augmentations are currently used in order to prevent server overload.

The pipeline is visualized in Figure 5.1: The web- and mobile apps query the recognition server via Representational State Transfer (REST) API. The server feeds the query image

---

[4] `https://github.com/sulc/fungi-recognition` Last accessed 2nd Apr 2020.

into the Convolutional Network and responds with the list of predicted species probabilities. The apps then display a shortlist of the most likely species for the query. The observation is also uploaded into the Atlas of Danish Fungi database. The user can manually inspect the proposed species and select the best result for annotation of the fungus observation. Screenshots of the web and mobile interfaces are shown in Figure 5.2 and Figure 5.3 respectively.

Observations uploaded into the Atlas of Danish Fungi database and the proposed species identifications are then verified by the community. Images with verified species labels will be used to further fine-tune the recognition system.

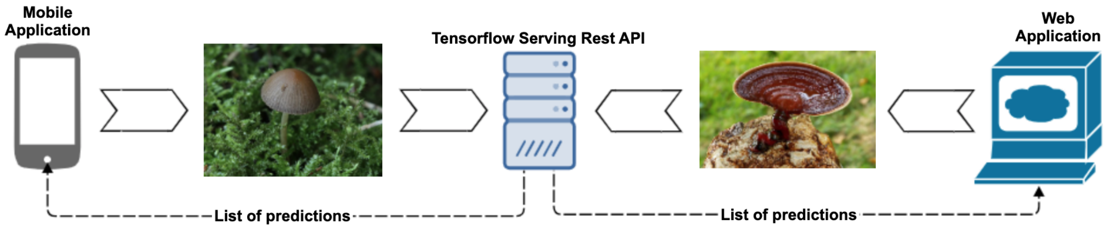
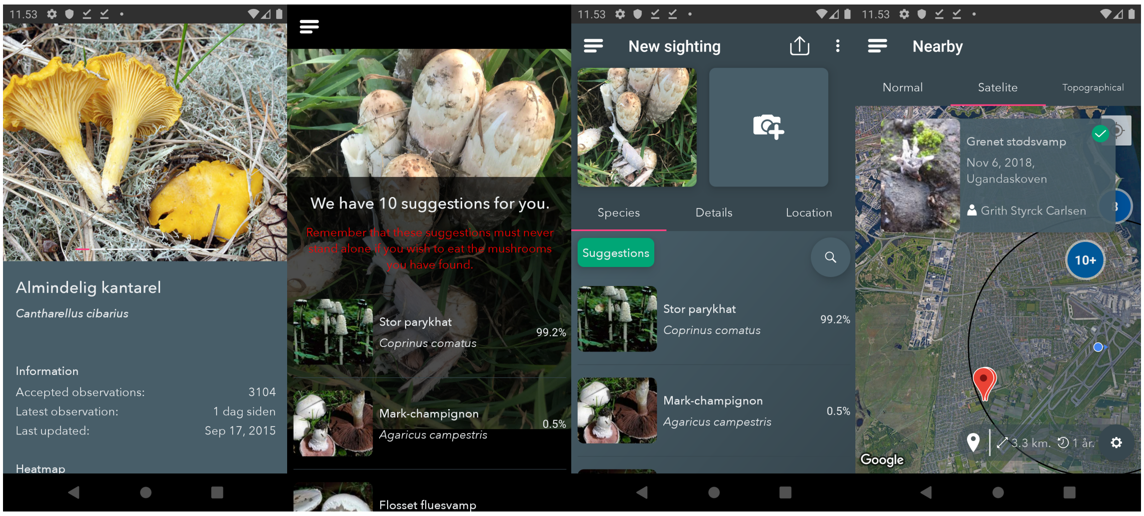


Figure 5.1: The fungi recognition serving pipeline.



Almindelig kantarel
Cantharellus cibarius

Information
Accepted observations: 3104
Latest observation: 1 dag siden
Last updated: Sep 17, 2015

Heatmap

We have 10 suggestions for you.

Remember that these suggestions must never stand alone if you wish to eat the mushrooms you have found.

Stor parykhat *Coprinus comatus* 99.2%

Mark-champignon *Agaricus campestris* 0.5%

Flosset fluesvamp

Stor parykhat *Coprinus comatus* 99.2%

Mark-champignon *Agaricus campestris* 0.5%

Grenet stødsvamp
Nov 6, 2018,
Ugandaskoven
Grith Styrck Carlsen

3.3 km. 1 år.

(a) (b) (c) (d)

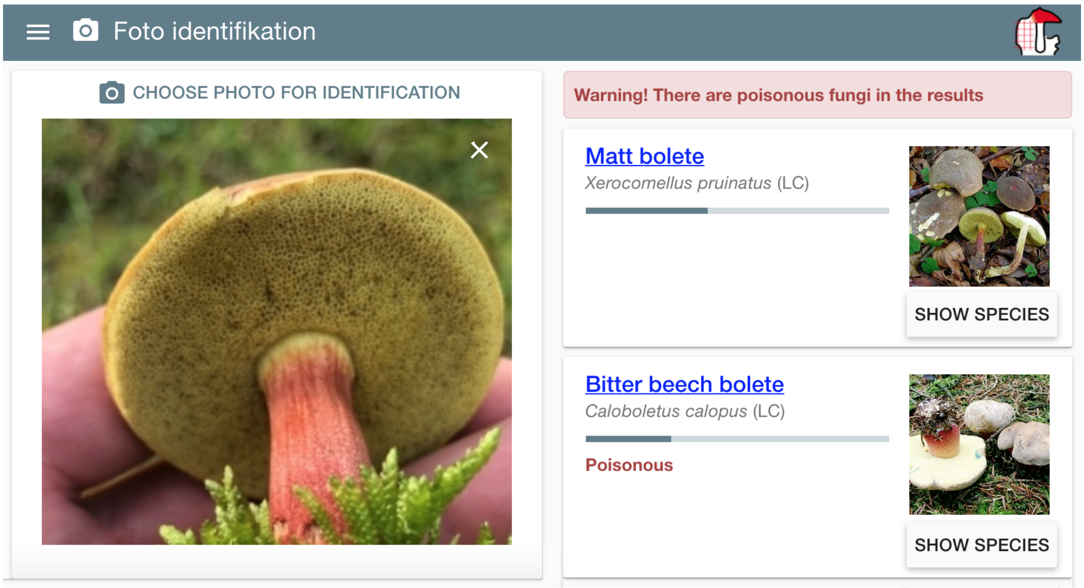
Figure 5.2: Screenshots from the Atlas of Danish Fungi mobile application showing: (a) A detailed description of selected species, (b,c) Image based recognition suggesting species for a query image, (d) Map with nearby observations.

5.3 Results

The experts behind the Atlas of Danish Fungi have been highly impressed by the performance of the system[5]. From the first 5760 records that have been submitted for automatic

---

[5] Personal communication with the Atlas of Danish Fungi.



Warning! There are poisonous fungi in the results

Matt bolete
Xerocomellus pruinatus (LC)

Bitter beech bolete
Caloboletus calopus (LC)
Poisonous

Figure 5.3: Screenshot from the web-based recognition app (https://svampe.databasen.org/imagevision Last accessed 2nd Apr 2020).

recognition, only 904 (16 %) were not approved by community- or expert validation. This is a far better performance than most non-expert users in the system. Almost two thirds (64 %) of the approved species identifications were based on the highest ranking AI suggesting species ID, while another 7 % were based on the second highest ranking AI suggested species ID and another 6 % were based and top 3-5 suggestions.

It has not been possible to collect data on identification attempts where no useful match was returned from the AI, and the user therefore picked a taxon name not in the top 10 AI results. However, users generally stated that this rarely happened. So far the system has been tested by 652 users, each submitting between one and 526 records. For users submitting more than ten records the accuracy in terms of correct identifications guided by the system varied from 17% to 100%, pointing to quite considerable differences in how well different users have been able to identify the correct species using the system. Hence, the tool is not fully reliable, but helps the non-expert users to gain better identification skills. The accuracy was variable among the fungal morphogroups defined in the fungal atlas, varying from 24 % to 100 % for groups with more than 10 records. The accuracy was tightly correlated with the obtained morphogroup user score based on the algorithms deployed in the Atlas of Danish Fungi to support community validation.

The operators of the Atlas of Danish Fungi received positive feedback from several users about the new AI-identification feature.

The observation statistics in Table 5.1 and Figure 5.4 show a significant increase of submitted observations after releasing the automatic fungi recognition service in October 2019.

Table 5.1: Observation statistics from the Atlas of Danish Fungi: Number of sightings and images uploaded by the users before and after introducing our automatic species identification service.

	Sightings	Images
Before (Oct.-Dec.2018)	17 025	10 779
After (Oct.-Dec.2019)	30 167	20 666

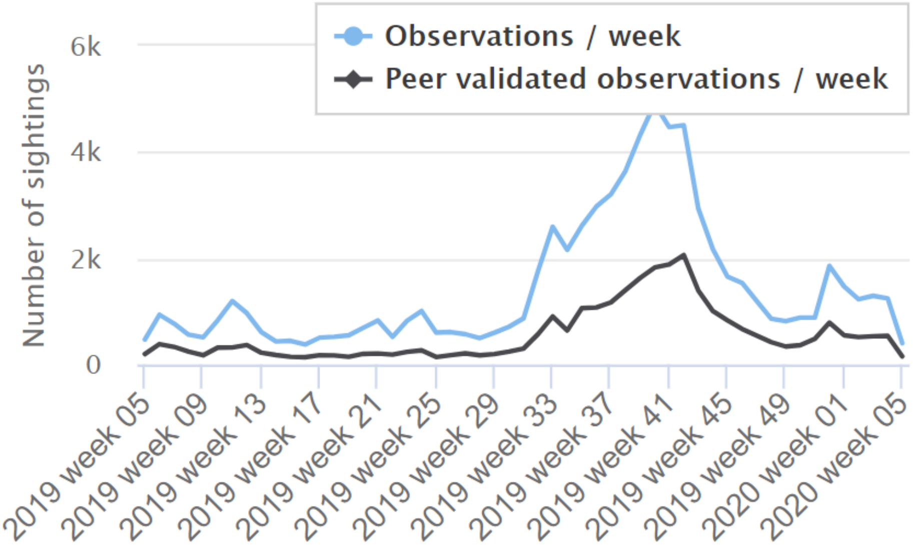


Figure 5.4: Observation statistics from the Atlas of Danish Fungi: Number of submitted and validated observations per week.

5.4 Discussion

This chapter described the application of a fungi recognition system, which was based on our submission to a computer vision Kaggle challenge, that aims at helping citizen-scientists to identify species of observed specimen and motivating their contributions to a citizen-science project.

Integration of the image recognition system into the Atlas of Danish Fungi makes community-based fungi observation identification easier: from the first 592 approved annotations, 89% were based on the top-2 predictions of our model.

Cross science efforts such as the collaboration described here can develop tools for citizen-scientists that improve their skills and the quality of the data they generate. Along with data generated by DNA sequencing this may help lowering the taxonomic bias in the biodiversity information data available in the future.

The server-based inference allows computation of accurate predictions with good response time, and it motivates users to upload images. On-device mobile inference would allow real-time recognition in areas with limited access to mobile data, however, decreasing the model size and complexity would be necessary. Possible directions for future work include applying efficient architectures [81, 162, 190], weight pruning and quantization [72, 85, 204].

A future deeper integration into mycological information systems may allow on-line learning of the classifier. Extending the collaboration with more mycological institutes or information systems may help to improve the system even further, as it would learn from all available data. As species distribution differs based on geographical locations and local environment, estimating the priors for different locations may be used in future work to adjust the predictions for observations with GPS information. The recent work of Mac Aodha et al. [125] may be relevant for modelling such spatio-temporal prior.

CHAPTER 6

Evaluation of Adjusting Predictions to New Class Prior Probabilities

This chapter returns to the problem of classification on a test set with different class priors than the training set, as introduced in Section 3.3. While some of the proposed methods for estimating the new priors and adjusting the predictions have been used in the computer vision challenges described in Chapter 4, this Chapter aims at a more rigorous evaluation.

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

CIFAR-100 [102], which was introduced in Section 3.2.1, 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 Chapter.

The **PlantCLEF 2017** [62] recognition challenge and dataset have been described in Section 4.2. The provided training images for 10 000 plant species consisted from an EoL[1] "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 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[2][3]. PlantCLEF presents an example of a real-world fine-grained classification task, where the number of available images per class is highly unbalanced.

The **FGVC iNaturalist 2018** large scale species classification competition and dataset have been described in Section 4.5.2. 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.

The **FGVCx Fungi 2018** species classification competition, focused only on fungi, and the related dataset have been described in Section 4.5.1. The dataset covers nearly

---

[1] downloaded from the Encyclopedia of Life [2]

[2] `http://imageclef.org/lifeclef/2017/plant` Last accessed 2nd Apr 2020.

[3] `http://imageclef.org/lifeclef/2016/plant` Last accessed 2nd Apr 2020.

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 [116] (also known as 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.

6.1 Adjusting Predictions When Test-time Priors Are Known

To experiment with known test-time prior probabilities $p_Y^e(k)$, we use the training and validation sets from the FGVC iNaturalist[4] and the FGVCx Fungi[5] 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 [186], was fine-tuned for each task. The predictions were corrected as defined by Eq. 3.20.

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. [116], the change of class prior probabilities is not taken into consideration. Similarly to [116] 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 6.1 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.

6.2 Estimation of New Priors From the Test Set

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 6.1 show that the top-1 accuracy increases by **3.4%** when estimating the test set priors using the EM algorithm [161]. 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).

---

[4] `https://sites.google.com/view/fgvc5/competitions/inaturalist` Last accessed 2nd Apr 2020.

[5] `https://sites.google.com/view/fgvc4/competitions/fgvcx/fungi` Last accessed 2nd Apr 2020.

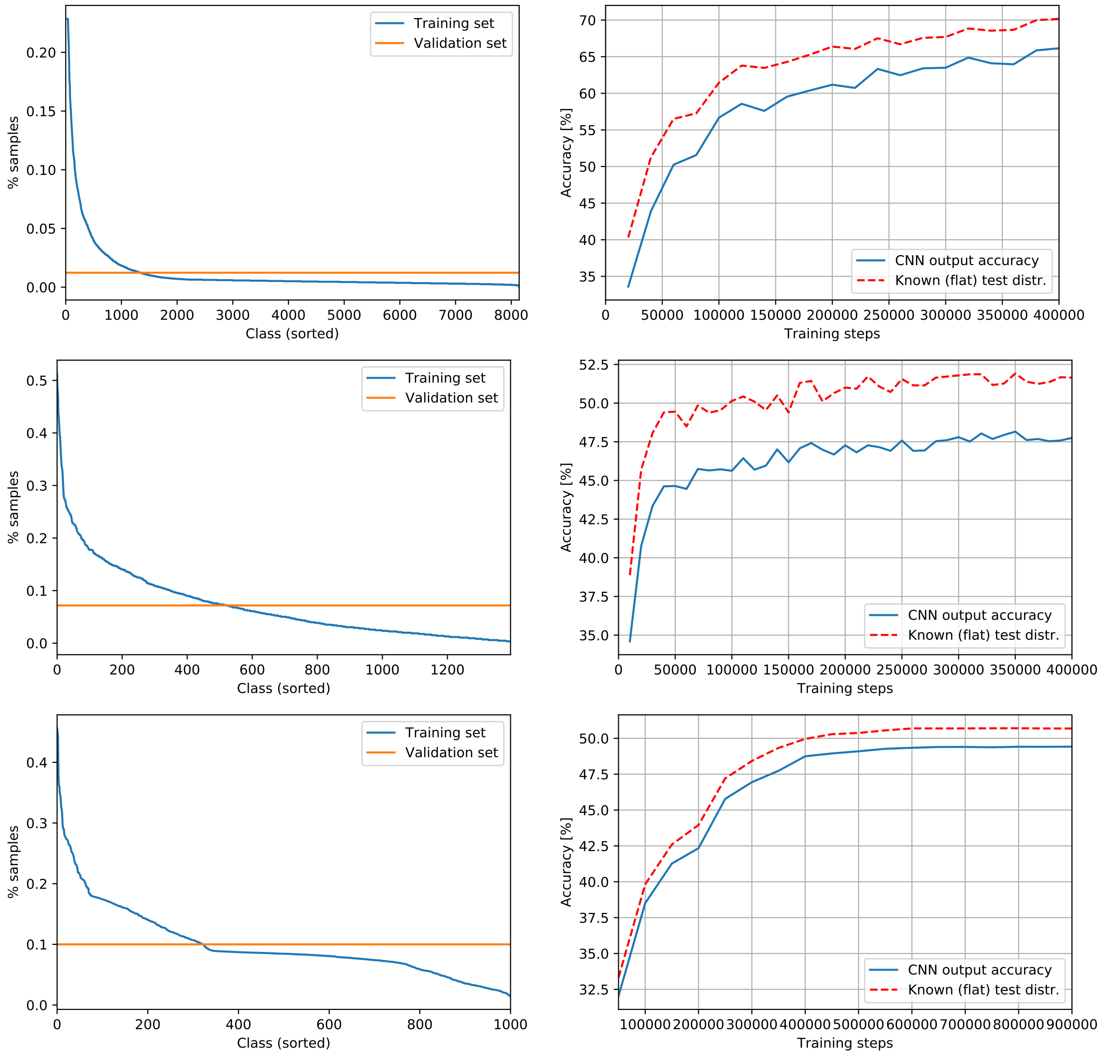
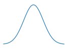
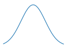
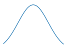
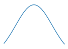
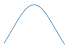
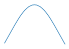
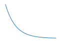
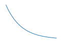
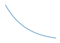
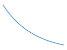
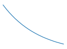
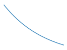


Figure 6.1: Training and validation set distributions (top) and accuracy before and after correcting predictions with the known/uniform val. set distribution (bottom) for FGVC iNaturalist 2018 (left), FGVCx Fungi 2018 (middle) and Webvision 2017 (right).

Table 6.1: Improvement in accuracy after applying the iterative test set prior estimation in the PlantCLEF 2017 plant identification challenge.

Model	Accuracy	Accuracy after EM	Acc. per observation, (our method after EM)	Acc. per observation, $p^e_Y(k)$ known
Inception V4	83.3%	86.7%	**90.8%**	93.7%
12 CNNs ensemble [107] (PlantCLEF2017 winner)	–	–	88.5%	–

Table 6.2: 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).

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 $p_Y^e(k)$	51.20	57.61	62.23	64.73	65.92	67.44	40.62	50.07	55.86	57.49	60.92	62.11

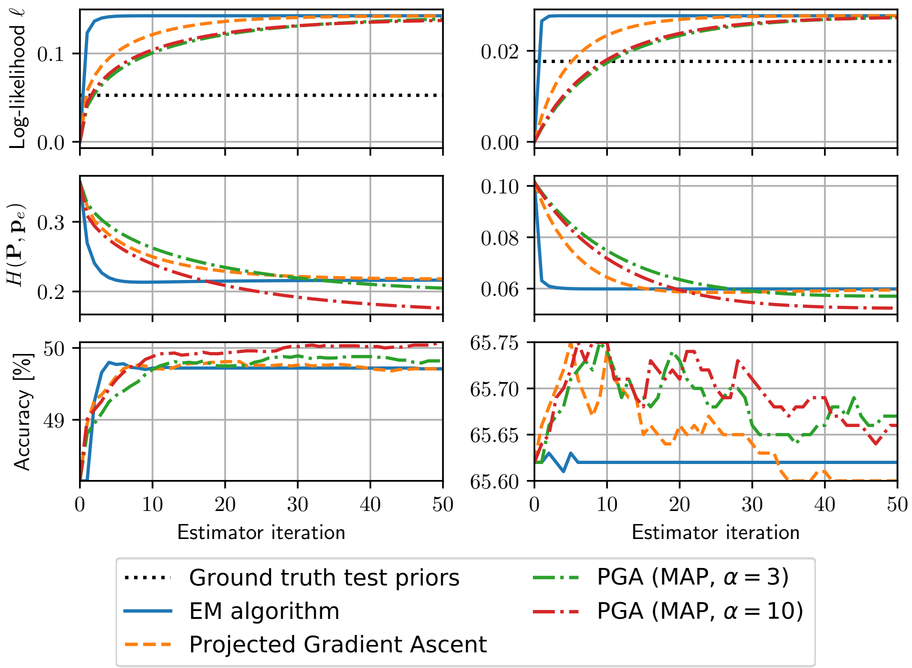


Figure 6.2: Iterative estimation of test-time priors on the full CIFAR-100 test set from CNNs trained on unbalanced CIFAR-100 subsets.

Networks trained on the selected subsets of CIFAR-100 from Section 3.2.1 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 6.2. 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 6.2, the likelihood of its estimate is lower than of the ML estimates. This demonstrates the importance of adding prior assumptions on the estimated class prior probabilities. The EM algorithm for ML estimation, however, converges noticeably faster.

Figure 6.3 summarizes the estimation of class priors on the fine-grained datasets Plant-

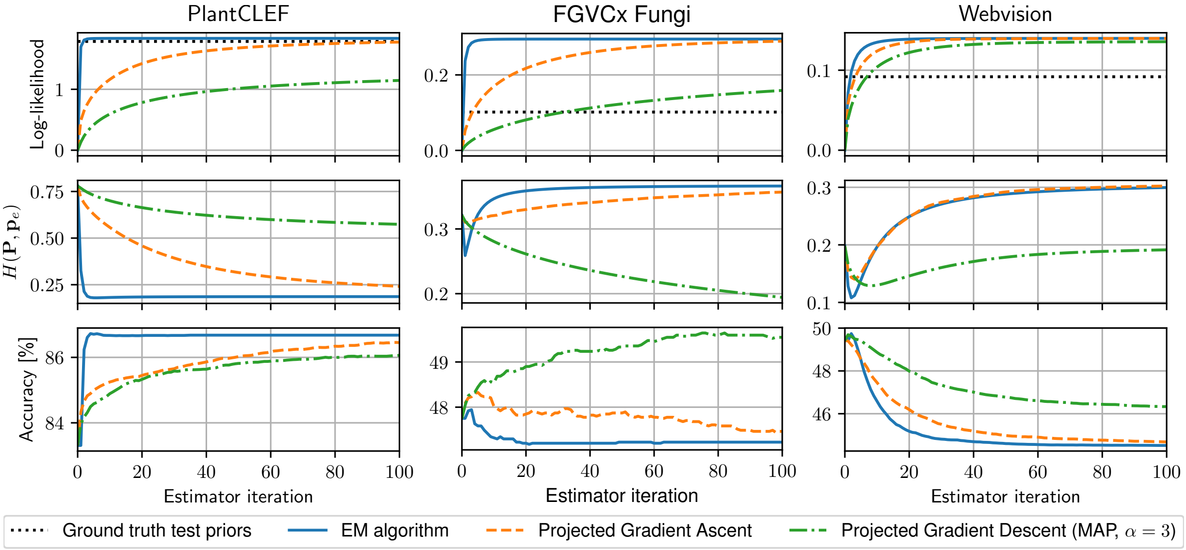


Figure 6.3: Iterative estimation of test-time priors on fine-grained datasets: PlantCLEF (Inception-v4), FGVCx Fungi (Inception-v4), and Webvision 1.0 (AlexNet). Top row: The log-likelihood surrogate ℓ. Middle row: Hellinger distance between the prior estimate and ground truth class frequencies. Bottom row: Accuracy.

CLEF, 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. [116] 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 MLE. 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.

6.2.1 Cross-validation of the Prior Estimate Likelihood

The experiments in Section 6.2 show that increasing the likelihood does not always lead to a more precise estimate. One possible reason may be over-fitting to the predictions on the test set (to a_{ik} in Equation 3.26). Let us "cross-validate" the likelihood on the test set: We will optimize the estimate only on a random half of the test set (likelihood-optimization set), and use the other half for likelihood-validation. Note that for this experiment, we use the projected gradient descent with a lower learning rate, in order to observe the changes in convergence in more detail.

Figure 6.4 shows, that even for the "unseen" half of the data (likelihood-validation set), the likelihood of the solution still increases, while the accuracy on both sets is decreasing. Therefore, this is not a case of over-fitting to the seen predictions, and the decreasing accuracy when maximizing the likelihood function remains an open problem.

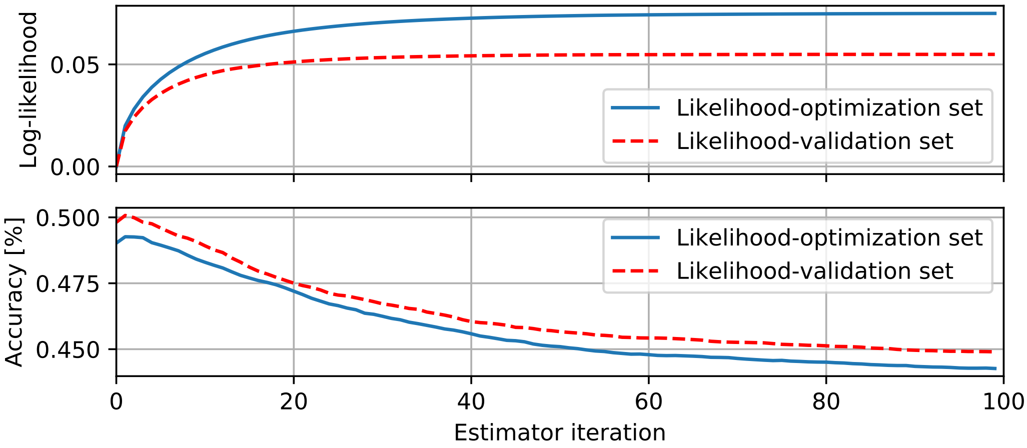


Figure 6.4: "Cross-validation" of the likelihood optimization on Webvision 1.0, using only half of the test set (likelihood-optimization set) to estimate the class priors, and observing the log-likelihood on the other half (likelihood-validation set).

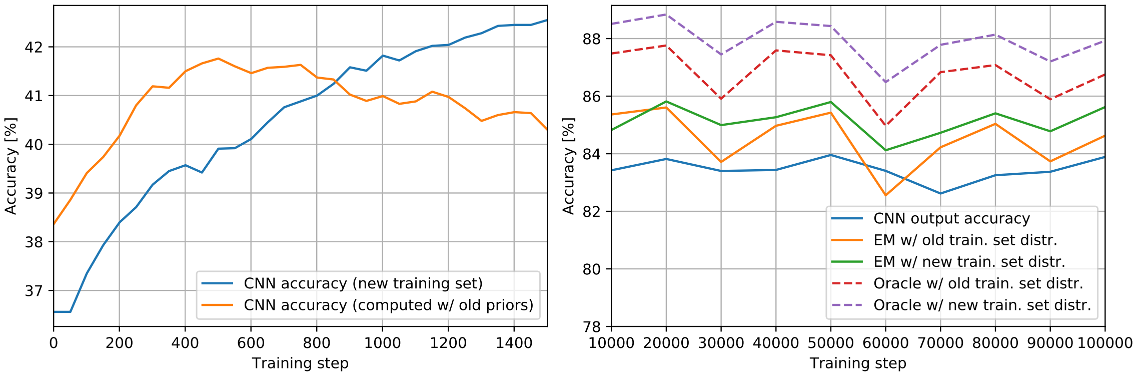


Figure 6.5: CNN pre-trained on unbalanced CIFAR-100 subset fine-tuned on the full CIFAR-100 training set (left). CNN pre-trained on *PlantCLEF-All* fine-tuned on *PlantCLEF-Trusted* (right).

6.3 Changing the Training Set Priors

How fast do the effective "learned" priors change when the training set changes during training? In this experiment, new samples are added into the training set. We take a network from Section 3.2.1 pre-trained on an unbalanced subset of CIFAR-100 and we fine-tune it on the full (balanced) CIFAR-100 training set. The predictions are evaluated on the complete (and balanced) test set. From the results in Figure 6.5 (left), it is clearly visible that using the old training set priors is still favorable for a few fine-tuning steps, but the effective priors of the CNN classifier seem to change fast.

The second experiment covers the other case: removing samples from the training set. On the PlantCLEF 2017 dataset, we used all training data (*PlantCLEF-All*) and then removed the major subset with noisy labels and fine-tuned only on the trusted data (*PlantCLEF-Trusted*). As visible in Figure 6.5 (right), in the second experiment the difference between the results with the old and new priors is significantly lower, but displays a similar case.

6.4 Adjusting Posterior Probabilities Online 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 affects the results on-line, i.e. when the priors are estimated from the already seen examples, see Figure 6.6. 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.

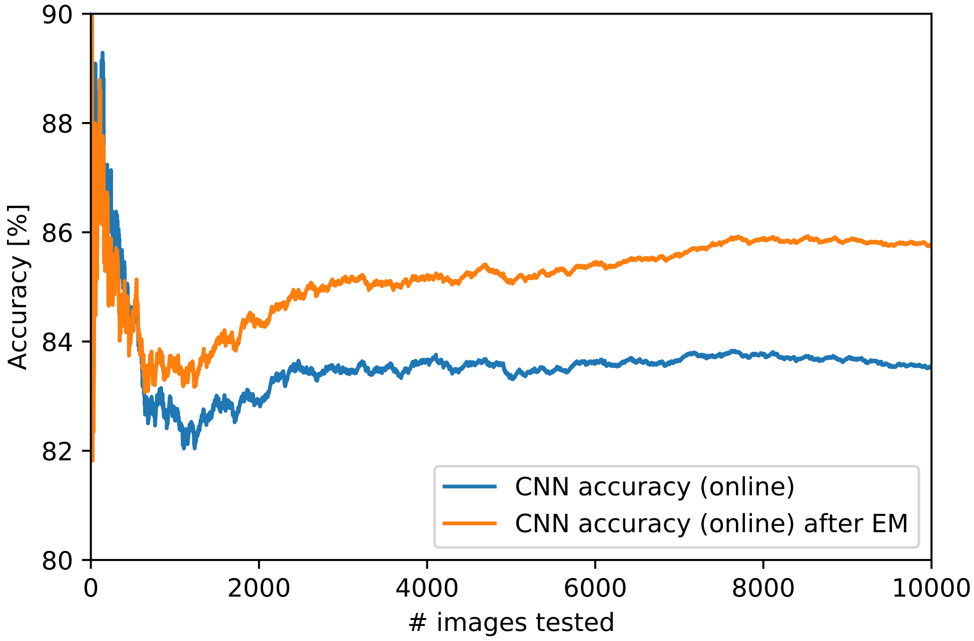


Figure 6.6: On-line test-prior estimation for PlantCLEF 2017.

6.5 Temperature Scaling

As discussed in Section 3.2.2, CNNs tend to provide over-confident prediction which can be calibrated by temperature scaling. Would such calibration improve the estimation of test set priors? Let us answer the question experimentally using the CNNs from Section 3.2.1 trained on the unbalanced subsets of the CIFAR-100 dataset.

The temperature scaling optimization should be performed on labeled samples from a development set, which was not used for training. In order to reuse the previously trained networks, we sample the development set (with the same distribution as the training set) from one half of the original CIFAR-100 test set, and use the other half (5000 images) as the test set for evaluation.

The reliability diagrams in Figure 6.7 show that on the development set used for temperature optimization, temperature scaling calibrates the prediction confidence well. The reliability diagrams on the test set in Figure 6.8 also show a noticeable improvement in the reliability of prediction confidences after temperature scaling, although the calibration error is slightly higher compared to the development set.

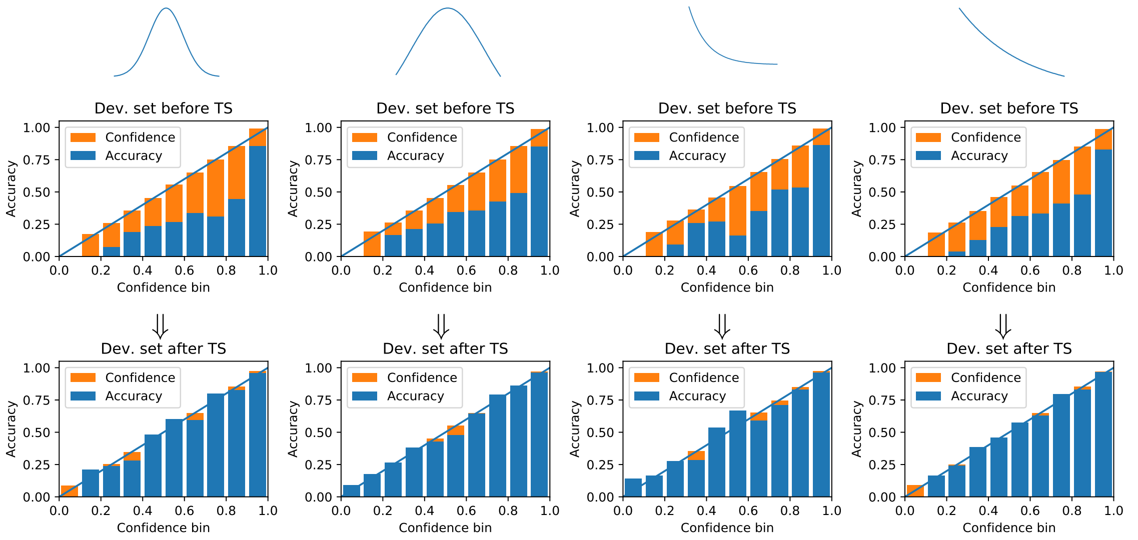


Figure 6.7: Reliability diagrams on the **development set** before (middle) and after (bottom) temperature scaling displayed for the 4 classifiers trained on different unbalanced (top) subsets of CIFAR-100, in the same order as in Figure 3.6.

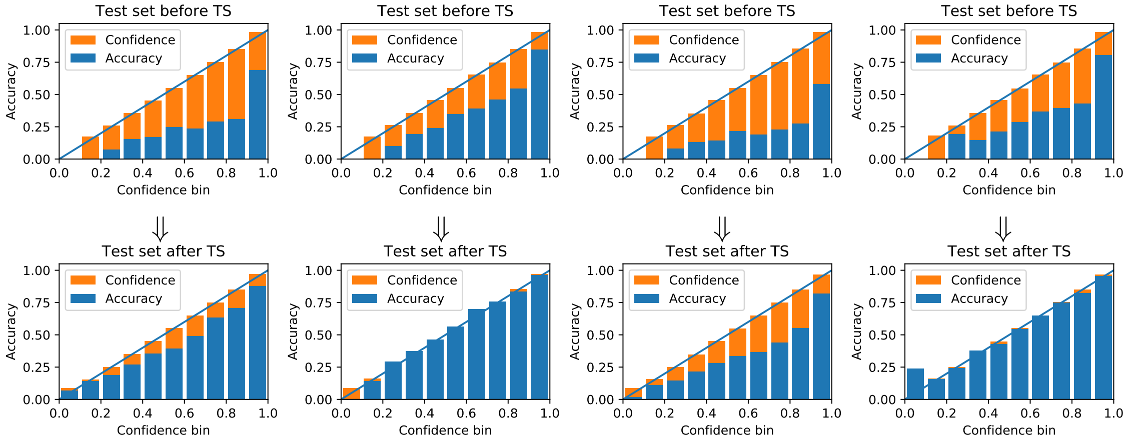
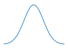
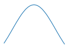
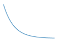
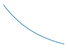


Figure 6.8: Reliability diagrams on the **test set** before (top) and after (bottom) temperature scaling, displayed for the 4 classifiers trained on different unbalanced subsets of CIFAR-100, in the same order as in Figure 3.6.

Results in Table 6.3 show how temperature scaling affects the results with MLE or MAP estimation of new categorical priors. Interestingly, in all experimented cases, better results are achieved without temperature scaling of the original predictions. Moreover, in the cases where the training distribution was very unbalanced (first and third row), adding temperature scaling strongly decreased the final recognition accuracy.

The results indicate, that while temperature scaling calibrates the reliability of the predictions as confidence scores, it impairs some statistical properties of the posterior estimate, making it less suitable for further processing in the prior estimation methods. Figure 6.9 shows the average of the predictions on the development set, similar to the initial validation in Section 3.2.1, before and after temperature scaling. The latter is slightly more prone to over-estimating the priors on the tail classes while under-estimating the most common classes, but still approximates the priors fairly well.

Table 6.3: Accuracy of CNN classifiers trained on unbalanced CIFAR-100 subsets (top) and evaluated on one half of the CIFAR-100 test set. We compare the accuracy of predictions directly adjusted by the MLE/MAP estimated class priors and the accuracy of predictions first calibrated by temperature scaling, followed by the MLE/MAP estimation.

Train. distribution				
Accuracy (%)	48.08	66.58	37.04	60.76
Acc. (%) after MLE	49.58	66.54	39.02	61.24
Acc. (%) after MAP, $\alpha = 3$	49.80	66.70	39.00	61.18
Optimal temparature T	2.26	2.11	2.05	2.30
Acc. (%) after temp. scaling and MLE	27.22	66.12	23.16	60.74
Acc. (%) after temp. scaling and MAP, $\alpha = 3$	38.40	66.58	33.66	61.42

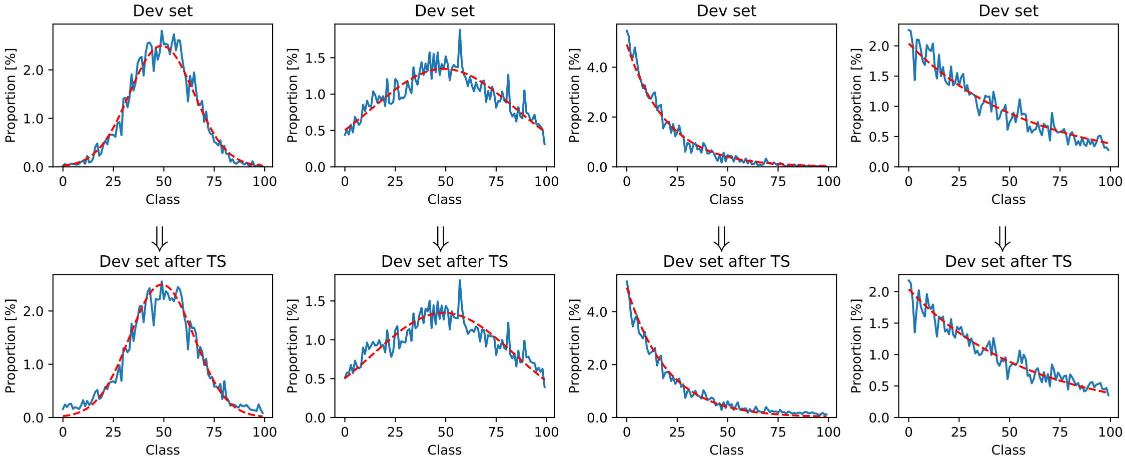


Figure 6.9: Comparison of class frequency and the averaged predictions over all images in the development set before (top) and after (bottom) temperature scaling.

6.6 Discussion

This chapter 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 [161] 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 prevents the class prior estimates from getting too close to zero. This improves the results and stability in most cases, including the FGVCx Fungi dataset, where it increased the accuracy by

1.8% while the ML estimate would lead to a decrease. It brings a slightly lower 2.8% increase in accuracy on the PlantCLEF dataset, where many classes are actually missing in the test set. The estimation of new priors did not help only on Webvision dataset - this may be related to the high amount ($\approx 34\%$) of outliers in the dataset. The analysis of the effect of noisy data on the estimation of new categorical priors is a topic for future work.

Experiments with calibrating the classifier confidence by temperature scaling suggest that while temperature scaling performed well in calibrating the reliability of the predictions as confidence scores, it impaired some statistical properties of the posterior estimate, making it unsuitable for further processing in the prior estimation methods.

CHAPTER 7

Conclusions

The thesis addressed the problem of fine-grained image classification, in particular plant and fungi species identification from images, ranging from canonical views in controlled conditions – recognition of leaf scans or photos of leaves on white background and cropped photos of tree bark – to unconstrained observations of plants and fungi "in the wild", where photos of arbitrary parts of the plant often appear with complex background and clutter in the scene. The tasks, i.e. the variants of the identification problem, are interesting instances of fine-grained classification because of the diverse appearance and complex structure of the organisms, high intra-class variability and small inter-class differences, and potentially a high number of classes (up to 10 000 in the LifeCLEF datasets).

The constrained tasks of leaf and bark classification in Chapter 2 were addressed with a texture-recognition approach. The proposed method, Fast Features Invariant to Rotation and Scale of Texture (*Ffirst*), achieved excellent results in bark and leaf classification: the recognition rates were above 99% on most leaf datasets, suggesting that texture is a highly discriminative feature for leaf recognition. *Ffirst* also achieved very competitive results on standard texture classification datasets, achieving above 99% accuracy on the Brodatz32, UIUCTex, UMD, CUReT and KTH-TIPS datasets. This almost perfect precision basically retires most of the standard texture classification datasets. The method is computationally efficient and fast: processing 200x200 px images takes about 0.05 seconds on a laptop without using a GPU. Comparing Ffirst, which only processes gray-scale images, to other state-of-the-art texture descriptors, we noticed a significant color bias on several standard texture recognition datasets and proposed improvements to global color descriptors.

We adopted a deep learning approach for the more complex "in the wild" species recognition. We tackled the problem of change in categorical priors, which is common to many species recognition datasets: the class distribution on the training set is often long-tailed and the proportion of individual classes in the training set and in the test set often differs. Chapter 3 interpreted the CNN classifiers trained by cross entropy minimization as estimators of posterior probabilities and experimentally validate some of their properties. For estimation of the new categorical priors, a Maximum Likelihood estimation approach is compared with a proposed Maximum a Posteriori method, adding a hyper-prior favouring dense prior distributions. The results presented in Chapter 6 show that adding such hyper-prior increases the reliability of the estimate and increases the classification accuracy in several fine-grained classification tasks. Our experiments suggest that calibration of over-confident classifiers by temperature scaling impairs some statistical properties of the posterior estimate, decreasing the performance of the prior estimation methods. Calibration of CNN predictions before estimaion of new categorical priors thus remains an open problem.

Our contributions to “in the wild” species recognition challenges and benchmarks are described in Chapter 4. The presented fine-grained recognition challenges entail interesting sub-problems such as training with noisy labels and changes of the categorical priors between training and test data. The results validate that with large amounts of training data available, state-of-the-art Convolutional Neural Network architectures achieve the best results in the complex tasks of “in the wild” species classification. Our results in the international challenges, including the best results in ExpertLifeCLEF 2018, FGVCx Fungi 2018, FGVCx Flowers 2018 and PlantCLEF 2019 (post-challenge), confirm the benefits of practices such as combining predictions from an ensemble of models, filtering potentially noisy data, data augmentation, or using the moving averages of the trained variables. Experiments comparing the accuracy of machine learning models with human experts in plant identification suggest that the accuracy of our models reaches the human expert accuracy in image-based species recognition. However, it is important to note, that unlike the proposed model(s), human experts are mainly trained in active recognition – often deciding based on active physical examination of the specimen – and would be able to ask for missing information important for successful identification. Our models do not include such reasoning mechanisms.

The relatively high accuracy of our models motivated the application of the competition-winning method in a citizen-science project for fungi recognition, described in Chapter 5. With the integration of our fungi recognition system into the web and mobile interfaces of the Atlas of Danish Fungi, users get instant species recommendations. The feature of automatic species recognition increased the involvement of users in biodiversity data collection. This application shows the impact of our work on the species identification process, and the potential impact on other research areas, which will benefit from the collected data.

Bibliography

[1] Atlas of danish fungi. https://svampe.databasen.org/. Accessed: 2019-12-3.

[2] Encyclopedia of life. http://www.eol.org. Accessed: 2019-12-3.

[3] FGVCx flower classification challenge 2018. https://sites.google.com/view/fgvc5/competitions/fgvcx/flowers. Accessed: 2019-12-3.

[4] Global Biodiversity Information Facility. http://www.gbif.org. Accessed: 2019-12-3.

[5] iNaturalist. http://www.inaturalist.org. Accessed: 2019-12-3.

[6] Pairwise rotation invariant co-occurrence local binary pattern. http://qixianbiao.github.io. Accessed: 2019-12-3.

[7] Pl@ntnet. http://www.identify.plantnet.org. Accessed: 2020-01-14.

[8] Gaurav Agarwal, Peter Belhumeur, Steven Feiner, David Jacobs, W. John Kress, Ravi Ramamoorthi, Norman A. Bourg, Nandan Dixit, Haibin Ling, Dhruv Mahajan, et al. First steps toward an electronic field guide for plants. *Taxon*, 55(3):597–610, 2006.

[9] Timo Ahonen, Jiří Matas, Chu He, and Matti Pietikäinen. Rotation invariant image description with local binary pattern histogram fourier features. In *SCIA '09, in Proc.*, pages 61–70. Springer-Verlag, 2009.

[10] Carrie Andrew, Einar Heegaard, Paul M. Kirk, Claus Bässler, Jacob Heilmann-Clausen, Irmgard Krisai-Greilhuber, Thomas W. Kuyper, Beatrice Senn-Irlet, Ulf Büntgen, Jeffrey Diez, Simon Egli, Alan C. Gange, Rune Halvorsen, Klaus Høiland, Jenni Nordén, Fredrik Rustøen, Lynne Boddy, and Håvard Kauserud. Big data integration: Pan-european fungal species observations' assembly for addressing contemporary questions in ecology and global change biology. *Fungal Biology Reviews*, 31(2):88 – 98, 2017.

[11] Peter N. Belhumeur, Daozheng Chen, Steven Feiner, David W. Jacobs, W. John Kress, Haibin Ling, Ida Lopez, Ravi Ramamoorthi, Sameer Sheorey, Sean White, and Ling Zhang. Searching the world's herbaria: A system for visual identification of plant species. In *Computer Vision–ECCV 2008*, pages 116–129. Springer, 2008.

[12] Raquel Bello-Cerezo, Francesco Bianconi, Francesco Di Maria, Paolo Napoletano, and Fabrizio Smeraldi. Comparative evaluation of hand-crafted image descriptors vs. off-the-shelf CNN-based features for colour texture classification under ideal and realistic conditions. *Applied Sciences*, 9(4):738, 2019.

[13] Abhijit Bendale and Terrance Boult. Towards open world recognition. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1893–1902, 2015.

[14] Jean-Noël Biraben. Essai sur l'évolution du nombre des hommes. *Population*, 34(1):13–25, 1979.

[15] Christopher M. Bishop. *Pattern Recognition and Machine Learning.* Information Science and Statistics. Springer, 2006.

[16] Jean-Pierre Bocquet-Appel. When the world's population took off: the springboard of the neolithic demographic transition. *Science*, 333(6042):560–561, 2011.

[17] Pierre Bonnet, Hervé Goëau, Siang Thye Hang, Mario Lasseck, Milan Šulc, Valéry Malécot, Philippe Jauzein, Jean-Claude Melet, Christian You, and Alexis Joly. Plant identification: experts vs. machines in the era of deep learning. In *Multimedia Tools and Applications for Environmental & Biodiversity Informatics*, pages 131–149. Springer, 2018.

[18] Pierre Bonnet, Alexis Joly, Hervé Goëau, Julien Champ, Christel Vignau, Jean-François Molino, Daniel Barthélémy, and Nozha Boujemaa. Plant identification: man vs. machine. *Multimedia Tools and Applications*, 75(3):1647–1665, 2016.

[19] Safia Boudra, Itheri Yahiaoui, and Ali Behloul. A comparison of multi-scale local binary pattern variants for bark image retrieval. In *International Conference on Advanced Concepts for Intelligent Vision Systems*, pages 764–775. Springer, 2015.

[20] Stephen Boyd and Lieven Vandenberghe. *Convex optimization.* Cambridge university press, 2004.

[21] Christopher Brickell. *American horticultural society encyclopedia of plants and flowers.* Penguin, 2011.

[22] Phil Brodatz. *Textures: a photographic album for artists and designers*, volume 66. Dover New York, 1966.

[23] Gertjan J. Burghouts and Jan-Mark Geusebroek. Material-specific adaptation of color invariant features. *Pattern Recognition Letters*, 30(3):306–313, 2009.

[24] Stuart HM Butchart, Matt Walpole, Ben Collen, Arco Van Strien, Jörn PW Scharlemann, Rosamunde EA Almond, Jonathan EM Baillie, Bastian Bomhard, Claire Brown, John Bruno, et al. Global biodiversity: indicators of recent declines. *Science*, 328(5982):1164–1168, 2010.

[25] Barbara Caputo, Eric Hayman, and P. Mallikarjuna. Class-specific material categorisation. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 2, pages 1597–1604. IEEE, 2005.

[26] Mark Chandler, Linda See, Kyle Copas, Astrid MZ Bonde, Bernat Claramunt López, Finn Danielsen, Jan Kristoffer Legind, Siro Masinde, Abraham J. Miller-Rushing, Greg Newman, et al. Contribution of citizen science towards international biodiversity monitoring. *Biological Conservation*, 213:280–294, 2017.

[27] Arthur D. Chapman. *Numbers of living species in Australia and the world.* Department of the Environment, Water, Heritage and the Arts, Canberra, 2009.

[28] Kumar Chellapilla, Sidd Puri, and Patrice Simard. High performance convolutional neural networks for document processing. In *Tenth International Workshop on Frontiers in Handwriting Recognition*, 2006.

[29] Chi-hau Chen, Louis-François Pau, and Patrick Shen-pei Wang. *Handbook of pattern recognition and computer vision.* World Scientific, 2010.

[30] Zheru Chi, Li Houqiang, and Wang Chao. Plant species recognition based on bark patterns using novel gabor filter banks. In *International Conference on Neural Networks and Signal Processing, 2003. Proceedings of the 2003*, volume 2, 2003.

[31] 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.

[32] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. *arXiv preprint arXiv:1311.3618*, 2013.

[33] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, and Andrea Vedaldi. Deep filter banks for texture recognition, description, and segmentation. *International Journal of Computer Vision*, 118(1):65–94, 2016.

[34] Mircea Cimpoi, Subhransu Maji, and Andrea Vedaldi. Deep filter banks for texture recognition and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3828–3836, 2015.

[35] Dan Claudiu Ciresan, Ueli Meier, Jonathan Masci, Luca Maria Gambardella, and Jürgen Schmidhuber. Flexible, high performance convolutional neural networks for image classification. In *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.

[36] Department of Economic Citation: United Nations and Population Division Social Affairs. World population prospects 2019: Ten key findings. 2019.

[37] Robert T. Clemen and Robert L. Winkler. Combining probability distributions from experts in risk analysis. *Risk analysis*, 19(2):187–203, 1999.

[38] Quentin Cronk. Plant extinctions take time. *Science*, 353(6298):446–447, 2016.

[39] Gabriela Csurka. Domain adaptation for visual applications: A comprehensive survey. *arXiv preprint arXiv:1702.05374*, 2017.

[40] Yin Cui, Feng Zhou, Yuanqing Lin, and Serge Belongie. Fine-grained categorization and dataset bootstrapping using deep metric learning with humans in the loop. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1153–1162, 2016.

[41] Kristin J. Dana, Bram Van Ginneken, Shree K. Nayar, and Jan J. Koenderink. Reflectance and texture of real-world surfaces. *ACM Transactions on Graphics (TOG)*, 18(1):1–34, 1999.

[42] Danish Mycological Society. Atlas of Danish fungi. `https://svampe.databasen.org`, 2009. Accessed: 2019-12-3.

[43] Danish Mycological Society, Google Research. Danish Mycological Society fungi embedding model. TensorFlow hub. `https://doi.org/10.26161/mpbc-q144`.

[44] Morris H. DeGroot and Stephen E. Fienberg. The comparison and evaluation of forecasters. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 32(1-2):12–22, 1983.

[45] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. IEEE, 2009.

[46] Janis L. Dickinson, Jennifer Shirk, David Bonter, Rick Bonney, Rhiannon L. Crain, Jason Martin, Tina Phillips, and Karen Purcell. The current state of citizen science as a tool for ecological research and public engagement. *Frontiers in Ecology and the Environment*, 10(6):291–297, 2012.

[47] Alison Donnelly, Mike B. Jones, and John Sweeney. A review of indicators of climate change for use in Ireland. *International Journal of Biometeorology*, 49(1):1–12, 2004.

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

[49] John Duchi, Shai Shalev-Shwartz, Yoram Singer, and Tushar 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.

[50] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The Pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111(1):98–136, 2015.

[51] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The Pascal visual object classes (VOC) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

[52] Stefan Fiel and Robert Sablatnig. Automated identification of tree species from images of the bark, leaves and needles. Master thesis, Vienna University of Technology, 2010.

[53] Stefan Fiel and Robert Sablatnig. Automated identification of tree species from images of the bark, leaves and needles. In *Proc. of 16th Computer Vision Winter Workshop*, pages 1–6, Mitterberg, Austria, 2011.

[54] Mario Fritz, Eric Hayman, Barbara Caputo, and Jan-Olof Eklundh. The KTH-TIPS database, 2004.

[55] Tobias Guldberg Frøslev, Jacob Heilmann-Clausen, Christian Lange, Thomas Læssøe, Jens Henrik Petersen, Ulrik Søchting, Thomas Stjernegaard Jeppesen, and Jan Vesterholt. Danish mycological society, fungal records database, 2019.

[56] Kunihiko Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 36(4):193–202, 1980.

[57] Jonas Geldmann, Jacob Heilmann-Clausen, Thomas E. Holm, Irina Levinsky, Bo Markussen, Kent Olsen, Carsten Rahbek, and Anders P. Tøttrup. What determines spatial bias in citizen science? exploring four recording schemes with different proficiency requirements. *Diversity and Distributions*, 22(11):1139–1149, 2016.

[58] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *International conference on artificial intelligence and statistics*, pages 249–256, 2010.

[59] 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.

[60] Hervé Goëau, Pierre Bonnet, and Alexis Joly. Plant identification in an open-world (LifeCLEF 2016). In *Working Notes of CLEF 2016 - Conference and Labs of the Evaluation Forum*, 2016.

[61] 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.

[62] Hervé Goëau, Pierre Bonnet, and Alexis Joly. Plant identification based on noisy web data: the amazing performance of deep learning (LifeCLEF 2017). CEUR Workshop Proceedings, 2017.

[63] Hervé Goëau, Pierre Bonnet, and Alexis Joly. Overview of ExpertLifeCLEF 2018: how far automated identification systems are from the best experts? 2018.

[64] Hervé Goëau, Pierre Bonnet, and Alexis Joly. Overview of LifeCLEF plant identification task 2019: diving into data deficient tropical countries. In *CLEF working notes 2019*, 2019.

[65] Hervé Goëau, Pierre Bonnet, Alexis Joly, Vera Bakić, Julien Barbe, Itheri Yahiaoui, Souheil Selmi, Jennifer Carré, Daniel Barthélémy, Nozha Boujemaa, et al. Pl@ ntnet mobile app. In *Proceedings of the 21st ACM international conference on Multimedia*, pages 423–424. ACM, 2013.

[66] Yunchao Gong, Liu Liu, Ming Yang, and Lubomir Bourdev. Compressing deep convolutional networks using vector quantization. *arXiv preprint arXiv:1412.6115*, 2014.

[67] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[68] Ian J. Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron Courville, and Yoshua Bengio. Maxout networks. *arXiv preprint arXiv:1302.4389*, 2013.

[69] Yves Grandvalet and Yoshua Bengio. Entropy regularization. *Semi-supervised learning*, pages 151–168, 2006.

[70] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 1321–1330, 2017.

[71] Zhenhua Guo and David Zhang. A completed modeling of local binary pattern operator for texture classification. *IEEE Transactions on Image Processing*, 19(6):1657–1663, 2010.

[72] Song Han, Huizi Mao, and William J. Dally. Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding. *CoRR, abs/1510.00149*, 2, 2015.

[73] Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. In *Advances in Neural Information Processing Systems*, pages 1135–1143, 2015.

[74] Siang Thye Hang, Atsushi Tatsuma, and Masaki Aono. Bluefield (KDE TUT) at LifeCLEF 2016 plant identification task. In *Working Notes of CLEF 2016 - Conference and Labs of the Evaluation Forum*, 2016.

[75] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.

[76] Joseph K. Hawkins. Textural properties for pattern recognition. *Picture processing and psychopictorics*, pages 347–370, 1970.

[77] Eric Hayman, Barbara Caputo, Mario Fritz, and Jan-Olof Eklundh. On the significance of real-world conditions for material classification. In *Computer Vision-ECCV 2004*, pages 253–266. Springer, 2004.

[78] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2961–2969, 2017.

[79] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[80] Jacob Heilmann-Clausen, Hans Henrik Bruun, Rasmus Ejrnæs, Tobias Guldberg Frøslev, Thomas Læssøe, and Jens H. Petersen. How citizen science boosted primary knowledge on fungal biodiversity in denmark. *Biological Conservation*, 237:366 – 372, 2019.

[81] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[82] Zhi-Kai Huang, Chun-Hou Zheng, Ji-Xiang Du, and Yuan-yuan Wan. Bark classification based on textural features using artificial neural networks. In *Advances in Neural Networks-ISNN'2006*. Springer, 2006.

[83] Forrest N. Iandola, Song Han, Matthew W. Moskewicz, Khalid Ashraf, William J. Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and $<$ 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016.

[84] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[85] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2704–2713, 2018.

[86] Walter Jetz, Melodie A. McGeoch, Robert Guralnick, Simon Ferrier, Jan Beck, Mark J. Costello, Miguel Fernandez, Gary N. Geller, Petr Keil, Cory Merow, Carsten Meyer, Frank E. Muller-Karger, Henrique M. Pereira, Eugenie C. Regan, Dirk S. Schmeller, and Eren Turak. Essential biodiversity variables for mapping and monitoring species populations. *Nature Ecology & Evolution*, 3(4):539–551, 2019.

[87] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[88] Alexis Joly, Hervé Goëau, Christophe Botella, Hervé Glotin, Pierre Bonnet, Robert Planqué, Willem-Pier Vellinga, and Henning Müller. Overview of LifeCLEF 2018: a large-scale evaluation of species identification and recommendation algorithms in the era of ai. In *Proceedings of CLEF 2018*, 2018.

[89] 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.

[90] 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.

[91] Alexis Joly, Hervé Goëau, Christophe Botella, Stefan Kahl, Maximillien Servajean, Hervé Glotin, Pierre Bonnet, Willem-Pier Vellinga, Robert Planqué, Fabian-Robert Stöter, and Henning Müller. Overview of LifeCLEF 2019: Identification of Amazonian plants, South & North American birds, and niche prediction. In *Proceedings of CLEF 2019*, 2019.

[92] Walter S. Judd, Christopher S. Campbell, Elizabeth A. Kellogg, Peter F. Stevens, and Michael J. Donoghue. Taxonomy. In *Plant systematics - A Phylogenetic Approach, Third Edition*. Sinauer Associates, 2007.

[93] Abdul Kadir, Lukito Edi Nugroho, Adhi Susanto, and Paulus Insap Santosa. A comparative experiment of several shape methods in recognizing plants. *International Journal of Computer Science & Information Technology*, 3(3), 2011.

[94] Abdul Kadir, Lukito Edi Nugroho, Adhi Susanto, and Paulus Insap Santosa. Neural network application on foliage plant identification. *International Journal of Computer Applications*, 29, 2011.

[95] Abdul Kadir, Lukito Edi Nugroho, Adhi Susanto, and Paulus Insap Santosa. Experiments of zernike moments for leaf identification. *Journal of Theoretical and Applied Information Technology (JATIT)*, 41(1):82–93, 2012.

[96] Abdul Kadir, Lukito Edi Nugroho, Adhi Susanto, and Paulus Insap Santosa. Performance improvement of leaf identification system using principal component analysis. *International Journal of Advanced Science & Technology*, 44, 2012.

[97] G. Karuna, Birudu Sujatha, and P. Chandrasekhar Reddy. An efficient representation of shape for object recognition and classification using circular shift method. *Int. Journal of Scientific & Engineering Research*, 4(12):703–707, 2013.

[98] Fahad Shahbaz Khan, Rao Muhammad Anwer, Joost van de Weijer, Andrew D. Bagdanov, Maria Vanrell, and Antonio M. Lopez. Color attributes for object detection. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3306–3313. IEEE, 2012.

[99] Fahad Shahbaz Khan, Rao Muhammad Anwer, Joost van de Weijer, Michael Felsberg, and Jorma Laaksonen. Compact color–texture description for texture classification. *Pattern Recognition Letters*, 51:16–22, 2015.

[100] Fahad Shahbaz Khan, Joost Van de Weijer, and Maria Vanrell. Modulating shape features by color attention for object recognition. *International Journal of Computer Vision*, 98(1):49–64, 2012.

[101] Raees Khan, Joost Van de Weijer, Fahad Shahbaz Khan, Damien Muselet, Christophe Ducottet, and Christian Barat. Discriminative color descriptors. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2866–2873. IEEE, 2013.

[102] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009.

[103] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[104] Neeraj Kumar, Peter N. Belhumeur, Arijit Biswas, David W. Jacobs, W. John Kress, Ida C. Lopez, and João VB Soares. Leafsnap: A computer vision system for automatic plant species identification. In *Computer Vision–ECCV 2012*, pages 502–516. Springer, 2012.

[105] William Alexander Lambeth. *Trees, and how to know them: A manual with analytical and dichotomous keys of the principal forest trees of the South.* BF Johnson Publishing Company, 1911.

[106] Ron Lance. *Woody plants of the southeastern United States: a winter guide.* University of Georgia Press, 2004.

[107] Mario Lasseck. Image-based plant species identification with deep convolutional neural networks. *Working Notes of CLEF*, 2017, 2017.

[108] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. A sparse texture representation using local affine regions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1265–1278, 2005.

[109] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[110] Yann LeCun, Bernhard Boser, John S. Denker, Donnie Henderson, Richard E. Howard, Wayne Hubbard, and Lawrence D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[111] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[112] Kue-Bum Lee, Kwang-Woo Chung, and Kwang-Seok Hong. An implementation of leaf recognition system, 2013.

[113] Richard B. Lee, Richard Heywood Daly, Richard Daly, et al. *The Cambridge encyclopedia of hunters and gatherers.* Cambridge University Press, 1999.

[114] Peihua Li, Qilong Wang, and Lei Zhang. A novel earth mover's distance methodology for image matching with gaussian mixture models. In *Proc. of IEEE International Conference on Computer Vision (ICCV)*, 2013.

[115] Peihua Li, Jiangtao Xie, Qilong Wang, and Wangmeng Zuo. Is second-order information helpful for large-scale visual recognition? In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2070–2078, 2017.

[116] Wen Li, Limin Wang, Wei Li, Eirikur Agustsson, and Luc Van Gool. Webvision database: Visual learning and understanding from web data. *arXiv preprint arXiv:1708.02862*, 2017.

[117] Hsuan-Tien Lin, Chih-Jen Lin, and Ruby C. Weng. A note on platt's probabilistic outputs for support vector machines. *Machine learning*, 68(3), 2007.

[118] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017.

[119] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *Computer Vision – ECCV 2014*, pages 740–755. Springer, 2014.

[120] Carolus Linnaeus. *Species plantarum, exhibentes plantas rite cognitas ad genera relatas, cum differentiis specificis, nominibus trivialibus, synonymis selectis, locis natalibus, secundum systema sexuale digestas.* Laurentius Salvius, 1753.

[121] Carolus Linnaeus. *Systema naturae: per regna tria naturae, secundum classes, ordines, genera, species, cum characteribus, differentiis, synonymis, locis.* Laurentius Salvius, 1758.

[122] Chenxi Liu, Barret Zoph, Maxim Neumann, Jonathon Shlens, Wei Hua, Li-Jia Li, Li Fei-Fei, Alan Yuille, Jonathan Huang, and Kevin Murphy. Progressive neural architecture search. In *Computer Vision – ECCV 2018*, pages 19–34, 2018.

[123] Li Liu, Jie Chen, Paul Fieguth, Guoying Zhao, Rama Chellappa, and Matti Pietikäinen. From BoW to CNN: Two decades of texture representation for texture classification. *International Journal of Computer Vision*, 127(1):74–109, 2019.

[124] David G. Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. IEEE, 1999.

[125] Oisin Mac Aodha, Elijah Cole, and Pietro Perona. Presence-only geographical priors for fine-grained image classification. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9596–9606, 2019.

[126] Topi Mäenpää and Matti Pietikäinen. Multi-scale binary patterns for texture analysis. In *Image Analysis*, pages 885–892. Springer, 2003.

[127] P. Mallikarjuna, M. Fritz, A.T. Targhi, E. Hayman, B. Caputo, and J.O. Eklundh. The KTH-TIPS and KTH-TIPS2 databases. `http://www.nada.kth.se/cvap/databases/kth-tips`, 2006.

[128] Junhua Mao, Jun Zhu, and Alan L. Yuille. An active patch model for real world texture and appearance classification. In *Computer Vision–ECCV 2014*, pages 140–155. Springer, 2014.

[129] Majid Mirmehdi, Xianghua Xie, and Jasjit Suri. *Handbook of texture analysis*. Imperial College Press, 2009.

[130] Kevin P. Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[131] Alexandru Niculescu-Mizil and Rich Caruana. Predicting good probabilities with supervised learning. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 625–632. ACM, 2005.

[132] Petr Novotný and Tomáš Suk. Leaf recognition of woody species in central europe. *Biosystems Engineering*, 115(4):444–452, 2013.

[133] Kyoung-Su Oh and Keechul Jung. Gpu implementation of neural networks. *Pattern Recognition*, 37(6):1311–1314, 2004.

[134] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002.

[135] Timo Ojala, Matti Pietikainen, and David Harwood. Performance evaluation of texture measures with classification based on kullback discrimination of distributions. In *Proceedings of 12th International Conference on Pattern Recognition*, volume 1, pages 582–585 vol.1, 1994.

[136] Timo Ojala, Matti Pietikäinen, and David Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, 29(1):51–59, 1996.

[137] Christopher Olston, Noah Fiedel, Kiril Gorovoy, Jeremiah Harmsen, Li Lao, Fangwei Li, Vinu Rajashekhar, Sukriti Ramesh, and Jordan Soyke. Tensorflow-serving: Flexible, high-performance ML serving. *arXiv preprint arXiv:1712.06139*, 2017.

[138] Matthew G. Orton, Jacqueline A. May, Winfield Ly, David J. Lee, and Sarah J. Adamowicz. Is molecular evolution faster in the tropics? *Heredity*, 122(5):513, 2019.

[139] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *Automatica*, 11(285-296):23–27, 1975.

[140] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

[141] Jason Parham, Charles Stewart, Jonathan Crall, Daniel Rubenstein, Jason Holmberg, and Tanya Berger-Wolf. An animal detection pipeline for identification. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1075–1083. IEEE, 2018.

[142] Cynthia S. Parr, Nathan Wilson, Patrick Leary, Katja Schulz, Kristen Lans, Lisa Walley, Jennifer Hammock, Anthony Goddard, Jeremy Rice, Marie Studer, et al. The encyclopedia of life v2: providing global access to knowledge about life on earth. *Biodiversity data journal*, 2, 2014.

[143] Roger Phillips. *Mushrooms: And Other Fungi of Great Britain and Europe.* Pan Books, 1981.

[144] Lukáš Picek, Milan Šulc, and Jiří Matas. Recognition of the Amazonian flora by inception networks with test-time class prior estimation. *CLEF (Working Notes)*, 2019.

[145] Matti Pietikäinen. Texture recognition. *Computer Vision: A Reference Guide*, pages 789–793, 2014.

[146] Matti Pietikäinen, Timo Ojala, and Zelin Xu. Rotation-invariant texture classification using feature distributions. *Pattern Recognition*, 33(1):43–52, 2000.

[147] John Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3), 1999.

[148] Boris T. Polyak and Anatoli B. Juditsky. Acceleration of stochastic approximation by averaging. *SIAM journal on control and optimization*, 30(4):838–855, 1992.

[149] R. Pydipati, T.F. Burks, and W.S. Lee. Identification of citrus disease using color texture features and discriminant analysis. *Computers and electronics in agriculture*, 52(1):49–59, 2006.

[150] Xianbiao Qi, Rong Xiao, Jun Guo, and Lei Zhang. Pairwise rotation invariant co-occurrence local binary pattern. In *Computer Vision–ECCV 2012*, pages 158–171. Springer, 2012.

[151] Xianbiao Qi, Rong Xiao, Chun-Guang Li, Yu Qiao, Jun Guo, and Xiaoou Tang. Pairwise rotation invariant co-occurrence local binary pattern. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(11):2199–2213, 2014.

[152] Yuhui Quan, Yong Xu, Yuping Sun, and Yu Luo. Lacunarity analysis on image patterns for texture classification. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 160–167, 2014.

[153] Filip Radenović, Giorgos Tolias, and Ondrej Chum. Fine-tuning CNN image retrieval with no human annotation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(7):1655–1668, 2019.

[154] Joseph Redmon. Darknet: Open source neural networks in C. `http://pjreddie.com/darknet/`, 2013–2016.

[155] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016.

[156] 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.

[157] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, pages 91–99, 2015.

[158] Tim Robertson, Serge Belongie, Hartwig Adam, Christine Kaeser-Chen, Chenyang Zhang, Kiat Chuan Tan, Yulong Liu, Denis Brulé, Cédric Deltheil, Scott Loarie, Grant Van Horn, Oisin Mac Aodha, Sara Beery, Pietro Perona, Kyle Copas, and John Thomas Waller. Training machines to identify species using gbif-mediated datasets. *Biodiversity Information Science and Standards*, 3:e37230, 2019.

[159] Amelie Royer and Christoph H. Lampert. Classifier adaptation at prediction time. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1401–1409, 2015.

[160] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[161] Marco Saerens, Patrice Latinne, and Christine Decaestecker. Adjusting the outputs of a classifier to new a priori probabilities: a simple procedure. *Neural computation*, 14(1):21–41, 2002.

[162] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018.

[163] Walter J. Scheirer, Lalit P. Jain, and Terrance E. Boult. Probability models for open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(11):2317–2324, 2014.

[164] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

[165] Shai Shalev-Shwartz and Tong Zhang. Stochastic dual coordinate ascent methods for regularized loss minimization. *arXiv preprint arXiv:1209.1873*, 2012.

[166] Lavanya Sharan, Ruth Rosenholtz, and Edward Adelson. Material perception: What can you see in a brief glance? *Journal of Vision*, 9(8):784–784, 2009.

[167] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. CNN features off-the-shelf: an astounding baseline for recognition. In *2014 IEEE*

Conference on Computer Vision and Pattern Recognition Workshops, pages 806–813, 2014.

[168] Gaurav Sharma, Sibt ul Hussain, and Frédéric Jurie. Local higher-order statistics (lhs) for texture categorization and facial analysis. In *Computer Vision–ECCV 2012*, pages 1–12. Springer, 2012.

[169] Laurent Sifre and Stephane Mallat. Rotation, scaling and deformation invariant scattering for texture discrimination. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1233–1240. IEEE, 2013.

[170] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[171] Oskar Söderkvist. Computer vision classification of leaves from swedish trees. Master thesis, Computer Vision Laboratory, Linköping University, Sweden, 2001.

[172] Jiatao Song, Zheru Chi, Jilin Liu, and Hong Fu. Bark classification by combining grayscale and binary texture features. In *Proceedings of 2004 International Symposium on Intelligent Multimedia, Video and Speech Processing*, pages 4409–4417, 2004.

[173] Yang Song, Weidong Cai, Qing Li, Fan Zhang, David Dagan Feng, and Heng Huang. Fusing subcategory probabilities for texture classification. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4409–4417, 2015.

[174] Milan Šulc and Jiří Matas. Kernel-mapped histograms of multi-scale LBPs for tree bark recognition. In *Image and Vision Computing New Zealand (IVCNZ), 2013 28th International Conference of*, pages 82–87. IEEE, 2013.

[175] Milan Šulc and Jiří Matas. Tree identification from images. Master thesis, Czech Technical University in Prague, 2014.

[176] Milan Šulc and Jiří Matas. Fast features invariant to rotation and scale of texture. In *Computer Vision - ECCV 2014 Workshops*, volume 8926 of *LNCS*, pages 47–62. Springer, 2015.

[177] Milan Šulc and Jiří Matas. Texture-based leaf identification. In *Computer Vision - ECCV 2014 Workshop*, volume 8928 of *LNCS*, pages 185–200. Springer, 2015.

[178] Milan Šulc and Jiří Matas. Significance of colors in texture datasets. In *Proceedings of the 21st Computer Vision Winter Workshop*, Ljubljana, Slovenia, 2016. Slovenian Pattern Recognition Society.

[179] Milan Šulc and Jiří Matas. Fine-grained recognition of plants from images. *Plant Methods*, 13(1):115, 2017.

[180] Milan Šulc and Jiří Matas. Learning with noisy and trusted labels for fine-grained plant recognition. In *Working Notes of CLEF 2017 - Conference and Labs of the Evaluation Forum*, 2017.

[181] 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.

[182] Milan Šulc, Dmytro Mishkin, and Jiří Matas. Very deep residual networks with maxout for plant identification in the wild. In *Working Notes of CLEF 2016 - Conference and Labs of the Evaluation Forum*, 2016.

[183] Milan Šulc, Lukáš Picek, and Jiří Matas. Plant recognition by inception networks with test-time class prior estimation. In *CLEF (Working Notes)*, 2018.

[184] Milan Šulc, Lukáš Picek, Jiří Matas, Thomas Jeppesen, and Jacob Heilmann-Clausen. Fungi recognition: A practical use case. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 2316–2324, 2020.

[185] Milan Šulc, Albert Gordo Soldevila, Diane Larlus Larrondo, and Florent C. Perronnin. System and method for product identification, 2016. US Patent 9,443,164.

[186] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, volume 4, page 12, 2017.

[187] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015.

[188] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826, 2016.

[189] Muhammad Waseem Tahir, Nayyer Abbas Zaidi, Adeel Akhtar Rao, Roland Blank, Michael J. Vellekoop, and Walter Lang. A fungus spores dataset and a convolutional neural network based approach for fungus detection. *IEEE transactions on nanobioscience*, 17(3):281–290, 2018.

[190] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114, 2019.

[191] Jessica Thorn. State of the worlds plants 2016. Technical report, Royal Botanical Gardens, Kew, 2016.

[192] Julien Troudet, Philippe Grandcolas, Amandine Blin, Régine Vignes-Lebbe, and Frédéric Legendre. Taxonomic bias in biodiversity data and societal preferences. *Scientific Reports*, 7(1):9132, 2017.

[193] Kimmo Valkealahti and Erkki Oja. Reduced multidimensional co-occurrence histograms in texture classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):90–94, 1998.

[194] Koen E.A. Van De Sande, Theo Gevers, and Cees G.M. Snoek. Evaluating color descriptors for object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1582–1596, 2010.

[195] J.G. Van De Vooren, G. Polder, and G.W.A.M. van der Heijden. Identification of mushroom cultivars using image analysis. *Transactions of the ASAE*, 35(1):347–350, 1992.

[196] Joost Van De Weijer, Cordelia Schmid, Jakob Verbeek, and Diane Larlus. Learning color names for real-world applications. *IEEE Transactions on Image Processing*, 18(7):1512–1523, 2009.

[197] Arco J. van Strien, Menno Boomsluiter, Machiel E. Noordeloos, Richard J. T. Verweij, and Thomas W. Kuyper. Woodland ectomycorrhizal fungi benefit from large-scale reduction in nitrogen deposition in the netherlands. *Journal of Applied Ecology*, 55(1):290–298, 2018.

[198] Andrea Vedaldi and Brian Fulkerson. VLFeat: An open and portable library of computer vision algorithms. `http://www.vlfeat.org/`, 2008.

[199] Andrea Vedaldi and Andrew Zisserman. Efficient additive kernels via explicit feature maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(3):480–492, 2012.

[200] Yuan-Yuan Wan, Ji-Xiang Du, De-Shuang Huang, Zheru Chi, Yiu-Ming Cheung, Xiao-Feng Wang, and Guo-Jun Zhang. Bark texture feature extraction based on statistical texture analysis. In *Proceedings of 2004 International Symposium on Intelligent Multimedia, Video and Speech Processing*, 2004.

[201] Qilong Wang, Jiangtao Xie, Wangmeng Zuo, Lei Zhang, and Peihua Li. Deep CNNs meet global covariance pooling: Better representation and generalization. *arXiv preprint arXiv:1904.06836*, 2019.

[202] Weiran Wang and Miguel A. Carreira-Perpinán. Projection onto the probability simplex: An efficient algorithm with a simple proof, and an application. *arXiv preprint arXiv:1309.1541*, 2013.

[203] Jianxin Wu and Jim M. Rehg. Centrist: A visual descriptor for scene categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1489–1501, 2011.

[204] Jiaxiang Wu, Cong Leng, Yuhang Wang, Qinghao Hu, and Jian Cheng. Quantized convolutional neural networks for mobile devices. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4820–4828, 2016.

[205] Stephen Gang Wu, Forrest Sheng Bao, Eric You Xu, Yu-Xuan Wang, Yi-Fan Chang, and Qiao-Liang Xiang. A leaf recognition algorithm for plant classification using probabilistic neural network. In *Signal Processing and Information Technology, 2007 IEEE International Symposium on*, pages 11–16. IEEE, 2007.

[206] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1492–1500, 2017.

[207] Yong Xu, Hui Ji, and Cornelia Fermüller. Viewpoint invariant texture description using fractal analysis. *International Journal of Computer Vision*, 83(1):85–100, 2009.

[208] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Computer Vision – ECCV 2014*, pages 818–833. Springer International Publishing, 2014.

[209] Jianguo Zhang and Tieniu Tan. Brief review of invariant texture analysis methods. *Pattern recognition*, 35(3):735–747, 2002.

[210] Guoying Zhao, Timo Ahonen, Jiri Matas, and Matti Pietikainen. Rotation-invariant image and video description with local binary pattern features. *IEEE Transactions on Image Processing*, 21(4):1465–1477, 2012.

[211] Aojun Zhou, Anbang Yao, Yiwen Guo, Lin Xu, and Yurong Chen. Incremental network quantization: Towards lossless CNNs with low-precision weights. *arXiv preprint arXiv:1702.03044*, 2017.

[212] Bartosz Zielinski, Agnieszka Sroka-Oleksiak, Dawid Rymarczyk, Adam Piekarczyk, and Monika Brzychczy-Wloch. Deep learning approach to description and classification of fungi microscopic images. *CoRR*, abs/1906.09449, 2019.

[213] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V. Le. Learning transferable architectures for scalable image recognition. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8697–8710, 2018.

APPENDIX A

Author's Publications

* Citations indexed by Google Scholar as of 11th May 2020, without self-citations.

A.1 Publications Related to the Topic of the Thesis

A.1.1 Impacted Journal Paper

[179] Milan Šulc and Jiří Matas. Fine-grained recognition of plants from images. *Plant Methods*, 13(1):115, 2017.

(Impact Factor 3.170, WoS SCI-Expanded Q1)

12 citations*

A.1.2 Conference and Workshop Papers

[174] Milan Šulc and Jiří Matas. Kernel-mapped histograms of multi-scale LBPs for tree bark recognition. In *Image and Vision Computing New Zealand (IVCNZ), 2013 28th International Conference of*, pages 82–87. IEEE, 2013.

(Indexed in Web of Science)

18 citations*

[177] Milan Šulc and Jiří Matas. Texture-based leaf identification. In *Computer Vision - ECCV 2014 Workshop*, volume 8928 of *LNCS*, pages 185–200. Springer, 2015.

(Indexed in Web of Science)

13 citations*

[176] Milan Šulc and Jiří Matas. Fast features invariant to rotation and scale of texture. In *Computer Vision - ECCV 2014 Workshops*, volume 8926 of *LNCS*, pages 47–62. Springer, 2015.

(Indexed in Web of Science)

13 citations*

[178] Milan Šulc and Jiří Matas. Significance of colors in texture datasets. In *Proceedings of the 21st Computer Vision Winter Workshop*, Ljubljana, Slovenia, 2016. Slovenian Pattern Recognition Society.

[182] Milan Šulc, Dmytro Mishkin, and Jiří Matas. Very deep residual networks with maxout for plant identification in the wild. In *Working Notes of CLEF 2016 - Conference and Labs of the Evaluation Forum*, 2016.

9 citations*

[180] Milan Šulc and Jiří Matas. Learning with noisy and trusted labels for fine-grained plant recognition. In *Working Notes of CLEF 2017 - Conference and Labs of the Evaluation Forum*, 2017.

5 citations*

[183] Milan Šulc, Lukáš Picek, and Jiří Matas. Plant recognition by inception networks with test-time class prior estimation. In *CLEF (Working Notes)*, 2018.

5 citations*

[144] Lukáš Picek, Milan Šulc, and Jiří Matas. Recognition of the Amazonian flora by inception networks with test-time class prior estimation. *CLEF (Working Notes)*, 2019.

2 citations*

[181] 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.

1 citation*

[184] Milan Šulc, Lukáš Picek, Jiří Matas, Thomas Jeppesen, and Jacob Heilmann-Clausen. Fungi recognition: A practical use case. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 2316–2324, 2020.

(CORE rank A conference)

A.1.3 Book Chapter

[17] Pierre Bonnet, Hervé Goëau, Siang Thye Hang, Mario Lasseck, Milan Šulc, Valéry Malécot, Philippe Jauzein, Jean-Claude Melet, Christian You, and Alexis Joly. Plant identification: experts vs. machines in the era of deep learning. In *Multimedia Tools and Applications for Environmental & Biodiversity Informatics*, pages 131–149. Springer, 2018.

12 citations*

A.2 Publications Not Related to the Topic of the Thesis

A.2.1 US Patent

[185] Milan Šulc, Albert Gordo Soldevila, Diane Larlus Larrondo, and Florent C. Perronnin. System and method for product identification, 2016. US Patent 9,443,164.

11 citations*

APPENDIX B

MILAN ŠULC

Author's Curriculum Vitae

EDUCATION

PhD candidate in Artificial Intelligence and Biocybernetics (2014-2020)
Czech Technical University in Prague, FEE, Center for Machine Perception
Thesis: Fine-grained Recognition of Plants and Fungi from Images
Advisor: Prof. Jiří Matas

MSc in Computer Vision and Image Processing (2012-2014)
Czech Technical University in Prague, FEE
Thesis: Tree Identification from Images
Graduated Summa Cum Laude. Minor: Artificial Intelligence.

Engineering Exchange student (Spring semester 2014)
University of Wisconsin–Madison. GPA: 4.0

MSc in Entrepreneurship and Management in Industry (2012-2015)
Czech Technical University in Prague, MIAS

BSc in Cybernetics and Robotics (2009-2012)
Czech Technical University in Prague, FEE
Thesis: Image-based Recognition of Plants

PROFESSIONAL EXPERIENCE

Toyota/TRACE and **Czech Technical University in Prague** (2019-2020)
3D object detection from monocular camera(s) with applications to autonomous driving.

Google, **Mobile Vision team** (2017)
Internship, applications of Generative Adversarial Networks to fine-grained domains.

Electrolux and **Czech Technical University in Prague** (2015-2020)
Computer vision R&D projects of Electrolux and FEE CTU in Prague.

Xerox Research Centre Europe, Now NAVER Labs Europe (2014)
A computer vision R&D internship, US Patent no. 9443164.

University of Oxford, **Visual Geometry Group** (2013)
Machine learning internship, contributing to the open-source VLFeat library.

Czech Technical University in Prague (2011-2014)
Computer vision R&D at the Center for Machine Perception.

PUBLICATIONS

Fungi Recognition: A Practical Use Case,
Šulc M., Picek L., Matas J., Jeppesen T., Heilmann-Clausen J. WACV 2020.

Improving CNN Classifiers by Estimating Test-time Priors,
Šulc M., Matas J. ICCV Workshops, 2019.

Recognition of the Amazonian Flora by Inception Networks with Test-time Class Prior Estimation,
Picek L., Šulc M., Matas J. LifeCLEF 2019, in Working Notes of CLEF 2019.

Plant Recognition by Inception Networks with Test-time Class Prior Estimation,
Šulc M., Picek L., Matas J. ExpertLifeCLEF 2018, in Working Notes of CLEF 2018.

Plant Identification: Experts vs. Machines in the Era of Deep Learning (Book Chapter), Bonnet P., Goeau H., Hang S.T., Lasseck M., Šulc M., Malecot V., Jauzein P., Melet J-C., You Ch., Joly A. In A. Joly, S. Vrochidis, K. Karatzas, A. Karppinen, P. Bonney (Ed.) Multimedia Tools and Applications for Environmental & Biodiversity Informatics. 2018. ISBN: 978-3-319-76445-0.

Fine-grained Recognition of Plants from Images,
Šulc M., Matas J. Plants in Computer Vision [Special Issue], Plant Methods. 2017.
ISSN: 1746-4811. Impact Factor 3.51

Learning with Noisy and Trusted Labels for Fine-Grained Plant Recognition,
Šulc M., Matas J. LifeCLEF 2017, in Working Notes of CLEF 2017.

Very Deep Residual Networks with Maxout for Plant Identification in the Wild,
Šulc M., Mishkin D., Matas J. LifeCLEF 2016, in Working Notes of CLEF 2016.

Significance of Colors in Texture Datasets,
Šulc M., Matas J. 21st Computer Vision Winter Workshop, 2016.

Fast Features Invariant to Rotation and Scale of Texture,
Šulc M., Matas J. ECCV Workshops, Springer LNCS, 2014.

Texture-Based Leaf Identification,
Šulc M., Matas J. ECCV Workshops, Springer LNCS, 2014.

Kernel-mapped Histograms of Multi-scale LBPs for Tree Bark Recognition,
Šulc M., Matas J. Image and Vision Computing New Zealand 2013.

Patents

System and Method for Product Identification,
Šulc M., Gordo A., Larlus D., and Peronnin F. US Patent no. 9443164. Owner: Xerox Corp.

Attended Conferences, Workshops, etc.

CVWW 2020, **ICCV 2019**, CLEF 2019, CVWW 2019, CLEF 2018, **CVPR 2018**, **ICCV 2017**, CLEF 2017, Google PIRC 2017, 1st Winter School in CSE on Computer Vision in Jerusalem 2017, BMVA tech. meeting on Plants in Computer Vision 2016, Google Computer Vision PhD Summit 2016, **CVPR 2016**, CVWW 2016, VS3 2015, **ECCV 2014**, ERC ALLEGRO workshop 2014, IVCNZ 2013.

Other Activities

Labs teacher at FEE CTU in Prague: Computer Vision Methods for MSc students (2015-2017); Problem Solving and Games for BSc students (2016-2018), Pattern Recognition and Machine Learning for BSc students (2018-2020). Master thesis advisor (2018).

Member of the **Disciplinary commission of FEE CTU**. (2016-2018)

International Student Club, CTU in Prague: Language Teacher (2013), Visa Coordinator (2015).

Prizes

1st place in the FGVCx Fungi and FGVCx Flowers fine-grained recognition challenges organized with the FGVC5 workshop at CVPR 2018.

1st place in the ExpertLifeCLEF 2018 Plant Identification Task.

Prize of Josef Hlávka for the best students and graduates, 2014,
The Foundation of Josef, Marie and Zdeňka Hlávka.

Best internship project presentation, 2014, Xerox Research Centre Europe.

Dean's prize for outstanding Master thesis, 2014, CTU in Prague, FEE.

Prize of the Masaryk Institute director for an original and precise thesis, 2015, CTU in Prague, MIAS.