Towards Multi-Level Classification in Deep Plant Identification

Thesis submitted for consideration of an evaluating tribunal
of the Interuniversity Program of Doctorate of Engineering to opt for the degree of Doctor of Engineering

Jose Carranza-Rojas
Post-graduate Studies Unit-Instituto Tecnológico de Costa Ricay-Universidad de Costa Rica

Cartago
November 28th, 2018

In compliance with the regulations of the Doctoral Program of Engineering at the Instituto Tecnológico de Costa Rica, the thesis presented by the candidate to doctor is accepted. The Doctoral Examination was presented on the day 28 of November of the year 2018. The Evaluating Court of this thesis was integrated by the following members:

Erick Mata-Montero, Ph.D.	Director—Advisor
Juan Luis Crespo Mariño, D.Eng.	Co-advisor 1
Dagoberto Arias, Ph.D.	Co-advisor 2
Teodolito Guillen, Dr.-Ing.	Post Graduate Program Director
Herve Goeau, Ph.D.	External Researcher/Professor

Dedication

To my advisor, Prof. Erick Mata-Montero, for his guidance during this research.

To my family, my life companion Adriana and our child Santiago, for feeding me with immeasurable love and energy to keep going. To my mother, for being a life example to me.

To Dr. Herve Goeau, Dr Pierre Bonnet, Dr. Alexis Joly, Remi Knaff and Antoine Affouard, for their insights to complement this research, but more importantly, for their friendship and making me feel like home during my stay in France.

To Nelson Zamora, Armando Estrada and Alexander Rodríguez, as well as the National Museum of Costa Rica, for their constant support in the taxonomy side of this research. To the Costa Rica Institute of Technology, for granting me the opportunity to learn and teach about science and engineering.

To my dissertation committee formed by Dr. Erick Mata-Montero, Dr. Juan Luis Crespo, Dr. Dagoberto Arias, Dr. Teodolito Guillén, and Dr. Herve Goeau, for taking the time to read, correct and support this dissertation.

To all scientists out there, that crave for an understanding of how the cosmos works.

To science and nature, for allowing my natural curiosity about our cosmos to be fulfilled.

This research wouldn't have been possible without you all. Thank you.

Abstract

In the last decade, automatic identification of organisms based on computer vision techniques has been a hot topic for both biodiversity scientists and machine learning specialists. Early on, plants became particularly attractive as a subject of study for two main reasons. On the one hand, quick and accurate inventories of plants are critical for biodiversity conservation; for example, they are indispensable in conducting ecosystem inventories, defining models for environmental service payments, and tracking populations of invasive plant species, among others. On the other hand, plants are a more tractable group than, for instance, insects. First of all, the number of species is smaller (around 400,000 compared to more than 8 million). Secondly, they are better understood by the scientific community, particularly with respect to their morphometric features. Thirdly, there are large, fast growing databases of digital images of plants generated by both scientists and the general public. Finally, an incremental approach based first on "flat elements" such as leaves and then the whole plant made it feasible to use computer vision techniques early on. As a result, even mobile apps for the general public are available nowadays.

This document presents the key results obtained while tackling the general problem of fully automating the identification of plant species based solely on images. It describes the key findings in a research path that started with a restricted scope, namely, identification of plants from Costa Rica by using a morphometric approach that considers images of fresh leaves only. Then, species from other regions of the world were included, but still using hand-crafted feature extractors. A key methodological turn was the subsequent use of Deep Learning techniques on images of any components of a plant. Then we studied and compared the accuracy of a Deep Learning approach to do identifications based on datasets of images of fresh plants and compared it with datasets of herbarium sheet images for the first time. Among the results obtained during this research, potential biases in automatic plant identification dataset were found and characterized. Feasibility of doing transfer learning between different regions of the world was also proven. Even more importantly, it was for the first time demonstrated that herbarium sheets are a good resource to do identifications of plants mounted on herbarium sheets, which provides additional levels of importance to herbaria around the globe. Finally, as a culmination of this research path, this document presents the results of developing a novel multi-level classification approach that uses knowledge about higher taxonomic levels to carry out not only family and genus level identifications but also to try to improve the accuracy of species level identifications. This last step focuses on the creation of a hierarchical loss function based on known plant taxonomies, coupled with multi-level Deep Learning architectures to guide the model optimization with the prior knowledge of a given class hierarchy.

Keywords: Deep Learning, Hierarchical Loss Functions, Hierarchical Classification, Automatic Plant Identification, Herbarium, Computer Vision, Artificial Intelligence.

Resumen

En la última década, la identificación automática de organismos basada en técnicas de visión artificial ha sido un tema popular tanto entre los científicos de la biodiversidad como para los especialistas en aprendizaje automático. Al principio, las plantas se volvieron particularmente atractivas como tema de estudio por dos razones principales. Por un lado, los inventarios rápidos y precisos de plantas son críticos para la conservación de la biodiversidad; por ejemplo, son indispensables para realizar inventarios de ecosistemas, definir modelos para pagos de servicios ambientales y rastrear poblaciones de especies de plantas invasoras, entre otros. Por otro lado, las plantas son un grupo más manejable que, por ejemplo, los insectos. En primer lugar, la cantidad de especies es menor (alrededor de 400,000 en comparación con más de 8 millones de insectos). En segundo lugar, la comunidad científica las comprende mejor, en particular con respecto a sus características morfométricas. En tercer lugar, existen grandes bases de datos de imágenes digitales de plantas generadas tanto por científicos como por el público en general. Finalmente, un enfoque incremental basado primero en "elementos planos" como hojas y luego en toda la planta hizo posible el uso de técnicas de visión por computadora desde el principio. Como resultado, incluso las aplicaciones móviles para el público en general están disponibles en la actualidad.

Este documento presenta los resultados clave obtenidos mientras se aborda el problema general de automatizar por completo la identificación de especies de plantas basándose únicamente en imágenes. Describe los hallazgos clave en un camino de investigación que comenzó con un alcance restringido, a saber, la identificación de plantas de Costa Rica mediante el uso de un enfoque morfométrico que considera imágenes de hojas frescas solamente. Luego, se incluyeron especies de otras regiones del mundo, pero todavía se utilizaban extractores de características hechos a mano. Un giro metodológico clave fue el uso posterior de técnicas de aprendizaje profundo (deep learning) en imágenes de cualquier componente de una planta. Luego, estudiamos y comparamos la exactitud de un enfoque de aprendizaje profundo para realizar identificaciones basadas en conjuntos de datos de imágenes de plantas frescas y las comparamos con conjuntos de datos de imágenes de hojas de herbario por primera vez. Entre los resultados obtenidos durante esta investigación, se encontraron y caracterizaron posibles sesgos en el conjunto de datos de identificación automática de plantas. La viabilidad de hacer un aprendizaje de transferencia (transfer learning) entre diferentes regiones del mundo también se demostró. Aún más importante, por primera vez se demostró que las láminas de herbario son un buen recurso para hacer identificaciones de plantas montadas sobre láminas de herbario, lo que proporciona niveles adicionales de importancia para herbarios en todo el mundo. Finalmente, como una culminación de este camino de investigación, este documento presenta los resultados del desarrollo de un nuevo enfoque de clasificación multi-nivel (multi-level) que utiliza el conocimiento sobre niveles taxonómicos superiores para llevar a cabo identificaciones a nivel de familia y género, y también para tratar de mejorar la exactitud de identificaciones a nivel de especie. Este último paso se centra en la creación de una función de pérdida jerárquica basada en taxonomías de plantas conocidas, junto con arquitecturas de aprendizaje profundo de niveles múltiples para guiar la optimización del modelo con el conocimiento previo de una jerarquía de clases dada.

Palabras clave: Aprendizaje Profundo, Funciones de Pérdida Jerárquica, Clasificación Jerárquica, Identificación Automática de Plantas, Herbario, Visión Artificial, Inteligencia Artificial.

Contents

List of Figures

List of Tables

Acronyms

ADBC Advancing the Digitization of Natural History Collections

ANN Artificial Neural Network

BHC Binary Hierarchical Classifier

CBD Convention on Biological Diversity

CeNAT Costa Rica National High Technology Center

CIRAD Centre de Cooperation Internationale en Recherche Agronomique pour le Developpement

CNN Convolutional Neural Network

CNNs Convolutional Neural Networks

CR CRLeaves

DAG Directed Acyclic Graph

DN Deconvolutional Networks

EM Expectation-Maximization

EoL Encyclopedia of Life

FCM Flat Classification Model

FFT Fast Fourier Transform

GAN Generative Adversarial Network

GLM General Linear Model

GTSDM Gray Tone Spatial Dependency Matrix

GPU Graphics Processing Unit

H1K Herbaria1K

H255 Herbarium255

HCoS Histogram of Curvature over Scale

HD-CNN Hierarchical Deep Convolutional Neural Network

H-Softmax Hierarchical Softmax

HSV Hue Saturation Value

I ImageNet

ICA Independent Component Analysis

IH1K ImageNet, Herbarium1K

IH255 ImageNet, Herbarium255

IM1K ImageNet, Merged1K

INBio Instituto Nacional de Biodiversidad, Costa Rica

INRIA Institut National de Recherche en Informatique et en Automatique

IPC ImageNet, PlantCLEF

kNN k Nearest Neighbors

LBP Local Binary Pattern

LBPU Local Binary Pattern Uniform

LBPV Local Binary Pattern Variance

LCL Local Classifier Per Level

LCN Local Classifier Per Node

LCPN Local Classifier Per Parent Node

LSTM Long-Short Term Memory

LVQ Learning Vector Quantization

M1K Merged1K

MCM Multi-Task Classification Model

MLNP Mandatory Leaf Node Prediction

NLP Natural Language Processing

NMLNP Non-mandatory Leaf Node Prediction

NSF United States National Science Foundation

PC PlantCLEF

PCA Principal Component Analysis

T-Softmax Taxonomy Softmax

PNN Probabilistic Neural Network

PRELU Parametric Rectified Linear Unit

RBF Radial Base Function

RELU Rectified Linear Unit

RGB Red Green Blue

RNN Recurrent Neural Networks

SIFT Scale-Invariant Feature Transform

SGD Stochastic Gradient Descent

SPP Spatial Pyramid Pooling

SSPB Same-Specimen-Picture Bias

SURF Speeded Up Robust Features

SVM Support Vector Machines

TCN Thematic Collections Networks

TSLA Triangle Side Lengths and Angle

UHMT Unconstrained Hit-or-Miss Transform

XML Extensible Markup Language

Chapter 1

Introduction and General Background

Absence of evidence is not evidence of absence.

Carl Sagan

1 Introduction

In the last decade, automatic plant identification based on computer vision techniques has been studied by pioneering research groups around the globe, resulting even in mobile apps for the general public (Joly et al. 2014a; Kumar et al. 2012). Research was initially focused on leaves for easier segmentation and on traditional computer vision approaches based on hand-crafted feature extractors (Kumar et al. 2012; Mata-Montero et al. 2015; Wu et al. 2007). However, botanists use not only leaves but also other parts of plants to do identifications. Thus, more realistic, global, and complete datasets and techniques were required and started to appear, for instance, the dataset defined by the PlantCLEF challenge (Goëau et al. 2015), which used images of plants taken in the field.

Another important source of information is herbaria. Herbaria maintain treasures of information in the form of dried plants that have been compiled over long periods of time and have been or are in the process of being fully identified by experts. However, herbaria images are hard to work with, given the great amount of visual noise and artifacts present in the herbaria sheets (Mata-Montero et al. 2016). This research explores the possibility of using such data for automatic plant identification.

With the rise of Deep Learning in 2012, during the ImageNet competition (Krizhevsky et al. 2012), along with the availability of Graphics Processing Unit (GPU) for extensive computing, it is now possible to work with complex background images and go beyond images of leaves or individual components for plant identifications. Thus, current state-of-the art research is focused on deep models based on convolutional neural networks that infer which visual patterns are significant, instead of hand-crafting the feature extractors as done previously.

Regardless of datasets, research in this area has been done at the species level. However, in many cases predicting the species gives a very low accuracy given inter- and intra-specific variability and similarity. This is often due to taxonomically unbalanced datasets, which is common in biodiversity informatics. Thus, it is relevant to explore hierarchies/taxonomy for classification and optimization of the Deep Learning models, in order to do identification at several taxon levels. Classification can be achieved by using multi-label deep architectures, where a plant image can be classified with more than one label, where each label belongs to a different taxonomic level in the hierarchy. This is a type of multi-label problem, but focused on the plant taxonomy domain, meaning, different labels of a training example have a hierarchical relation. We call this special case of multi-label classification, *multi-level classification*, where a training sample belongs to several labels which form part of a hierarchy of classes. In this case, of the plant taxonomy.

Another approach is to take into account higher taxon levels to classify species, with a more informed model and/or loss function that uses the taxonomy. For example, under the same genus one could have two species X and Y, where species X has too few images but species Y has enough images. Because X and Y share some features (they belong to the same genus) intuition suggests that the identification of specimens that belong to species X could benefit from referring to the images of species Y.

Given that the number of species of plants on the planet is estimated at around 400,000, hierarchical and multi-level approaches could also help avoid huge flat classifiers for large number of classes defined by extensive geographic regions or rich taxonomic groups.

One can devise at least two approaches to exploit the class hierarchy present in the plant taxonomy. One is the creation of new model architectures that deal with multi-level classification, where a single model can actually classify species of plants into several class levels, such as species, genera and families. The other is the creation of new Deep Learning approaches including new loss functions, that guide the model optimization in a novel way by using information about the hierarchy to calculate the loss.

This document is a dissertation for a Doctor of Engineering degree in the domain of Artificial Intelligence to study multi-level Deep Learning models with hierarchical loss functions for automatic plant identification. However, it also presents the key results obtained while addressing the general problem of fully automating the identification of plant species based exclusively on images. It describes the key findings in a four year research path that started with a restricted scope, namely, identification of plants from Costa Rica by using a morphometric approach that considers only images of fresh leaves (Chapter 4, Chapter 5). Then, species from other regions of the world were included, but still using hand-crafted feature extractors. A fundamental methodological turn was the subsequent use of Deep Learning techniques on images of any components of a plant (Chapter 6). Then we studied and compared the accuracy of a Deep Learning approach to do identifications based on datasets of images of fresh plants and compared it with datasets of herbarium sheet images for the first time (Chapter 7, Chapter 8). Additionally, because these are data-driven processes, it is critical to use statistically representative data. Thus, potential biases in automatic plant identification dataset creation and usage were found and characterized (Chapter 9). Feasibility of doing transfer learning between different regions of the world was also proven (Chapter 7). Even more importantly, it was for the first time demonstrated that herbarium sheets are a good resource to do identifications of plants mounted on herbarium sheets, which provides additional levels of value and importance to herbaria around the globe (Chapter 7). As a culmination of this research path, this document presents the results of developing novel multi-level classification architectures that use knowledge about higher taxonomic levels to carry out not only species identification but also family and genus level identifications. This last step responds to the research goals established for this dissertation but is the result of ground work which has already been published as a peer-reviewed paper (Carranza-Rojas et al. 2018). Finally, to improve the accuracy of species level identifications, the last chapter of this document focuses on the creation of a hierarchical loss function based on known plant taxonomies, used to guide the model optimization with the prior knowledge of a given class hierarchy, such as genus or family (Chapter 11).

The dissertation is structured as a collection of papers, where each chapter corresponds to published paper (with exception of the last chapter that has not yet been published). Nevertheless, Chapter 1 provides an introduction to the whole research, also related work and the problem statement. Chapter 2 describes the objectives of this research. Chapter 4 depicts the methodology as a whole, even when each other chapter has its own methodology. Middle chapters correspond to each paper. Also, results and discussion are present in each auto-contained chapter corresponding to the results of each paper, so it is missing as a stand alone

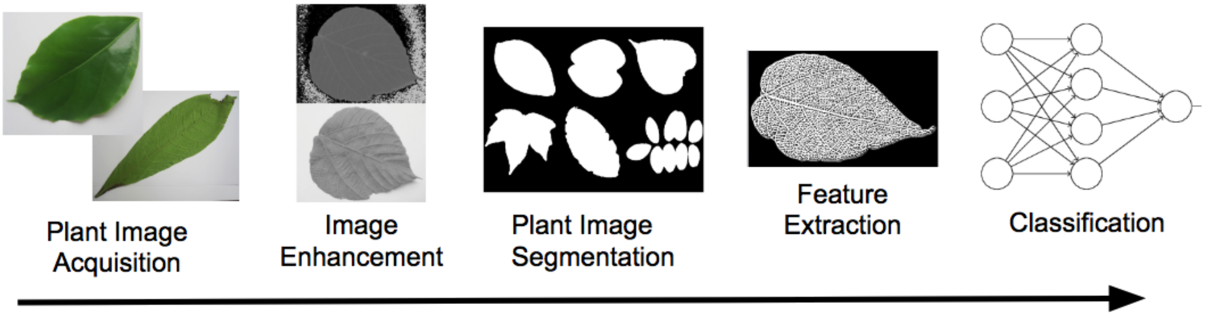


Figure 1.1. Typical plant identification phases of traditional approaches used before Deep Learning.

chapter. Finally, Chapter 12 shows the aggregated conclusions and future work of the whole research.

The rest of this chapter is organized as follows: Section 2 describes the theoretical framework of the proposed research. It starts with a description of traditional computer vision approaches, then it presents a summary of Deep Learning models and concludes with a description of hierarchical approaches to species identification. Section 3 summarizes the problem description and Section 4 states the hypothesis of this research.

2 Theoretical Framework

Automated plant identification research has evolved from using traditional computer vision techniques to so called Deep Learning approaches (Goodfellow et al. 2016). The latter provides better accuracy results across the board (Joly et al. 2014b) and avoids explicit segmentation from uniform background images by using convolutions instead of hand-crafted feature extractors. Overall, it offers the opportunity of classifying very complex images by using images of leaves, flowers and other components of plants, as well as images of the whole organism (Goëau et al. 2015). However, in general, it requires larger datasets of images and higher computing power (during its training phase) than traditional techniques. In the remaining of this section, we briefly summarize most relevant traditional approaches, introduce state-of-the-art techniques based on Deep Learning, and conclude with a literature review of hierarchical approaches for Deep Learning, in particular for multi-level classification and optimization, where a hierarchy of classes is present.

2.1 Traditional Plant Identification Techniques

Traditional computer vision approaches are based on hand-crafted feature extractors and classifiers (Beghin et al. 2010; Kumar et al. 2012; M. Z. Rashad 2011; Wu et al. 2007). Most authors divide the plant recognition process into five phases, namely, plant image acquisition, image enhancement, image segmentation, feature extraction, and classification. Most of the work focuses on plant identification based on leaf images. Figure 1.1 shows these different phases of the pipeline.

Plant Image Acquisition. Many of the datasets that have been developed use uniform backgrounds for easier segmentation, particularly for leaf recognition. For instance, the LeafSnap dataset (Kumar et al. 2012) consists of images from 184 tree species from Northeastern USA. It includes 23,916 images of fresh leaves with white backgrounds. The Swedish Dataset (Söderkvist 2001) consists of 15 species with high inter-species similarity. It contains 75 images per species. The Flavia Dataset (Wu et al. 2007) comprises 32 species with 3,621 leaf images with white backgrounds. Leaves were sampled from the Nanjing University campus, and Sun Yat-Sen Arboretum, Nanking, China. Jointly with the National Museum of Costa Rica we developed a dataset with a total of 255 species and 7.5∼k leaf images (Mata-Montero et al. 2015). These datasets based only on leaves are small in size and in amount of species, making them not very suitable for Deep Learning techniques and for current expectations of taxonomic coverage.

To our knowledge, only few studies have created their dataset directly from herbarium sheet images. In (Wijesingha et al. 2012) 79 images of the Stemonoporus genus were used. They were obtained from the National Herbarium at the Royal Botanic Garden, Sri Lanka. The images have rather low resolution of 120×120 pixels, and the dataset is small. Nevertheles plenty of herbarium images are available now thanks to systems like iDigBio [1] and can be used to increase the size of a global dataset as of September 2018 it has a total of $26{,}685{,}787$ media records.

In all previous studies, the amount of data used is too small to obtain conclusive results for a given flora from one region. This because thousands of species may have high intra- and inter-specific variability such as belonging to the same genus. It is imperative to create a consolidated, big dataset aiming to have all species of the world.

Image Enhancement. Once images are acquired, the next phase consist on pre-processing the image to enhance important features (Vishakha Metre 2013). This step includes gray scale conversion (Aggarwal et al. 2012; Arun et al. 2013; Herdiyeni et al. 2012; Kadir et al. 2011; Larese et al. 2014; Li et al. 2006; M. Z. Rashad 2011; Nguyen et al. 2013; Pietikäinen et al. 2011; R.D et al. 2011), noise reduction and other color domain conversions such as HSV (Kumar et al. 2012). The goal is to delete undesired noise and distortions that may affect the subsequent image segmentation and feature extraction (Beghin et al. 2010; J. et al. 2012).

Plant Image Segmentation. Once the dataset is created, images are cleaned and noise is filtered out. The next step is to extract the leaf from the image (Kumar et al. 2012). Most studies deal with clean leaf images with uniform backgrounds and use color clustering techniques to extract the leaf pixels. For instance, some very basic thresholding is used in (Larese et al. 2014; Lee et al. 2013a,b), while Expectation-Maximization (EM) is used in (Kumar et al. 2012; Mata-Montero et al. 2015) to cluster the HSV space pixels into leaf and non-leaf groups. In addition, Graph-Cut has also been used to find a global optimum segmentation solution (Soares et al. 2013).

Feature Extraction. Several hand-crafted features have been used to describe plant images. Early approaches use morphological characteristics such as area, leaf perimeter, and rectangularity, among others (Arora et al. 2012; Bhardwaj et al. 2013; Herdiyeni et al. 2012; Lee et al. 2013a,b; R.D et al. 2011; Wu et al. 2007). Later approaches use shapes or contour with different descriptors such as Histogram of Curvature over Scale (HCoS) (Kumar et al. 2012; Wu et al. 2007). Also, texture data has been used as alternative or complementary information by using, for example, Local Binary Pattern (LBP) and Local Binary Pattern Variance (LBPV) (Arun et al. 2013; Beghin et al. 2010; Mata-Montero et al. 2015; Vishakha Metre 2013; Wu et al. 2007). Very few studies have focused on veins (Larese et al. 2014; Lee et al. 2013a,b), as vein extraction is a very hard problem by itself. Nguyen et al. 2013 use Scale-Invariant Feature Transform (SIFT) descriptors for leaf recognition as well.

[1] https://www.idigbio.org/

Classification. Support Vector Machines (SVM) have been used most of the time as binary classifiers to determine if an image has a leaf present of if it is an image related to some other domain (Herdiyeni et al. 2012; Kumar et al. 2012; Nguyen et al. 2013). This is actually a pre-classification before everything else, to avoid wasting time classifying non-leaf or non-plant images. The last piece of the traditional pipeline consists of a classifier such as Artificial Neural Network (ANN) (Beghin et al. 2010; Herdiyeni et al. 2012, 2013; Kadir et al. 2011; M. Z. Rashad 2011; Wu et al. 2007) or k Nearest Neighbors (kNN) (Arun et al. 2013; Bhardwaj et al. 2013; Kumar et al. 2012; Mata-Montero et al. 2015), which is fed with the features extracted in previous steps of the pipeline and allows to predict the species.

2.2 State-of-the-Art Plant Identification

The PlantCLEF challenge is part of the LifeCLEF challenge (Joly et al. 2015a). LifeCLEF has been running since 2011. It aims at improving the state of the art in image-based organismal identifications by making scientists compete with predefined image datasets of plants, birds, and fish. The PlantCLEF challenge not only includes leaf-scan images but also other components such as fruits, stems, flowers, and others. This makes sense since botanists in real life do not use only leaves, as for most species leaves are not enough for identification (Choi 2015). Since 2015, the best results have been obtained by using Deep Learning models, in particular Convolutional Neural Networks (CNNs) (Joly et al. 2015b). Since then, this has been the norm, but it is not surprising since CNNs learn which features minimize the loss the best, given a different number of kernels, instead of having previous hand-crafted feature extractors. In 2012, the ImageNet challenge (Russakovsky et al. 2015), being a more generalized Computer Vision challenge (not only plants), saw for the first time the power of Deep Learning with the work of Krizhevsky et al. 2012, bringing down the error rate from 26.2% to 15.3% on general classes. The following sections describe the state-of-the-art in automatic plant identification based on Deep Learning techniques and the PlantCLEF challenges as source of both datasets and better identification systems.

Deep Learning Models

The first type of Deep Learning model used during the ImageNet competition is the so called Convolutional Neural Networks (CNNs) (Krizhevsky et al. 2012). These models make use of convolutions in order to extract meaningful patterns of the data. The idea is not new. It dates back to 1990 (Cun et al. 1990). However, until now we did not have the computational resources to make it possible, particularly, computers with multiple, powerful, and relatively inexpensive GPUs. CNNs have only one input vector and map it to a unique output vector. They also make use of convolutions. A convolution is an operation over two functions that produces a third function. It is shift-invariant, meaning we compute the same operation for every point in the image. Convolutions are also linear operations. CNNs are differentiable, thus, trainable using back propagation and techniques such as Stochastic Gradient Descent (SGD). In practice, a *kernel* or *filter* is initialized randomly or using some data distribution, and the other function is the image itself. After applying the convolution across all points in the image, we obtain a list of new images where the kernel was applied to each point in the image. These resulting images are called *feature maps* or *activation maps*. We then move the kernel to the next region both horizontally and vertically. The kernels are actually formed by trainable parameters, that change using back-propagation with respect to the model loss, in order to minimize it. This way, kernels are learned and once the model converges, they allow to detect useful and meaningful patterns in the image.

Figure 1.2 shows a typical CNN architecture. The input image is passed as a matrix to the initial 2d convolution, which has certain number of kernels that will convolute over the image. Typically, pooling layers are present after the convolution layer, in order to reduce dimensionality of the convolutional layer output. Min

Pooling, Max Pooling, are some of the types of pooling layers available. Additionally, both convolutional layers and pooling layers can be stacked on top of previous ones. Then, a dense or fully connected layer is added as a classifier, followed by a softmax probability distribution and finally a loss function such as cross entropy loss. The model can be optimized based on the value of the loss function, normally using back-propagation.

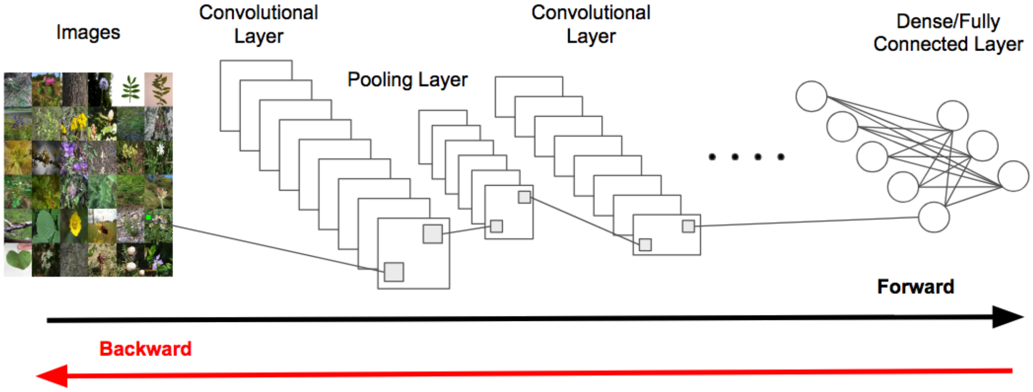


Figure 1.2. A Convolutional Neural Network (CNN) representation.

Several architectures have been used for plant identification, particularly with images in-the-wild. GoogLeNet is a very popular one (Szegedy et al. 2015), as well as VGGNet (Simonyan et al. 2014). These models originated in the ImageNet challenge (Russakovsky et al. 2015) and were ported to become plant identification systems with some adaptations. Figure 1.3(taken from (Canziani et al. 2016)) shows different current popular models. It can be noticed that there is a proportional relation between top-1 accuracy using ImageNet and the amount of operations done in the models. Inception-v4 (Szegedy et al. 2016) remains as one of the best models tested with ImageNet, providing high accuracy and having good amount of operations and parameters, revealing how good the inception modules work (Szegedy et al. 2015). This model however has not been used for plant identification to date.

Transfer learning is a technique used by most Deep Learning practitioners to train models based on previous training. It consists of using the weights of a previously trained model for a different problem domain (Yosinski et al. 2014). In practice, most people use weights related to ImageNet. (Yosinski et al. 2014) study in depth how transfer learning behaves through the layers of a CNN. They measure the accuracy by preserving each of the layers of the model to transfer learn to another domain. They found that the first layers of models pre-trained with ImageNet are generic enough to be used on any domain, as they learn generic patterns of the images that are not attached to any particular domain. As the layers distance from the input, the meaning and patterns become more abstract and related to the domain at hand. The usage of pre-trained models using ImageNet-related weights improves the accuracy around 9% to 10% according to results of the competition (Choi 2015). In Chapter 7, we prove that it is possible to improve automatic identification systems from one region of the world to another by using transfer learning.

Deep Learning for Automatic Plant Identification

Lee et al. 2015 show one of the first attempts of using Deep Learning for plant identification, particularly with CNNs. They also use Deconvolutional Networks (DN) to describe visually how the patterns are built, starting with generic blobs until some vein patterns emerge. Venation of different orders are chosen by the model for pattern recognition at different layers, which reflects how texture is key for plant identification (Mata-

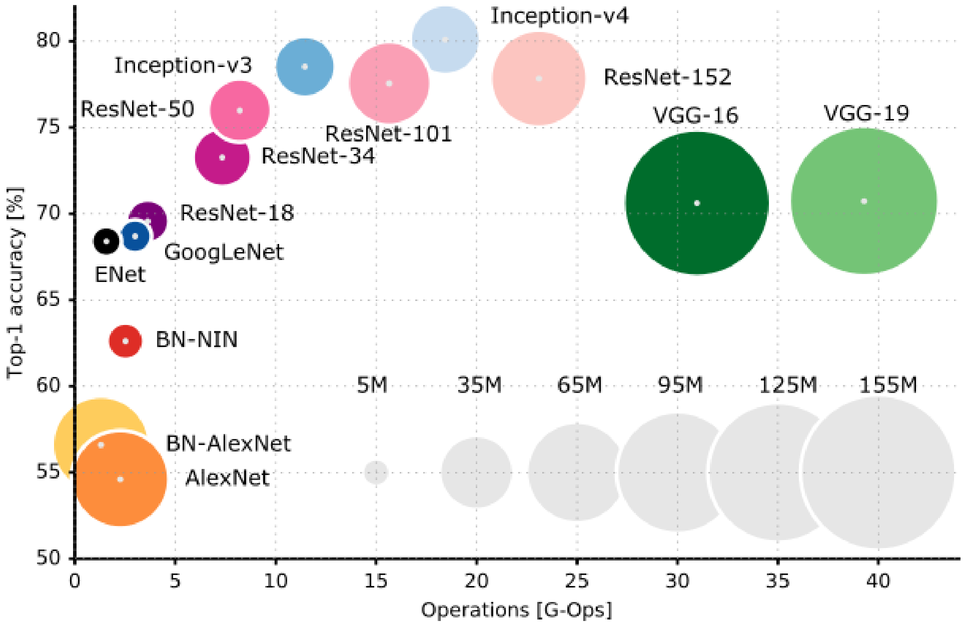


Figure 1.3. Deep Learning Model Comparison (Canziani et al. 2016)

Montero et al. 2016). The model used is AlexNet (Krizhevsky et al. 2012), and transfer learning is applied from ImageNet (Russakovsky et al. 2015). The used dataset contains leaf images from England, with a total of 44 species. The achieved accuracy is 99.5%. This, however, may not be conclusive because the amount of classes is rather small for a deep learning model, and the dataset may have hidden biases, as covered by Chapter 9 of this thesis.

Grinblat et al. 2016 use a custom CNN formed by two convolution layers and Rectified Linear Unit (RELU) non-linearities between them. Other developed models contain up to five layers, all with a softmax at the end. The reported accuracy is very high, of 98.8%, however the dataset used in very small and has a limited number of species of white bean, red bean and soy bean from their previous work (Larese et al. 2014).

Bonnet et al. 2015 study how the state-of-the-art computer vision systems perform compared to humans. Using a subset of the PlantCLEF 2014 dataset (Joly et al. 2014b), 500 species of trees, herbs and ferns are used, with a total of 19,504 observations. An observation consists of several images of the same specimen. It is important to notice that this dataset is closer to the real world: it has pictures taken in the wild, from both novice and expert users, containing flowers, leaves, fruits, branches, stems, using the Pl@ntNet application (Joly et al. 2014a). The best results obtained are correlated with better quality of both images and human annotations across the board. The authors conclude that predictions are affected by the quality of the testing images as well as the acquisition conditions, and that humans are still better at identification.

During the PlantCLEF 2015 challenge, species quantity escalated from 500 to 1,000 species from the West Europe region, with 113,205 images belonging to 41,794 observations from 8,960 contributors (Goëau et al. 2015). Also, challenge authors allowed to use external data for training. Images belong to leaves, flowers, stems, entire plant, fruit, and leaf scans. The test set was built from users from Pl@ntNet (Joly et al. 2014a). From a total of 26 runs, the best 9 runs used the GoogLeNet architecture, demonstrating the supremacy of Deep Learning methods over traditional computer vision approaches. Especially with the usage of transfer learning from ImageNet weights. Overall, best results were obtained from running the models over the flower and leafscan images. The winner team was from Korea (named SNUMED) which based their GoogLeNet model with transfer learning on five-fold different classifiers and then combined them altogether for better predictions, reaching a score of 67.7% (Choi 2015).

The following year, during the PlantCLEF 2016 challenge, species number was kept to 1,000 with more than 110k images. The organizers released a paper summarizing the challenge results (Goëau et al. 2016). The dataset from 2015 was kept but enriched with additional meta-data, in particular plant organ data. Teams from Japan, Hungary, Czech Republic, France, Belgium, Australia, Turkey, Malasya participated in the challenge. The best average precision achieved was of 71.8% by the Japanese team Bluefield (Hang et al. 2016) which used a VGGNet (Simonyan et al. 2014) deep model with Spatial Pyramid Pooling (SPP). SPP allows to use images of several resolutions without previously cropping and/or resizing (He et al. 2014), allowing to use the real size images prior to start the convolution process. They also used PRELU as activation functions (He et al. 2015). An important approach by Hang et al. 2016 is that they used convolutional layers for species and organs separately, and then they merge both into a single set of features. This is possible due to the metadata present in the PlantCLEF dataset. The top 26 runs with best performance were based on CNNs. In general, by adding non-plant images and unknown species to the challenge dataset, the performance of the model degrades (Goëau et al. 2016), as expected in a real life scenario, where users may upload unrelated images from time to time.

In 2017, the PlantCLEF 2017 challenge provided a bigger dataset, with a total of 10,000 species from North America and Europe (Goeau et al. 2017). This dataset, even though is big, does not reflect the total amount of plant species in the world, but does reflect the efforts towards building such dataset. The dataset is actually divided in two subsets: a "trusted" dataset, with 10,000 species and 256,287 images, which is taken from Encyclopedia of Life (EoL), with a lot of unbalanced classes. The "noisy" training dataset, built using a web crawler such as Google and Bing image search engines, contains 1.1M images, and it is highly unbalanced. The test set comes from the Pl@ntNet application, however the number of images is not provided in (Goeau et al. 2017). The winner of 2017 was the Mario TSA team (Lasseck 2017). They used GoogLeNet, ResNeXT and ResNet-152 architectures. Data augmentation was key for their results, as they generated 5 times more data augmented artificially. They achieved a top-1 accuracy of 88.5% and top-5 accuracy of 96.2%, which are extremely impressive given the difficulty of the task. PlantCLEF 2017 has the biggest plant identification dataset known to date. But is far from having the aproximate of 400,000 species on earth. To help building such dataset, we proved in Chapter 7 and Chapter 8 the possibility of usage of herbarium sheet images for automatic plant identification using Deep Learning technologies, which provides additional importance to herbarium institutions around the globe, allowing the usage of "old" data for innovative purposes.

2.3 *Multi-level* Classification

All previous work mentioned, even state-of-the-art Deep Learning models, do not make use of genera, family of higher taxon levels beyond species. In general, very few researchers have focused in hierarchical classification or hierarchical loss functions. Some datasets do offer some sort of class hierarchy, but in the image domain is not common to exploit it.

We define *multi-level* classification as a special case of multi-label classification. In normal multi-label classification, a training sample is associated with more than one label (Goodfellow et al. 2016), but the labels are not part of a class hierarchy. In what we define as *multi-level* classification, the labels have a ancestor-descendant relationship, where a label belongs to a certain level of a hierarchy. In our case, this hierarchy is a taxonomy of plants.

To our knowledge, no authors have tried formally to use plant taxonomies to classify organisms not only at the species level but also at the genus, family, order or other higher taxon level. Also, to our knowledge, no authors have tried to use other taxon level knowledge to do the species classification with a more informed approach. Our work in Chapter 10 shows some of the first work towards multi-level classification of plants based on images. The work presents several architectures that allow classification at several levels of the hierarchy at the same time, even with parameter sharing between them. Additionally, in Chapter 11 we present a new loss function that takes into account higher class levels to calculate its loss. This means it guides the loss optimization in a different fashion than the commonly used cross entropy with softmax functions.

Flat Classification

Flat Classification consists of completely ignoring the class hierarchy, and it is the most widely used nowadays in Deep Learning (Silla et al. 2011). The class hierarchy is looked as a flat class sequence, where only the leaves of the hierarchy are taken into account. One classifier, particularly a dense layer, is used after the convolutions, pooling layers, and others, allowing to classify only one level of classes. The final layer is actually the loss function calculation, normally a cross-entropy function using softmax to calculate the probability distribution over the predictions of the flat dense layer (Goodfellow et al. 2016). In case of plant taxonomies, the leaves of the taxonomy are the species, so the deep model ends with a classifier over the number of species or classes. However, there is an opportunity to exploit additional information of a class hierarchy, where species are grouped in genera, genera in families, and so on.

Hierarchical Classification

Deep learning models build concepts based on learning simpler concepts (Goodfellow et al. 2016). They are considered hierarchical in the sense of concept abstractions, however, as classification techniques, most models are flat: they classify only one level of classes and very few studies have taken into account hierarchies of classes in the loss functions for classification. We believe this is due to lack of datasets where there is an intrinsic need for having hierarchies of classes. However, in our domain, plants do have a completely defined taxonomy which can be exploited.

To our knowledge very few studies have tackle the problem of real hierarchical classification. Silla et al. 2011 present a very thorough survey about different techniques used for hierarchical classification. They also layout a unifying framework to classify existing approaches. It is important to notice that the survey does not focus on Deep Learning, but on traditional machine learning.

One possible solution for hierarchical classification is to use a multi-label approach, where each item has several labels associated with different classes or level of classes (Silla et al. 2011). With this approach, there is actually no need to have levels of a hierarchy. In a sense, a multi-label approach is not a hierarchical approach but often is used as a first attempt to tackle hierarchical problems (Silla et al. 2011). In (Goodfellow et al. 2014), Google researchers created a multi-label model for house number recognition. In total they add five different dense layers at the end of a CNN. Each dense layer is associated with each digit of the house numbers, plus an additional dense layer for the length of the house number. In this case it is not exactly a hierarchy, however there is one classifier per digit. The achieved accuracy is 97.84% and it is currently used in Google Maps for house numbering effects. A similar approach can be taken for plants, where one classifier exists for each level of the taxonomy, without exploiting knowledge between the dense layers. In Chapter 10 we base our hierarchical architectures on this same idea for multi-label classification, but we also provide an architecture that does multi-level classification.

Another approach is to add new layers that somehow capture semantical information about the class hierarchies. In (Goo et al. 2016), the network learns feature maps calculated by category. These feature maps are fed to the taxonommy layers. The first taxonomy layer, a generalization layer, learns supercategory feature maps by using min-pooling which capture shared features between categories. Then, the specialization layer learns exclusive per category feature maps by using difference-pooling.

Katole et al. 2015 create an architecture that has one root level classifier that classifies the image in a high level category. Then, several sub-classifiers are available at the low level categories, where the highest score is selected as the right class among all of those sub-classifiers. This approach is similar to the approach in (Yan et al. 2015) called Hierarchical Deep Convolutional Neural Network (HD-CNN). Again, this approach works on two level hierarchies. Their approach is a two level architecture and does not scale well as each parent category has its own classifier.

Class Hierarchy Representation. A hierarchy of classes can be represented with a Directed Acyclic Graph (DAG), where a node or class may have one or more parents. Using DAG's for hierarchical classification is more complex than using trees. In this proposal, we focus on hierarchies with tree topology, such as the plant taxonomy, in which a species has only one parent genus; a genus has only one parent family; and so on.

Inference of the Hierarchy. Also called Hierarchical Clustering or sometimes Structured Clustering (Silla et al. 2011), Yan et al. 2015 attempt to build a hierarchy directly from the data, most of the time using clustering techniques. The question of how many clusters remains open, since most of the time they are picked up manually. Also, by inferring the taxonomy, there is no warranty that the resulting hierarchy is useful for the user (Silla et al. 2011). In this research, we go another route in regards of the taxonomy: instead of inferring it using unsupervised learning, we use the pre-existing one defined by plant botanists, in order to make use of the expert knowledge already defined during centuries of research in the botanical domain.

Global versus Local Classifiers

When doing hierarchical classification, literature mentions 2 types of classifier configuration: local and global. A global classifier basically is just one big, complex classifier that takes into account the whole taxonomy, while the local classification configuration is divided into several classifiers with a local goal in mind (Silla et al. 2011).

Local Classifier Approaches, also called "top-down" classifiers, employ a set of local classifiers at different levels. This means the most generic class level is predicted first, then the next one and so on. As a result, an error can be propagated downwards after a bad decision. The system uses local information at each node, parent or even at the whole hierarchy level (Silla et al. 2011). In Local Classifier Per Node (LCN) the system consists of training a binary classifier at each node of the hierarchy as shown in Figure 1.4. For Local Classifier Per Parent Node (LCPN) there is a multi-class classifier trained to distinguish between child classes, as shown in Figure 1.5. Finally, in Local Classifier Per Level (LCL), there is a complete classifier for each level of the hierarchy, as shown in Figure 1.6. The major drawback in this last one is the class membership inconsistency. A Global Classifier Approach uses a single and relatively complex classifier is built from the training set. Basically any approach not considered as a local classification approach is considered global or also "big-bang" approach (Silla et al. 2011).

In (Yan et al. 2015) a model called HD-CNN is created, which makes use of the local classification approach. They define a 2-level classification scheme where the first classifier classifies a coarse classification followed by several fine-grained classifiers. They learn a coarse classification from the data. It is important to notice that their approach allows only for two level classification, while we aim to use the knowledge of all levels available in the plant taxonomy.

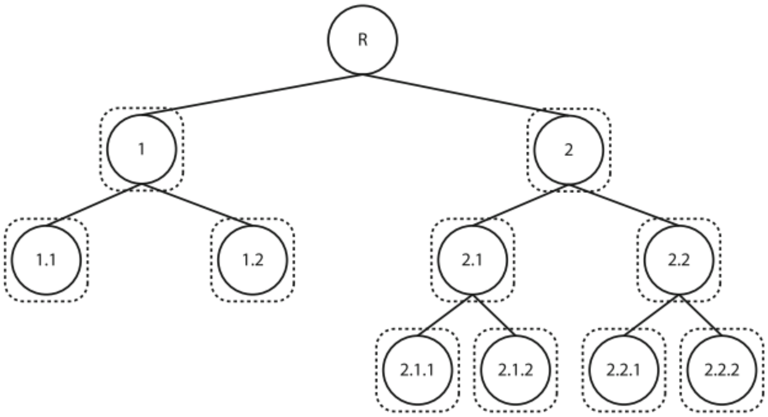


Figure 1.4. Local Classifier Per Node (LCN) (Silla et al. 2011)

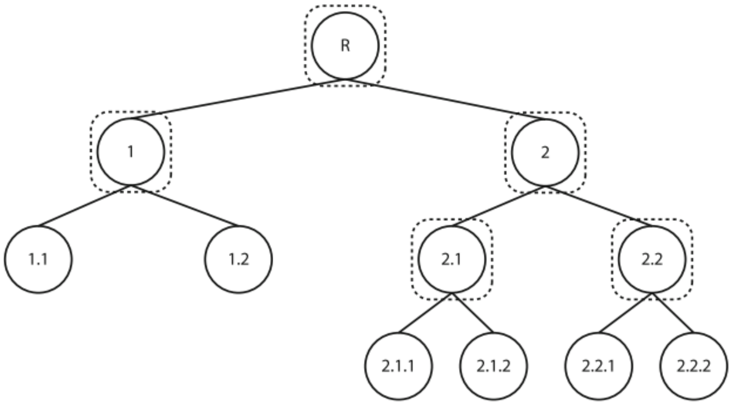


Figure 1.5. Local Classifier Per Parent Node (LCPN) (Silla et al. 2011)

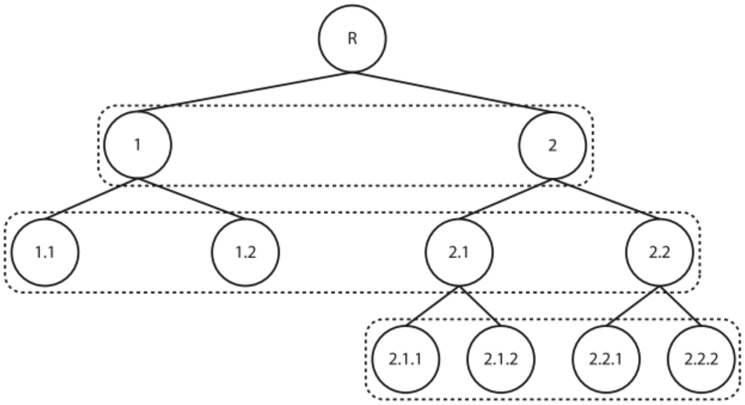


Figure 1.6. Local Classifier Per Level (LCL) (Silla et al. 2011)

Hierarchical Loss Functions

Most of the available literature about Deep Learning loss functions is related to flat loss functions such as the widely used cross entropy with softmax (Goodfellow et al. 2016). There are, however, a some existing previous work on the domain of hierarchical loss functions.

Mnih et al. 2009 define the foundations of the Hierarchical Softmax (H-Softmax) loss function, based on the softmax non-linearity. This is by far the most known and used hierarchical loss function. Their work is focused on text prediction, in order to predict the next word given the context (several previous words). Their technique requires a definition of a tree to index all the words of a vocabulary. The tree is a Huffman tree: a balanced binary tree. The defined loss function is good to predict one or maybe a few words, in which case the algorithm complexity becomes logarithmic. The path to any leaf is always different and takes $O(log(N))$ decisions instead of $O(N)$ as the traditional flat softmax, with N being the number of leaves of the tree, or in this case, the number of words (classes). However, if the complete probability distribution is needed, then it becomes linear with respect to all the classes N. In our case, this technique is not good since it requires the class hierarchy to be balanced and formed by binary nodes. The plant taxonomy is quite unbalanced, especially the datasets available for automatic plant identification, and a class may not be subdivided in just 2 child classes.

A hierarchical loss function that penalizes mistakes of the model at several hierarchy levels is discussed in (Wu et al. 2017), created during the same time as this research. Their approach uses ultra-metric trees, which have the same distance from the root to all leaves. The probability of one node is the sum of probabilities of all the leaves under such node. Then, a weighted sum is done along the path from the root to the leaf node corresponding to the correct class. The authors use a softmax probability distribution to calculate the probability of all leaves. In general, they did not get much better accuracy results compared to normal cross entropy loss with softmax, and even for some cases, got worst accuracy. They conclude that a good model optimization with their hierarchical loss function will depend on the dataset and the hierarchy at hand.

In Chapter 11 we tackle the creation of a different loss function that uses the sum of probabilities of species by grouping them by higher class levels, such as genera or family

3 Problem Description

The estimated number of plant species on Earth is around 400,000 species. In order to classify all known species, scientists will face lack of computational resources with flat classifiers that have too many classes. A natural way to deal with such huge number of classes is to take into account the taxonomy, i.e., a hierarchy of classes.

Traditional organism classification has been done mostly at species level (Joly et al. 2014b; Kumar et al. 2012; Mata-Montero et al. 2016). However, sometimes models are not very accurate for certain species. This is due to unbalanced datasets and lack of enough images for certain species, as well as intra- and inter-specific similarities (Mata-Montero et al. 2016).

In contrast, to our knowledge, very few studies tackle the problem of classifying plants at different taxonomic levels beyond species, such as genus and family. This is a particular instance of what we call *multi-level* classification. This problem could be tackled using new model architectures that allow classification at more than one taxonomic level simultaneously.

Another open problem is the guided optimization of the deep learning model by using the hierarchy of classes. New loss functions could be developed that penalize those classes that were correctly estimated when their siblings were not, or when their ancestor was not. In case of plants, the whole taxonomy knowledge established by taxonomists could guide this optimization process. Additionally, the taxonomy may have also a regularization effect depending on the definition of such loss functions.

Finally, a global dataset of plant species has to be developed. The PlantCLEF (Goëau et al. 2015) organization team has started this process, adding not only leaf scans but also in-situ images of fruits, flowers, stems, entire plants, and so on. However for some species they is a huge lack of images. A potential solution to this problem is the usage of herbaria images, which have been digitalized across the world and systems like iDigBio offer publicly. However, these images contain lots of visual noise as per human manipulation for preservation in controlled environments. This makes it a particularly hard problem for computer vision techniques, however remains to be seen how Deep Learning tackles this problem. Thus, a big dataset of herbarium images must be created and evaluated with these Deep Learning techniques in order to check the viability of plant identification using herbarium data.

4 Hypothesis

1. By taking into account an established plant taxonomy, the development of new loss functions leads to improvements in accuracy and convergence in Deep Learning models for automatic plant identification.

2. Learning constrained weights for each taxon level term in the hierarchical loss function provides better accuracy.

3. The new hierarchical loss functions coupled with multi-label classification architectures provide better accuracy than existing flat classifiers and allow to classify not only at species level but at any level of the taxonomy.

4. Images from herbarium sheets are useful for automated plant identification.

Chapter 2

Objectives and Contributions

> Science is organized knowledge. Wisdom is organized life.
>
> Immanuel Kant

This chapter explains the objectives and main contributions of this dissertation. Section 1 presents the main objective, followed by the specific objectives in Section 2. Section 3 provides the list of contributions to the scientific community. Finally, Section 4 depicts the scope and limitations.

1 General Objective

Develop and implement hierarchical loss function focused on loss functions, coupled with Deep Learning architectures for hierarchical or multi-level classification, taking into account plant taxonomy developed by botanists.

2 Specific Objectives

1. Present research results developed during previous phases of this work that lead to this dissertation.

2. Develop the theoretical framework for hierarchical loss functions, using plant taxonomy as the predefined hierarchy.

3. Implement the hierarchical loss function in a known Deep Learning framework for experimentation.

4. Design multi-level classification architectures using Deep Learning techniques.

5. Couple both the hierarchical loss function and the multi-level architectures for multiple taxon classification.

6. Provide a herbarium sheet based dataset for experimentation, in order to see how viable it is for plant identification.

3 Contributions

- Result of previous work that lead to this dissertation.
- Mathematical framework for hierarchical loss function based on existing hierarchies.
- Code for a new hierarchical loss function.
- Models and code for multi-level classification architectures using Deep Learning.
- Experimentation of convergence and accuracy by using both multi-level architectures and hierarchical loss function.
- A dataset based on herbaria sheet images for experimentation.
- A paper about the viability of using herbaria as dataset for plant identification.
- A paper about hierarchical loss function and multi-level classification of plant with Deep Learning.

4 Scope

This research includes work related to automatic plant identification using images, multi-level architectures of Deep Learning models, and hierarchical loss function development. It does not include a new purely hierarchical classifier. Instead, we will use multi-level architectures based on neural networks and hierarchical loss functions.

Chapter 3

Methodology

No amount of experimentation can ever prove me right; a single experiment can prove me wrong.

Albert Einstein

1 Introduction

This chapter covers the methods, techniques, tools, and tasks used in this research from a broad perspective. Methodological details for specific experiments are presented in each of the following chapters, each of which corresponds to a self-contained publication. However, the general methodological layout of all this research is explained in this chapter.

This chapter is organized as follows: Section 2 explains the different datasets used and also created for this research. Section 3 covers the methodology used on initial hand-crafted approaches, which were later replaced by Deep Learning, as explained on Section 5. In particular, this section covers new approaches using Deep Learning used to classify several levels of classes, beyond only one flat level as traditionally done in literature. Additionally, new loss functions are developed for this multi-level classification, in an attempt to optimize the models by taking advantage of the prior knowledge of the plant taxonomy. Finally, Section 6 discusses the different software tools used to implement the different models of this research.

2 Image Acquisition and Datasets

Several datasets were used during this research. Some of them already existed as part of the known literature, others were built from scratch as new contributions for the scientific community. Most datasets are unbalanced, as it is the common nature of the biodiversity informatics domain.

2.1 In-situ Datasets

The first datasets used were mostly leaf-only datasets. The leaves appear in an uniform background, as they were used with hand-crafted approaches that made harder the segmentation of plant organs in complex

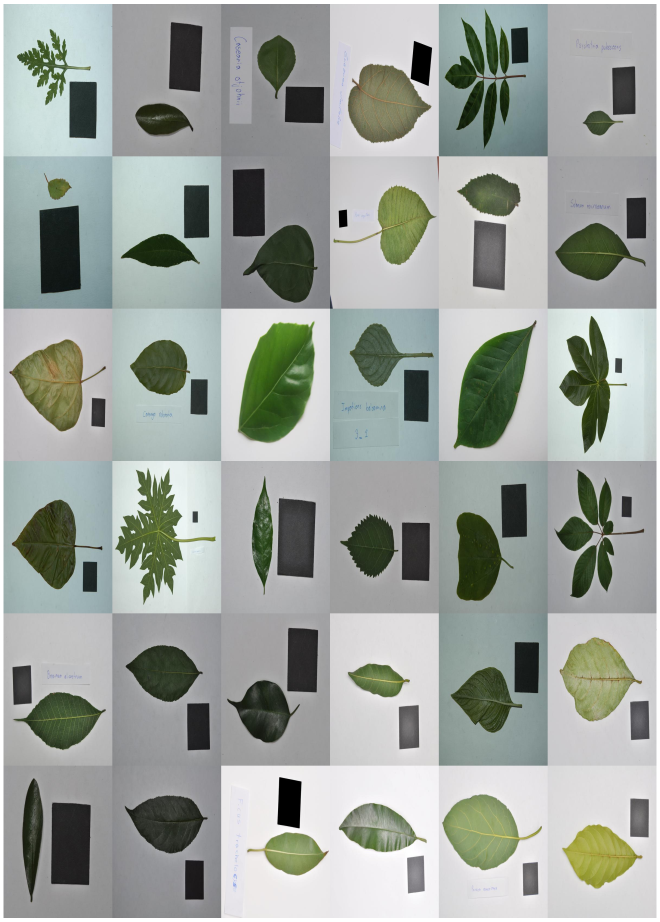


Figure 3.1. Random sample of different leaves with uniform background from the CRLeaves (CR) dataset used in Chapter 4, Chapter 5, Chapter 7 and Chapter 9.

backgrounds. Later, more complex dataset with more species and more images was used for Deep Learning approaches.

- CRLeaves (CR): the Costa Rica Leaf Scan Dataset includes a total of 255 species from the Central Plateau in Costa Rica on its complete version. An initial version of this dataset comprising 66 species was used in Chapter 4 and Chapter 5, as both were the first work of this research, done with hand-crafted features. Chapter 7 and Chapter 9 use the complete version which consists of 7,262 images digitized jointly by the National Museum of Costa Rica and the Costa Rica Institute of Technology (Mata-Montero et al. 2015). This dataset has been made available online[1] as a contribution to the scientific community. Figure 3.1 shows a random sample taken from such dataset.

- PlantCLEF (PC): this dataset contains 1,000 species and 91,759 images for training and 21,446 images for testing (Goëau et al. 2015), all images from plants taken in the wild. This dataset is used for the PlantCLEF challenge from 2015. It is used in Chapter 7 and Chapter 11 for experimentation. Figure 3.2 shows some samples taken from such dataset.

- ImageNet (I): the ImageNet dataset is used only for parameter initialization in Chapter 7, Chapter 8, Chapter 10 and Chapter 11.

[1] http://otmedia.lirmm.fr/LifeCLEF/GoingDeeperHerbarium/

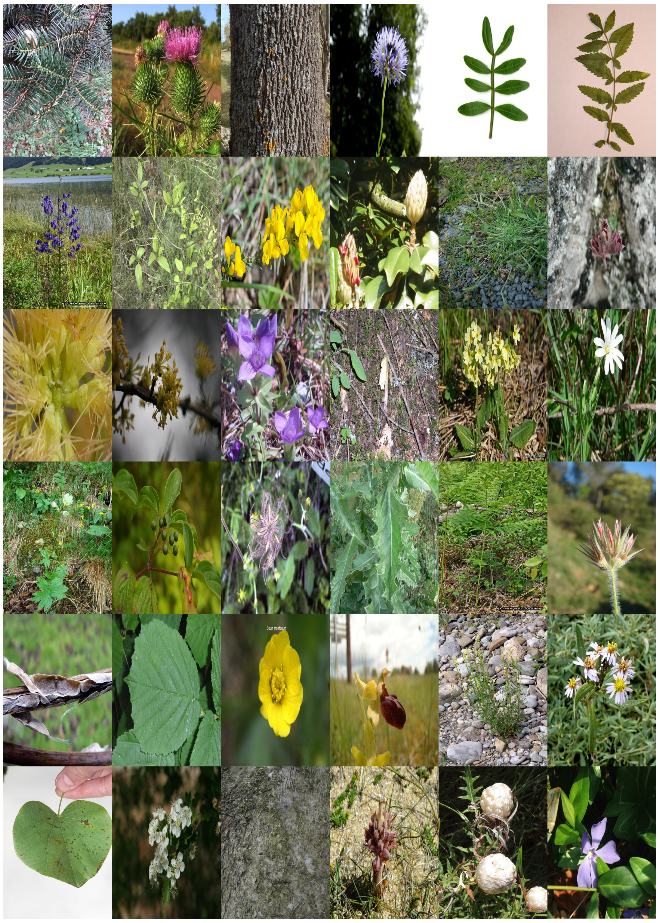


Figure 3.2. Random sample of in-situ plant images taken from the PlantCLEF (PC) dataset used in Chapter 7 and Chapter 11.

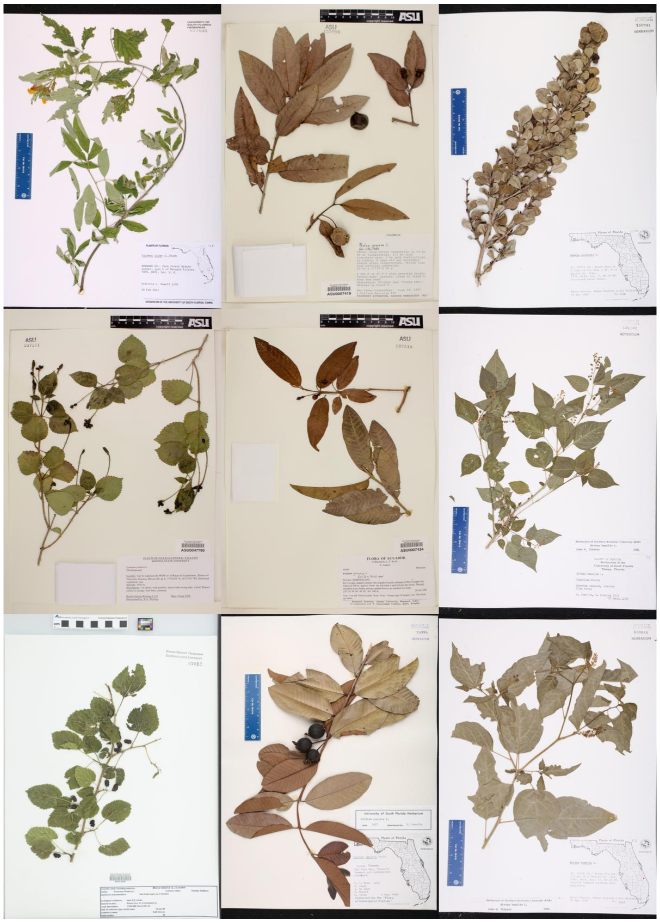


Figure 3.3. Random sample of different herbarium specimens taken from the Herbarium255 (H255) dataset used in Chapter 7.

2.2 Herbarium Sheets Datasets

One of the main contributions of this research is the use of herbarium data for automatic plant identification. At the time we published (Carranza-Rojas et al. 2017a,b), such type of datasets had not been used to identify plant species. Chapter 6 proposes the use of herbarium images as a new data source for automatic plant identification. We created two different datasets with herbarium images, one from Costa Rica and one from the Mediterranean region:

- Herbarium255 (H255): this dataset includes 255 species that match 213 of the species present in the CRLeaves (CR) dataset. It uses the iDigBio (*iDigBio* 2017) database and has a total of 11,071 images. This dataset is a new contribution to the scientific community. Figure 3.3 shows a sample of herbarium images taken from this dataset.

- Herbaria1K (H1K): this dataset comprises 1,225 species, which have an intersection with most of the 1,000 species of the PlantCLEF (PC) dataset. It contains 202,445 images for training and 51,288 for testing. All images have been resized to a width of 1,024 pixels and their height proportionally, given the huge resolutions available in herbarium images. This dataset is a contribution to the scientific community. It is used in the experimets described in Chapter 7, Chapter 8, Chapter 10 and Chapter 11.

2.3 Biases in Datasets

One important aspect of this research is to understand if there are any hidden biases in the plant datasets in order to avoid them. Chapter 9 contains experiments that show how leaf images of the same specimen can cause a bias in the accuracy results of both hand-crafted features as well as in Deep Learning approaches. Similarly, herbarium sheet datasets used in in Chapter 7, Chapter 8, Chapter 10, and Chapter 11 were separated in training and testing by taking the author of the herbarium sheet into account, so no same author is in both training and testing.

3 Traditional Hand-crafted Approaches

Initial experimentation on automatic plant identification based on images was done using hand-crafted features with the traditional Computer Vision pipeline, explained in Chapter 4. These techniques are prone to segmentation error and do not learn the feature extraction method by themselves, in contrast to Convolutional Neural Networks (CNNs). Chapter 4, as the first work of this research, uses this approach to measure accuracy per species. Later, Chapter 5 uses the same approach to measure if the leaf side is a discriminant factor for automatic plant identification. Finally, Chapter 9 uses hand-crafted feature as well as Deep Learning for experimentation on potential bias effects of the leaf datasets.

4 From Hand-crafted to Deep Learning

This research started in an inflection point where Machine Learning and Computer Vision practitioners started to switch to Deep Learning, more particularly to Convolutional Neural Network (CNN) approaches, from more traditional hand-crafted techniques. Chapter 6 covers this inflection point. This chapter defines several challenges and opportunities in the automatic plant identification domain. One of those, at the time, was to migrate to CNNs as they provided better results in more general image-based problems such as ImageNet (Russakovsky et al. 2015).

5 Deep Learning Models

After the publication of the work presented in Chapter 6 and the creation and availability of bigger datasets, Deep Learning became the de facto technology to be used in our work. We started building Convolutional Neural Networks (CNNs) that converged faster by using Batch Normalization and Parametric Rectified Linear Unit (PRELU). The model used for experimentation in Chapter 7, Chapter 8 and Chapter 10 was GoogleNet (Szegedy et al. 2015). In Chapter 11 we used ResNet (He et al. 2016). All these chapters used herbarium images as the main data source, which made a bigger impact in the scientific community, as they are the first studies to use such data for automatic plant identification.

5.1 Multi-level Hierarchical Architectures

Traditionally, plant identification has been done at species level. However, this research is also oriented to study Deep Learning models that can classify in a multi-level environment, meaning, to classify not only at species but at higher taxonomic levels. Chapter 10 covers the efforts towards building new CNN architectures that can classify species, genus and family at the same time, with the same set of parameters, instead of training three different, separated models. Chapter 10 compares the baseline accuracy obtained with three separated Flat Classification Model (FCM) (one for species, one for genera and one for families), versus two approaches that share the model parameters among the different classification tasks, namely, Multi-Task Classification Model (MCM) and TaxonNet. The last two model architectures allow a faster training period since all tasks can be classified at the same time.

5.2 Hierarchical Loss Functions

One of the main contributions of this research is the development of new loss functions that use prior knowledge of the plant taxonomy. We began the development of the Hierarchical Regularization in France, at Centre de Cooperation Internationale en Recherche Agronomique pour le Developpement (CIRAD) and Institut National de Recherche en Informatique et en Automatique (INRIA). The equations are explained in Chapter 11. These equations correspond to the loss function for the species only. This loss function can guide the model training to get slightly better accuracy results than using traditional cross entropy with softmax, and may open the possibility to create a new family of loss functions that make use of the class hierarchy at hand.

6 Software and Hardware

For all Deep Learning code Python was the language of choice. For previous approaches C++ was used, using mostly OpenCV. For the implementation of the new hierarchical loss function in Chapter 11, PyTorch was used as a baseline (Paszke et al. 2017), thanks to its autograd capabilities. This means that PyTorch will automatically calculate the derivatives of whole network graph, so if new operations are made using simpler operations (in our case, a new hierarchical loss function), their derivatives will be calculated automatically too. Experiments in Chapter 10 use Theano (Theano Development Team 2016). Caffe (Jia et al. 2014) was used in Chapter 7 for the herbarium sheets datasets experiments, by implementing a modified version of GoogleNet model (Szegedy et al. 2015). Chapter 8 serves as an extension for Chapter 7, were a deeper analysis is done to understand accuracy per species, genera and families.

All Deep Learning models were implemented on NVIDIA Graphics Processing Unit (GPU) using CUDA. In Particular, we made use of GTX 980 GPU, GTX 1070 GPU and Tesla K40 GPU kindly facilitated by the Costa Rica National High Technology Center (CeNAT).

Chapter 4

Combining Leaf Shape and Texture for Costa Rican Plant Species Identification

Reference Jose Carranza-Rojas and Erick Mata-Montero (2016a). "Combining Leaf Shape and Texture for Costa Rican Plant Species Identification". en. In: *CLEI Electronic Journal* 19, pp. 7 –7. ISSN: 0717-5000. URL: http://www.scielo.edu.uy/scielo.php?script=sci_arttext&pid=S0717-50002016000100007&nrm=iso

Keywords Biodiversity Informatics, Computer Vision, Image Processing, Leaf Recognition

1 Abstract

In the last decade, research in Computer Vision has developed several algorithms to help botanists and non-experts to classify plants based on images of their leaves. LeafSnap is a mobile application that uses a multiscale curvature model of the leaf margin to classify leaf images into species. It has achieved high levels of accuracy on 184 tree species from Northeast US. We extend the research that led to the development of LeafSnap along two lines. First, LeafSnap's underlying algorithms are applied to a set of 66 tree species from Costa Rica. Then, texture is used as an additional criterion to measure the level of improvement achieved in the automatic identification of Costa Rica tree species. A 25.6% improvement was achieved for a Costa Rican clean image dataset and 42.5% for a Costa Rican noisy image dataset. In both cases, our results show this increment as statistically significant. Further statistical analysis of visual noise impact, best algorithm combinations per species, and best value of k, the minimal cardinality of the set of candidate species that the tested algorithms render as best matches is also presented in this research.

2 Introduction

Plant species identification is fundamental to conduct studies of biodiversity richness of a region, inventories, monitoring of populations of endangered plants and animals, climate change impact on forest coverage, bioliteracy, invasive species distribution modelling, payment for environmental services, and weed control, among many other major challenges for biodiversity conservation. Unfortunately, the traditional approach used by taxonomists to identify species is tedious, inefficient and error-prone (Carvalho et al. 2007). In addition, it seriously limits public access to this knowledge and participation as, for instance, citizen scientists. In spite of enormous progress in the application of computer vision algorithms in other areas such as medical imaging, OCR, and biometrics (Andreopoulos et al. 2013), only recently have they been applied to identify organisms. In the last decade, research in Computer Vision has produced algorithms to help botanists and non-experts classify plants based on images of their leaves (Aggarwal et al. 2012; Arun et al. 2013; Beghin et al. 2010; Bhardwaj et al. 2013; Herdiyeni et al. 2012; Kadir et al. 2011; M. Z. Rashad 2011; R.D et al. 2011; Wijesingha et al. 2012; Wu et al. 2007). However only a few studies have resulted in efficient systems that are used by the general public, such as (Kumar et al. 2012). The most popular system to date is LeafSnap (Kumar et al. 2012). It was considered a state-of-the-art mobile leaf recognition application that uses an efficient multiscale curvature model to classify leaf images into species. LeafSnap was applied to 184 tree species from Northeast USA, resulting in a very high accuracy method for species recognition for that region. It has been downloaded by more than 1 million users (Kumar et al. 2012). LeafSnap has not been applied to identified trees from tropical countries such as Costa Rica. The challenge of recognizing tree species in biodiversity rich regions is expected to be considerably bigger.

Vein analysis is an important, discriminative element for species recognition that has been used in several studies such as (Clarke et al. 2006; Larese et al. 2014; Lee et al. 2013a,b; Li et al. 2006). According to Nelson Zamora, curator of the herbarium at the Instituto Nacional de Biodiversidad, Costa Rica (INBio), venation is as important as the curvature of the margin of the leaf when classifying plant species in Costa Rica (Zamora 2014).

This paper focuses on studying the accuracy of a leaf recognition model based not only on the curvature of the leaf margin, but also on its texture (in which veins are visually very important). This is the first attempt to create such model for Costa Rican plant species.

The rest of this manuscript is organized as follows: Section 3 presents relevant related work. Section 4 and Section 5 cover methodological aspects and experiment design, respectively. Section 6 describes the results obtained. Section 1 presents conclusions and, finally, Section 8 summarizes future work.

3 Related Work

In LeafSnap (Kumar et al. 2012) the authors create a leaf classification method based on unimodal curvature features and similarity search using kNN. This method is tested against an image dataset from North American trees, using 184 species in total. Since their system requires images to have a uniform background, leaf segmentation works by estimating the foreground and background color distributions, and then classifying each pixel at a time into one of those two categories. A conversion to HSV color domain is applied before using Expectation-Maximization (EM) (Dempster et al. 1977) for the leaf segmentation. A 96.8% of accuracy is reported by the authors on their dataset with $k = 5$.

Researchers in (Herdiyeni et al. 2013) use LBP features to classify medicinal and house plants from Indonesia. They extract LBP descriptors from different sample points and radius, calculate a histogram for each radius length feature set, and concatenate those histograms, similarly to HCoS of LeafSnap (Kumar et al. 2012). As a classifier, a four layer Probabilistic Neural Network (PNN) is used. Their dataset consists of two subsets; one comprises 1,440 images of 30 species of tropical plants, and the other one has 300 images of 30 house plant species. The image background of the medicinal plants is uniform, while house plant images have non-uniform backgrounds. For medicinal plants the reported precision is 77% and for house plants 86.67%, revealing that using LBP for complex image backgrounds is a suitable technique.

Authors in (Nguyen et al. 2013) use Speeded Up Robust Features (SURF) to develop an Android application for mobile leaf recognition. For the species classification task, SURF features are extracted from the gray scale image of the leaf. The feature set is reduced to histograms in order to reduce dimensionality since the resulting SURF feature vector may be too big. The precision reported is 95.94% on the Flavia dataset (Wu et al. 2007), which consists of 3,621 leaf images of 32 species.

4 Methodology

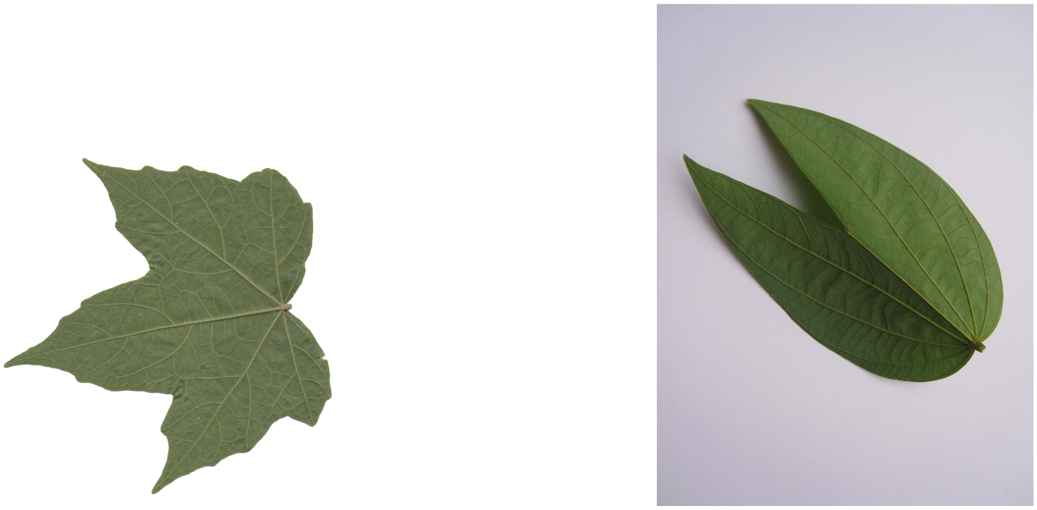
This section describes how the leaf recognition process was set up. Section 4.1 describes the image datasets used. Section 4.2 summarizes the techniques used to segment each image into leaf and non-leaf pixels clusters. Section 4.3 presents several image enhancements conducted, such as cleaning up undesirable artifacts and elements, stem removal, clipping and resizing. Section 4.4 describes the feature extraction approach for both the curvature and texture model. Finally, Section 4.5 presents the species classification metrics and algorithms used in this research.

4.1 Image Datasets

An image dataset of leaves from Costa Rica was created from scratch. To our knowledge, no other suitable Costa Rican datasets existed before. The dataset has both clean and noisy images, in order to identify how the amount of noise affects the algorithms. All images were captured from mainly two places: La Sabana Park, located in San Jose, and INBiopark, located in Santo Domingo, Heredia. In most cases, images for both surfaces of each leaf were taken. The dataset includes endemic species of Costa Rica and threatened species according to Zamora 2014. The complete list of species in the dataset can be found in (Carranza-Rojas 2014). The dataset consists of the following two subsets:

Clean Subset Fresh leaf images were captured during field trips to both La Sabana and INBiopark. If the leaves were not flat enough, a press was used to flatten them for 24 hours. A total of 1468 leaf images were scanned. The images have a white uniform background and a size of 2548x3300 pixels, scanned with 300 dpi in JPEG format. Photoshop CS6 was used to remove shadows, dust particles and other undesired artifacts from the background. Figure 4.1a shows a sample of a cleaned Costa Rican leaf image of this subset. The scanner used was an HP ScanJet 300.

Noisy Subset Fresh leaf images were captured during field trips to both La Sabana and INBiopark. No press was used to flatten them. A total of 2345 fresh leaf images were captured. This subset was captured against white uniform backgrounds (normally a sheet of paper). Each image has a 3000x4000 pixel resolution, in JPG format. No artifacts were removed manually. However as explained in Section 4.3 several automated image enhancements were performed both on the clean subset and the noisy subset. Figure 4.1b presents a noisy leaf image sample. The camera used is a Canon PowerShot SD780 IS.



(a) A *Robinsonella lindeniana var. divergens* sample scanned from a leaf sample from INBiopark, using a HP ScanJet 300 scanner, then cleaned using Photoshop CS6

(b) A *Bauhinia ungulata sample* taken using a Canon PowerShot SD780 IS camera at the Sabana Park

Figure 4.1. Collected image samples

4.2 Image Leaf Segmentation

The first step to process the leaf image is to segment which pixels belong to a leaf and which do not. We used the same approach as LeafSnap by applying color-based segmentation.

HSV Color Domain

When segmenting with color it is imperative to use the right color domains in order to exclude undesired noise. Kumar et al. 2012 states how, in the HSV domain, Hue had a tendency to contain greenish shadows from the original leaf pictures. Saturation and Value however, had a tendency to be clean. So we also used those two color components for leaf segmentation. Figure 4.2 shows the noise present in the Hue channel, but also shows how Saturation and Value are cleaner. This was useful for posterior segmentation using Expectation-Maximization (EM). We used OpenCV (Bradski 2000) to convert the original images into the HSV domain. Then, by using NumPy (Oliphant 2006) , we extracted the Saturation and Value components, which were fed to the EM algorithm.

Expectation-Maximization (EM)

Once images were converted to HSV and the desired channels were extracted, we applied EM to the color domain in order to cluster the pixels into one of 2 possible groups: leaf and non-leaf groups (Kumar et al. 2012). Figure 4.3 shows several samples of the final segmentation after applying EM. As shown, EM segments the image into the leaf and non-leaf pixel groups by assigning a 1 to the leaf pixels and a 0 to the non-leaf pixels. This method also works well on both simple and compound leaves. It is important to highlight that we did not assign weights to each cluster manually as the work done by (Kumar et al. 2012), because we wanted to leave the process as automatic as possible. In their work, they improve the segmentation of certain

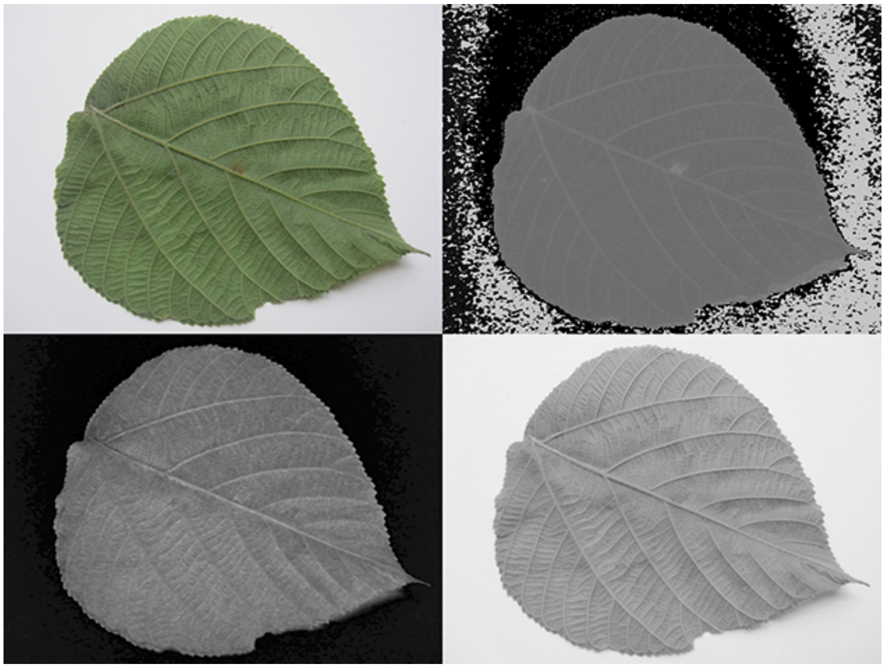


Figure 4.2. HSV decomposition of a leaf image
The top-left image shows the original sample. The top-right image shows the Hue channel of the image with noticeable noise. The bottom-left image shows the Saturation component and the bottom-right image shows the Value component

types of leaves, especially skinny ones, by manually assigning different weights to each cluster. Weights play a fundamental role into the segmentation process as reported in (Zhu et al. 2014).

Training Algorithm 1 describes the process to train the EM algorithm. We used OpenCV's implementation of EM. First we stacked all the pixels of the image matrix into a single vector. Then we trained the model using a diagonal matrix as a co-variance matrix, and we assigned two clusters to it, which internally were translated into two Gaussian Distributions, one for the leaf cluster and one for the non-leaf cluster. Once trained, we returned the EM object.

Algorithm 1 EM Training

```
$stackedPixels \leftarrow \emptyset$
for all $pixelRow$ in image do
    for all $pixel$ in pixelRow do
        $stackedPixels \leftarrow stackedPixels \cup pixel$
    end for
end for
$EM \leftarrow OpenCV.EM(nClusters = 2, covMatType = OpenCV.DIAGONAL)$
$EM.train(stackedPixels)$
return $EM$
```

Pixel Prediction Algorithm 2 explains how the owning cluster of a single pixel of the image was predicted. Once the EM object was trained, the OpenCV's implementation allowed to compute the probabilities of the pixel belonging to each cluster. However, for more efficiency, we created a dictionary containing each unique $(Saturation,Value)$ pair as key, and the cluster as value. If the key was not found in the dictionary, we

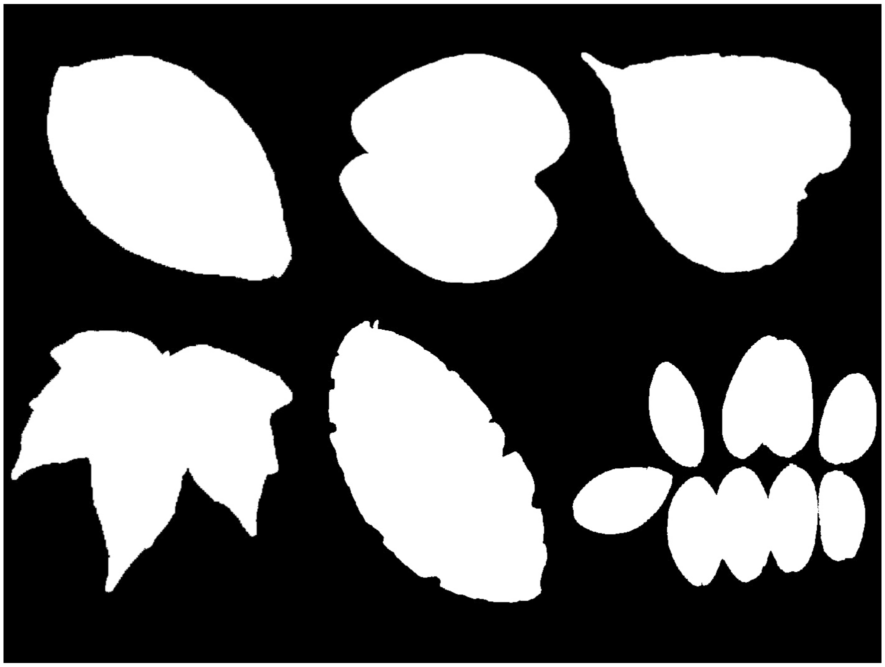


Figure 4.3. Segmented Samples
After applying EM to different Costa Rican species

then proceeded to predict the probabilities for each cluster, added the key and cluster to the dictionary, and returned the associated cluster with the biggest probability.

Algorithm 2 EM Pixel Prediction

```
$key \leftarrow hash(pixel[S], pixel[V])$
if $hash$ in pixelDictionary then
    return $pixelDictionary[key]$
end if
$probabilities \leftarrow EM.predict(pixel[S], pixel[V])$
$pixelDict[key] = probabilities[0] > probabilities[1]$
return $pixelDict[key]$
```

4.3 Image Enhancements/Post-Processing

After segmentation of the leaf using EM, some extra work was needed to clean up several false positives areas. We followed the process of LeafSnap (Kumar et al. 2012). First of all, each image was clipped to the internal leaf size provided by the segmentation. Then the image was resized to a common leaf area, followed by a heuristic applied to delete undesired objects. Finally, the stem was deleted since it added noise to the model of curvature (not that much to the texture model).

Clipping

Before extracting features, a clipping phase was needed in order to resize the region where the leaf was present to a common size. The clipping algorithm was trivial to implement once the contours were

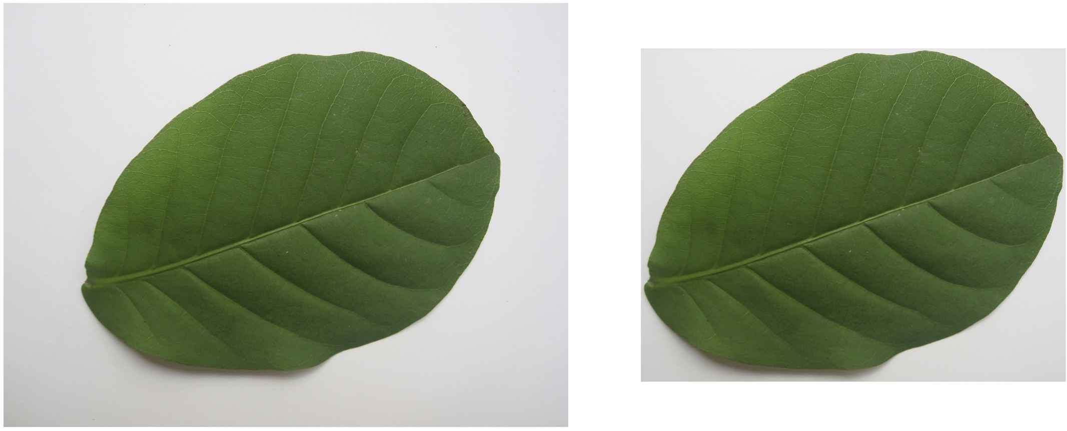


Figure 4.4. Clipping of a Coccoloba floribunda sample
The left image is the original leaf image, and the right one is clipped to the leaf size

calculated using OpenCV. As shown in Algorithm 3, the minimum and maximum coordinates were calculated for all contour x and y components, followed by a cut of the leaf image matrix to those resulting minimum and maximum coordinates. The ϵ was used to allow posterior algorithms ignore false positives regions that intersect the border. The results of the Clipping phase can be seen in Figure 4.4.

Algorithm 3 Clipping Leaf Portion of the Image

```
$xmin \leftarrow min(contours.xs) - \epsilon$
$ymin \leftarrow min(contours.ys) - \epsilon$
$xmax \leftarrow max(contours.xs) + \epsilon$
$ymax \leftarrow max(contours.ys) + \epsilon$
$clipped \leftarrow image[xmin : xmax, ymin : ymax]$
```

Resizing Leaf Area

Once the leaf area had been clipped, a resize was applied in order to standardize the leaf areas inside all images. If not, the model of curvature would be affected negatively since the amount of contour pixels varied significantly (Kumar et al. 2012). Our implementation of the resize was applied to the whole clipped image. Images may end up having different sizes, but the internal leaf areas were the same or almost the same. Algorithm 4 shows how a new width and height were obtained by calculating the ratio between the current leaf area, the desired new leaf area, and the current height and width of the image. Finally, OpenCV was used to resize the clipped image to a constant leaf size of $100{,}000$ pixels. This number was used empirically based on LeafSnap's original dataset resolution and the internal regions associated with leaf pixels. This approach means that the absolute measures of leafs are lost.

Algorithm 4 Common Leaf Area Resize

```
$newLeafArea \leftarrow 100000$
$imgArea \leftarrow height \times weight$
$newImgArea \leftarrow (imgArea \times newLeafArea)/leafArea$
$wGrowth \leftarrow weight/height + weight$
$hGrowth \leftarrow height/height + weight$
$a \leftarrow wGrowth \times hGrowth$
$x \leftarrow abs(\sqrt{4 \times a \times newImgArea}/(2 \times a))$
$newWidth = wGrowth \times x$
$newHeight = hGrowth \times x$
return $OpenCV.resize(image, newWidth, newHeight)$
```

Deleting Undesired Objects

Even when uniform background images were used, initial segmentation turned out not to be enough when the image contained undesired objects, such as dust, shadows, among others. (Kumar et al. 2012) attempted to delete these noisy objects by using the same heuristic we implemented as shown in Algorithm 5. By using Scikit-learn (Pedregosa et al. 2011) we calculated the connected components of the segmented image. We deleted the "small" components by area (in pixels). Small components were normally dust, small bugs or pieces of leaves, among other things. Once all small components were deleted, if the remaining was only one then we took that to be the leaf. If more than one component remained, then we calculated for each remaining component how many pixels had intersections with the image margin. We then deleted the component with the biggest number of intersections. The thinking behind this is to get rid of components that were not centered on the image, which tend to be non-leaf objects. Finally, the component with the biggest area from the remaining components was taken as the leaf.

Algorithm 5 Deleting Undesired Objects Heuristic

```
$n, components \leftarrow connectedComponents(segmentedImage)$
$components \leftarrow deleteSmallComponents(components, kMinimumArea)$
if $size(components) == 1$ then
    return $components[0]$
end if
$inters \leftarrow empty$
$areas \leftarrow empty$
for all $component$ in $components$ do
    $inters \leftarrow inters \cup getImageMarginIntersections(component)$
    $areas \leftarrow areas \cup getComponentArea(component)$
end for
$noisyObject \leftarrow max(inters)$
return $max(areas - noisyObject)$
```

Deleting the stem

We followed the approach for stem deletion described in (Kumar et al. 2012). If the stem was left intact, it would add noise to the model of curvature, given all the possible sizes the stem may take. Algorithm 7 shows the procedure. First, a Top Hat transformation was applied to the segmented image in order to leave only possible stem regions, as shown in Figure 4.5. Then all connected components were calculated from the Top Hat transformed image, and also their quantity. Then we looped over all the components, deleting every single one from the original segmentation and recalculating the new number of connected components. If the original number of recalculated connected components did not change upon deletion, that meant the current component was a good stem candidate (heuristically, a stem does not affect how many original connected components there are). Once all stem candidates were calculated, the one with the biggest area and largest aspect ratio was chosen to be the stem, as described in Algorithm 6.

Algorithm 6 Calculate Aspect Ratio Combined with Area

```
$width, heigth \leftarrow calculateRectangleAround(component)$
$area \leftarrow calculateArea(component)$
return $width/heigth * area$
```

Algorithm 7 Deleting the Stem

```
$candidates \leftarrow empty$
$candidatesRatios \leftarrow empty$
$possibleStemsImage \leftarrow topHatTransformation(segmentedImage)$
$n, components \leftarrow connectedComponents(possibleStemsImage)$
for all $component$ in $components$ do
    $tempSegmentation \leftarrow delete(component, segmentedImage)$
    $currentN \leftarrow connectedComponents(tempSegmentation)$
    if $currentN = n$ then
        $candidates \leftarrow candidates \cup component$
        $candidatesRatios \leftarrow candidatesRatios \cup calculateAspectRatio(component)$
    end if
end for
$bestCandidate \leftarrow candidates[max(candidatesRatios).index]$
$segmentedImage \leftarrow delete(bestCandidate, segmentedImage)$
```

4.4 Leaf Feature Extraction

Feature extraction was designed and implemented considering three main design goals:

- Efficiency: algorithms should be fast enough to support future mobile apps.
- Rotation invariance: the leaf may be rotated by any angle within the image.
- Leaf Size Invariance: datasets contain different sizes of leaves and users can capture images independently of the relative size of leaves.

Two different feature sets were calculated. The first one captures information about the contour of the leaf, while the second one captures information about its texture. Section 4.4 describes how we implemented Histogram of Curvature over Scale (HCoS) (Kumar et al. 2012) to extract contour information. Section 4.4 describes how we implemented Local Binary Pattern Variance (LBPV) to extract texture information. Both models generate histograms that are suitable for distance metric calculations.

Extracting contour information (HCoS)

The model of curvature used by LeafSnap comprises several steps. Previously explained segmentation and post-processing resulted in a mask of leaf and non-leaf pixels. The non-leaf pixels have values of 0, and the leaf pixels have values of 1. First, the different contour pixels were found, then 25 different masks with disk shapes were applied on top of each contour point, providing both an area of the intersection and an arc length. Then all calculations at each scale were turned into a histogram, resulting in 25 different histograms per image, one per scale. Finally, the 25 resulting histograms were concatenated, conforming the HCoS.

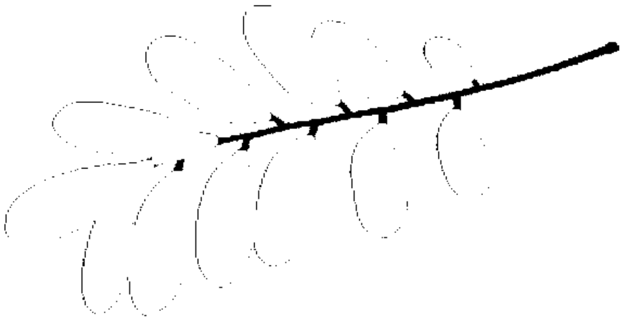


Figure 4.5. Top Hat Transformation applied to a segmented compound leaf image to detect the stem of the leaf

Contours On a binary image (resulted from the previous segmentation), the OpenCV implementation of contour finding worked very well, based on the original algorithm of Suzuki et al. 1985 for contour finding. The algorithm generated in a vector of pairs (x,y) that represent the coordinates where a contour pixel was found. A contour pixel can be defined as a pixel which is surrounded by at least another pixel with the opposite color of it. Figure 4.6 shows in red the contour pixels detected in the original image, calculated from the segmented mask. Notice how shadows affect the contour algorithm, since they were not segmented perfectly.

Scales The original algorithm of Kumar et al. 2012 makes use of 25 different scales, creating one disk per scale. We implemented a discrete version of the disks making use of matrices based on (Manay et al. 2006), whose code is available in Matlab [1].

The disks used are actually matrices of 1's and 0's. They were applied as masks over specific parts of the segmented leaf image (mostly contour points). The idea was to count how many pixels intersected the segmented image and each disk mask. We created two different types of disks. The first type is filled up with 1's, as shown in Figure 4.7a. It is used to measure the area of intersection. The second type is more like a ring, where 1's are present only in the circumference of the disk (see Figure 4.7b). It is used to determine the arc's length of the intersection of the disk with the leaf, at a given contour point.

Once all disks were created for both area and arc length versions, we applied them to each pixel of the contour vector, as shown by Algorithm 8.

Figure 4.8 shows how one specific area disk was applied to the segmented image, for an specific scale (radius=18 in this case), at a given contour pixel. The gray area shows the intersection of pixels with the leaf segmentation. This procedure was then repeated over all the pixels from the contour vector in the same way.

Histograms Using NumPy at each scale, a histogram was created from all the values generated from all contour pixels, as described by Algorithm 8. We used histograms of 21 bins, as Kumar et al. 2012 did. This means a total of 25 different histograms were created, each with 21 bins, per image. At each scale, each

[1] `https://www.ceremade.dauphine.fr/~peyre/numerical-tour/tours/shapes_4_shape_matching/`

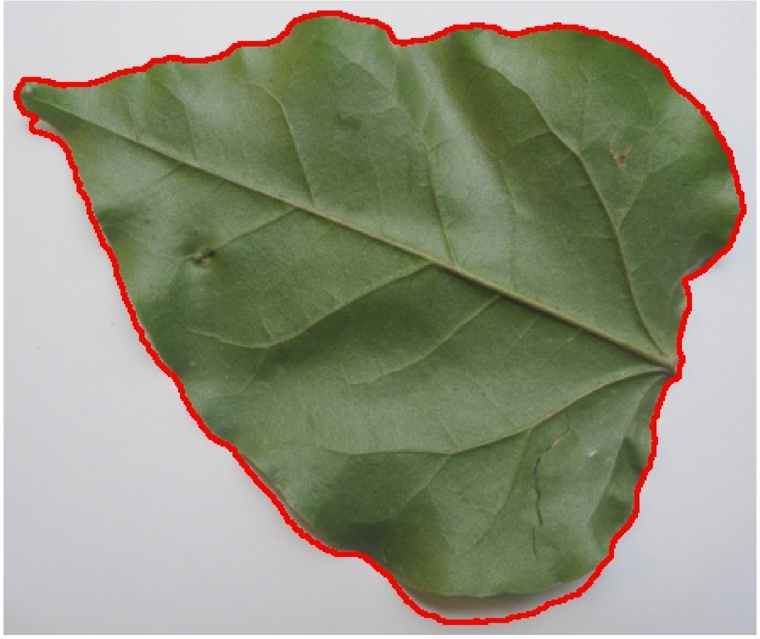


Figure 4.6. *Croton niveus* contours extracted using OpenCV

Algorithm 8 Area and Arc Length Vector Calculation

```
$arcs \leftarrow empty$
$areas \leftarrow empty$
**for all** $pixel$ of the contour vector **do**
    **for all** $areaMask, arcMask = 1$ to $25$ **do**
        center $areaMask, arcMask$ at current contour $pixel$
        $area \leftarrow count(areaMask \cap segmentation)$
        $areas \leftarrow areas \cup area$
        $arc \leftarrow count(arcMask \cap segmentation)$
        $arcs \leftarrow arcs \cup arc$
    **end for**
**end for**
```

$$\begin{matrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{matrix}$$

(a) A filled disk of *radius*=2 pixels used to calculate area of the intersection with the leaf segmentation

$$\begin{matrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{matrix}$$

(b) An unfilled disk (ring) of *radius*=2 *pixels* used to calculate arc length of the intersection with the leaf segmentation

Figure 4.7. Various discrete disks

Table 4.1. Variants of LBPV

Variant	Radius	Pixels
R1P8	1	8
R2P16	2	16
R3P16	3	16
R1P8 & R2P16	1 and 2	8 and 16
R1P8 & R3P16	1 and 3	8 and 16
R3P24	3	24

histogram was normalized to unit length. Then, all histograms were concatenated together (both the 25 for area and 25 for arc length), generating what Kumar et al. 2012 describes as the Histogram of Curvature over Scale (HCoS).

Extracting texture information (LBPV)

We aimed at improving the model of curvature by adding texture analysis. We used a Local Binary Pattern Variance (LBPV) implementation called Mahotas (Coelho 2013) that is invariant to rotation, multiscale, and efficient. This implementation of LBPV is based on the algorithm of Ojala et al. 2002 and makes use of NumPy libraries to represent the image and the resulting histograms. It works on gray images, so we used OpenCV to convert the Red Green Blue (RGB) images to gray scale images. The LBPV approach detects micro structures such as lines, spots, flat areas, and edges (Ojala et al. 2002). This is useful to detect patterns of the veins, areas between them, reflections, and even roughness. Figure 4.9 shows what two different LBPV implementations look like. The upper image shows a $radius = 2, pixels = 16$ (R2P16) implementation, and the one below shows a $radius = 1, pixel = 8$ (R1P8) pixel implementation. The different variants of the LBPV used are shown in Table 4.1. In some cases we concatenated two histograms of different scales such as R1P8 & R2P16. It is important to note that we did not use the variant which samples 24 pixels, since it generated too large histograms. We did, however, run some tests in which we noticed the 24 pixels variation didn't add more accuracy, so we decided to ignore this method.

Just like the HCoS, LBPV generates histograms that can be used for similarity search. Several histograms

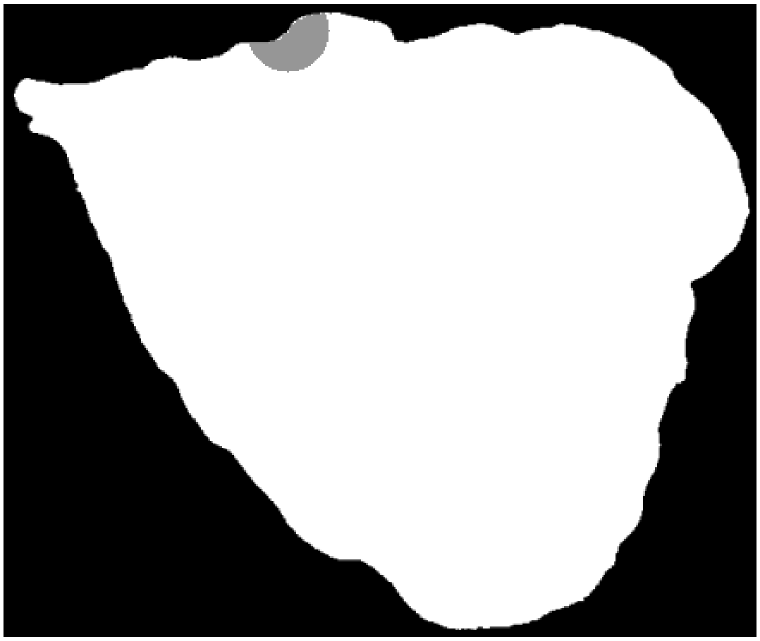


Figure 4.8. Area disk applied
to a *Croton niveus* sample at an specific pixel of the contour, with radius=18

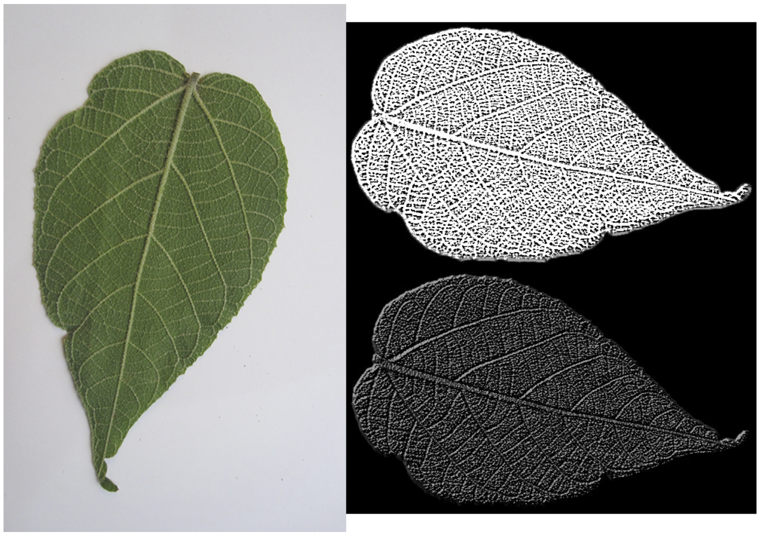


Figure 4.9. LBPV patterns of a *Croton draco* sample. The upper image corresponds to a $radius = 2, pixels = 16$ (R2P16) and the lower one to a $radius = 1, pixels = 8$ (R1P8) pattern

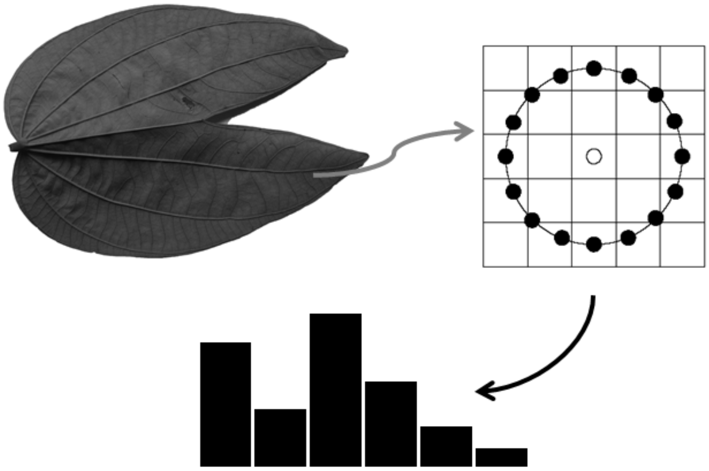


Figure 4.10. Process of extracting LBPV

were generated at different radius sizes and different circumference pixel sampling, in order to validate which combinations provided the best results. The Mahotas implementation returned a histogram of the feature counts, where position i corresponds the count of pixels in the leaf texture that had code i. Also, given that the implementation is a LBPV, non-uniform codes are not used. Thus, the bin number i is the $i - th$ feature, not just the binary code i (Coelho 2013). Figure 4.10 describes at a very high level how the process of extracting the local patterns histograms works. First, the image is converted to a gray scale image. Then, for each pixel inside the segmented leaf area, we calculated the local pattern with different radius and circumference using the mahotas implementation. Finally, each pattern was assigned to a bucket in the resulting histogram. Each pixel has a number assigned to it corresponding to a pattern, and the histogram was created using all those numbers from the segmented leaf pixels.

4.5 Species Classification based on Leaf Images

Once all histograms were ready and normalized, a machine learning algorithm was used to classify unseen images into species. We implemented the same classification scheme used by LeafSnap. The following paragraphs describe how k Nearest Neighbors (kNN) was implemented.

Scikit-learn's kNN implementation was used for leaf species classification. This process was fed with previously generated histograms from both the model of curvature using HCoS and the texture model using LBPV. Additional code was created to take into consideration only the first matching k species, not the first k images, as shown by Algorithm 9. The difference resides in taking into account only the best matching image per species, until completing the first k species (Kumar et al. 2012).

Algorithm 9 k Species Ranking

```
$neighborImages, distances \leftarrow knnSearch(histogram, k)$
$resultSpecies \leftarrow empty$
while each $neighborImage\ and\ k > 0$ do
    if not $neighborImage.species$ in $resultSpecies$ then
        $resultSpecies \leftarrow resultSpecies \cup neighborImage.species$
        $k \leftarrow k - 1$
    end if
end while
```

We used $1 <= k <= 10$ in order to measure how different algorithms behaved as the value of k increased.

4.6 Distance Metric - Histogram Intersection

We tested the basic Euclidean distance to measure similarity between histograms, however the results were not encouraging. We implemented the histogram intersection shown on Equation 4.1, where $I(x,y)$ is the histogram intersection between a histograms x and y of same size, n is the number of bins, and x_i and y_i are each a bin in histograms x and y, respectively. This distance metric is also normalized to unit length.

$$I(x,y) = \sum_{i=1}^{n} x_i - \sum_{i=1}^{n} min(x_i, y_i) \tag{4.1}$$

4.7 Accuracy

Let E be an identification experiment that consists of a model M, a set S that contains n images of leaves of n (not necessarily different) unknown tree species to be identified, and an integer value k, $k \geq 1$. We define $hit(M,k,x)$ as a boolean function that indicates if model M generates a ranking in which one of the top k candidate species is a correct identification of sample x. Equation 5.1 formally defines $Accuracy(M,S,k)$.

$$Accuracy(M,S,k) = \sum_{x \in S} \frac{hit(M,k,x)}{n} \tag{4.2}$$

5 Experiments

Several model variations were used in the experiments (see Table 4.2).

1. Our implementation of LeafSnap's model of curvature HCoS.
2. Several scales of the texture model based on LBPV.
3. The combination of HCoS and the best LBPV variant, which according to our tests was R1P8 & R3P16. This combination was further disaggregated by assigning different weights to HCoS and the texture model.

One Versus All One approach to test a model is to partition a dataset into two datasets: one for training and one for testing. Another approach is to use One versus All, that is, each image in a dataset with n elements is considered a test image and the remaining $n - 1$ images the training subset. We used both approaches as explained at the end of this section.

Combining Curvature and Texture When combining two different models, we faced the issue of having different scales in the resulting ranking of each model. This was resolved by normalizing the rankings to unit length.

After normalizing the rankings (one per combined algorithm), we assigned a factor to each combined model in order to rank the predicted species into a single ranking. This factor sums 1 in total. However we

Table 4.2. Models used in the experiments including curvature, variants of texture model, and combination of both

Model Name	Description	Type
HCoS	25 scales, 21 bins per scale	Curvature
R1P8	$radius = 1, pixels = 8$	Texture
R2P16	$radius = 2, pixels = 16$	Texture
R3P16	$radius = 3, pixels = 16$	Texture
R1P8 & R2P16	$radius = 1, pixels = 8$ & $radius = 2, pixels = 16$	Texture
R1P8 & R3P16	$radius = 1, pixels = 8$ & $radius = 3, pixels = 16$	Texture
HCoS & R1P8 & R3P16	Assigned a factor to curvature and texture. Factors summed 1, increasing by 0.10	Curvature and Texture

varied the factor associated with each model to see the behavior across different combinations. We used factors of (0.10, 0.90), (0.20, 0.80), (0.30, 0.70), (0.40, 0.60), (0.50, 0.50), (0.60, 0.40), (0.70, 0.30), (0.80, 0.20), (0.90, 0.10). For example, $(0.50, 0.50)$ means we gave the same level of importance to each model on that combination. Algorithm 10 describes how the merge between two methods was achieved.

Algorithm 10 Combining Two Rankings

```
$combinedRanking \leftarrow \emptyset$
$FACTORS \leftarrow \{0.10, 0.20, 0.30, 0.40, 0.50, 0.60, 0.70, 0.80, 0.90\}$
for all $factor$ in $FACTORS$ do
    $results \leftarrow empty$
    for all $species$ in $allSpecies$ do
        $distance1 \leftarrow resultsAlgorithm1[species]$
        $distance2 \leftarrow resultsAlgorithm2[species]$
        $results[species] \leftarrow (distance1 * factor) + (distance2 * (1 - factor))$
    end for
    $combinedRanking[factor] \leftarrow TakeBestKDistances(results)$
end for
```

5.1 Texture and Curvature Model Experiments

We ran all models M described in Table 4.2, with $1 \leq k \leq 10$, and the following data sets: Costa Rica clean subset (One versus All, $n = 1468$), Costa Rica noisy subset (One versus All, $n2345$), and Costa Rica complete data set (training set with all 1468 clean images and testing set with all 2345 noisy images). In each experiment, $Accuracy(M, S, k)$ was calculated for the corresponding dataset S. In addition, for model HCoS & R1P8 & R3P16, Algorithm 10 was used to comprehensively consider different weight combinations for HCoS and the texture model. Table 4.5 summarizes the results obtained.

5.2 Processing Times

To understand the duration of the recognition process, we measured the recognition time for all images from both Costa Rican noisy and clean subsets, as if a back-end received images from a mobile app. The

Table 4.3. Factors and levels for GLM per species

Factor	Number of Levels	Levels
Algorithm	5	R1P8 & R3P16 0.1 HCoS and 0.9 R1P8 & R3P16 0.5 HCoS and 0.5 R1P8 & R3P16 0.9 HCoS and 0.1 R1P8 & R3P16 HCoS
Noise is worse?	2	Yes, No
k	10	1,2,3,4,5,6,7,8,9,10

measured time includes image loading, segmentation, stem deletion, normalization, curvature calculations, texture calculations, and similarity search. It does not include network related times. We used a MacBook Pro with an Intel Core i7, 2.8 GHz, and 8 GBs on RAM.

5.3 Statistical Analysis For Noise Affectation, Best Algorithms per Species, and best value $\hat{k}$

Using the clean and noisy datasets, we calculated a General Linear Model (GLM) per species over a total of 65 species. We aimed at discovering the following:

- What is the minimum value of k that provides results statistically equivalent to those obtained when $k = 10$ for each species? Obviously, accuracy increases as the value of k increases. However, for practical reasons, we would like to test if there is a threshold value $\hat{k}$ after which accuracy remains statistically equivalent to using $k = 10$. For example, in a mobile app users would appreciate if the number of best ranked species is not the maximum value 10, but a smaller number.

- What is the best algorithm or combination of algorithms for each species? For this we used five different algorithms: R1P8 & R3P16 (texture alone), 0.1 HCoS and 0.9 R1P8 & R3P16, 0.5 HCoS and 0.5 R1P8 & R3P16, 0.9 HCoS and 0.1 R1P8 & R3P16 and HCoS (curvature alone). This also includes creating clusters of species based on their most significant algorithms, and understanding the clusters with more species and best accuracies.

- Does noise decrease the accuracy level obtained per species? Can we find some species that are not affected by noise in the data?

To achieve this, we calculated a GLM per species to detect significance of noise, algorithm used and value of k. We used a confidence level of 0.95. Once each GLM was calculated and each main effect significance known and proven, we calculated if all levels within each factor were statistically equivalent. We are actually trying to find the levels that are significantly different, for all three factors. We used a Tukey statistical test for each factor. Table 4.3 shows the different factors and levels used during this experiment.

Table 4.4. Other studies comparison of obtained results on the Flavia dataset

Study	Features	Classifier	Precision	Accuracy
Nguyen et. al. (Nguyen et al. 2013)	SURF	SVM	0.959	-
Lee et. al. (Lee et al. 2013a)	Fast Fourier Transform (FFT)	Centroid	-	0.9719
S.Wu et. al.(Wu et al. 2007)	Morphological Features	PNN	0.859	-
Kadir et. al. (Kadir et al. 2011)	Morphological, Color Features, FFT	PNN	-	0.9375
Lagerwall et. al. (R.D et al. 2011)	Morphological Features	Euclidean Distance	-	0.919
Mouine et. al. (Mouine et al. 2013)	Triangle Side Lengths and Angle (TSLA)	kNN, k=1	0.69	-
Our Texture Model	LBPV R1P8 & R3P1	kNN, k=1	-	0.892
Our Texture Model	LBPV R1P8 & R3P1	kNN, k=5	-	0.98
Our Texture Model	LBPV R1P8 & R3P1	kNN, k=10	-	0.985
Our HCoS Implementation	HCoS	kNN, k=1	-	0.371
Our HCoS Implementation	HCoS	kNN, k=5	-	0.697
Our HCoS Implementation	HCoS	kNN, k=10	-	0.813
0.5 HCoS and 0.5 R1P8 & R3P1	HCoS and LBPV R1P8, R3P16	kNN, k=10	-	0.991

5.4 Statistical Analysis of Best Algorithms for $k = 5$

Because $k = 5$ has become an informal benchmarking value in other research (Kumar et al. 2012), it is important to discover what algorithms got the best accuracy when $k = 5$. For this experiment, we ran a Binary Logistic Regression and optimized it, thus maximizing the probability of a successful identification. Based on the resulting regression model, we calculated the two best algorithms for both noisy and clean factors, and $k = 5$, per species.

6 Results

6.1 Comparison with Others Studies

In order to set a baseline, several other studies have used the Flavia dataset for their research (Wu et al. 2007). Table 4.4 shows the comparison of these studies and our approaches. Some studies do not report accuracy but precision only. The best accuracy of our work was achieved, on this dataset, by adding 0.5 HCoS and 0.5 R1P8 & R3P1 with $k = 10$ for a 0.991. We also attempted to use texture only, which shows to be very extremely accurate with up to 0.98. This dataset has been, however, artificially cleaned, so other studies should be evaluated on more complex datasets.

Table 4.5. Accuracy obtained when combining curvature and texture over the clean subset, the noisy subset, and the complete Costa Rican dataset

	Clean				Noisy				All			
		HCoS=a, R1P8.R3P16=b				HCoS=a, R1P8.R3P16=b				HCoS=a, R1P8.R3P16=b		
k	HCoS	a=0.1	a=0.5	a=0.9	HCoS	a=0.1	a=0.5	a=0.9	HCoS	a=0.1	a=0.5	a=0.9
		b=0.9	b=0.5	b=0.1		b=0.9	b=0.5	b=0.1		b=0.9	b=0.5	b=0.1
1	0.311	0.567	0.563	0.386	0.151	0.519	0.320	0.177	0.070	0.145	0.120	0.084
2	0.446	0.702	0.702	0.520	0.225	0.638	0.435	0.257	0.119	0.209	0.178	0.133
3	0.535	0.766	0.785	0.610	0.277	0.701	0.515	0.311	0.148	0.252	0.216	0.165
4	0.587	0.816	0.822	0.668	0.325	0.750	0.574	0.364	0.176	0.295	0.251	0.201
5	0.631	0.857	0.854	0.706	0.364	0.783	0.616	0.408	0.204	0.326	0.277	0.224
6	0.674	0.875	0.881	0.748	0.399	0.810	0.660	0.455	0.228	0.350	0.304	0.249
7	0.710	0.890	0.909	0.779	0.435	0.830	0.692	0.484	0.253	0.377	0.328	0.277
8	0.740	0.903	0.924	0.812	0.470	0.844	0.721	0.516	0.273	0.400	0.353	0.299
9	0.768	0.918	0.937	0.832	0.496	0.858	0.744	0.546	0.295	0.417	0.371	0.320
10	0.790	0.931	0.945	0.845	0.521	0.872	0.771	0.574	0.318	0.439	0.393	0.336

6.2 Texture and Curvature Model Experiments

Clean Subset As shown in Table 4.5, the best results were obtained when $k = 10$ and the model is 0.5 HCoS and 0.5 R1P8 & R3P16. The resulting accuracy is 0.945, in contrast with the accuracy of HCoS which is 0.79. Notice however that 0.5 HCoS and 0.5 R1P8 & R3P16 is also the best for all values of $6 <= k <= 10$. For $1 <= k <= 5$, 0.5 HCoS and 0.5 R1P8 & R3P16 and 0.1 HCoS and 0.9 R1P8 & R3P16 have very similar levels of accuracy. Figure 4.11a more clearly depicts these comparisons.

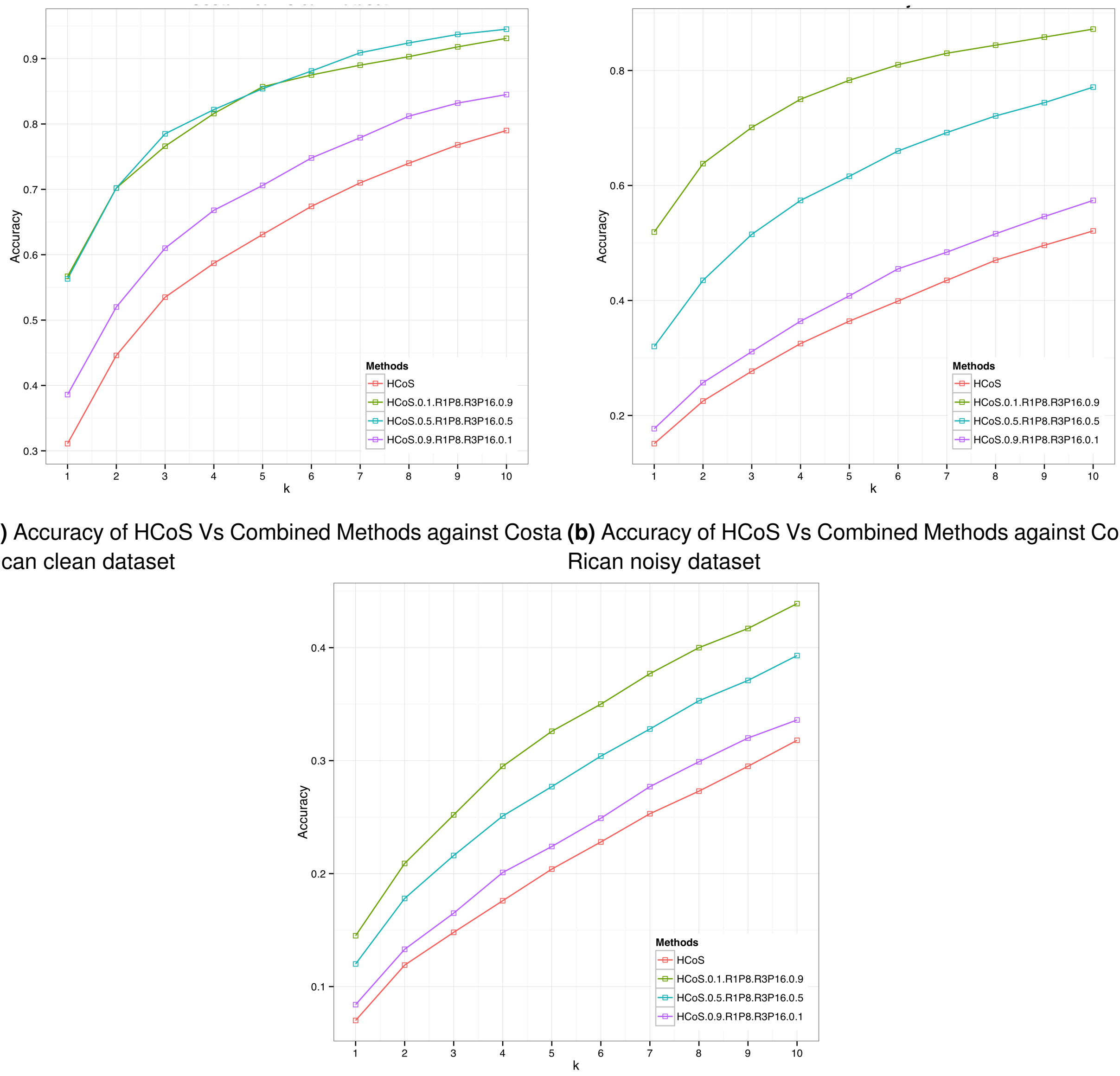
Noisy Subset Figure 4.11b clearly shows that 0.1 HCoS and 0.9 R1P8 & R3P16 has the best accuracy for all values of k. In addition, the level of accuracy improvement with respect to HCoS is considerably larger, ranging from 35.2% when $k = 10$ to 42.5% when $k = 4$ as shown in Table 4.7.

Complete Dataset As Figure 4.11c shows, the level of accuracy is considerably lower for all models, as compared to the previous two experiments. Even the best model achieves levels of accuracy in a poor $[14.5\%, 43.9\%]$ range.

Discussion These experiments show how, in general, the combination of HCoS and LBPV consistently increases the accuracy of HCoS alone. Accuracy declines as the combination factor assigned to curvature reaches 1. Overall, the best combination seems to be 0.1 HCoS and 0.9 LBPV. It is also important to notice how the accuracy is sensitive to the quality of the dataset. The clean subset has a tendency to improve the recognition accuracy, in contrast with the noisy subset. This reflects the importance of good pre-processing and good segmentation. Shadows, dust, and other artifacts affect the final accuracy results.

6.3 Measuring Significance of the Accuracy Increase

As shown in the previous section there is an increase in accuracy when texture is added to our implementation HCoS. This, however, may not be statistically significant. We proceeded then to apply a Statistical Proportion Test for Two Samples. Our null hypothesis $H0$ is that the accuracy of the implementation of HCoS equals the ones obtained by combining curvature and texture. In contrast, our alternative hypothesis $H1$ is that the accuracy of the implementation of HCoS is less than the combinations.



(a) Accuracy of HCoS Vs Combined Methods against Costa Rican clean dataset

(b) Accuracy of HCoS Vs Combined Methods against Costa Rican noisy dataset

(c) Accuracy of HCoS Vs Combined Methods against the complete Costa Rican dataset. Clean subset used for training and noisy subset for testing

Figure 4.11. Comparison of HCoS and Combinations

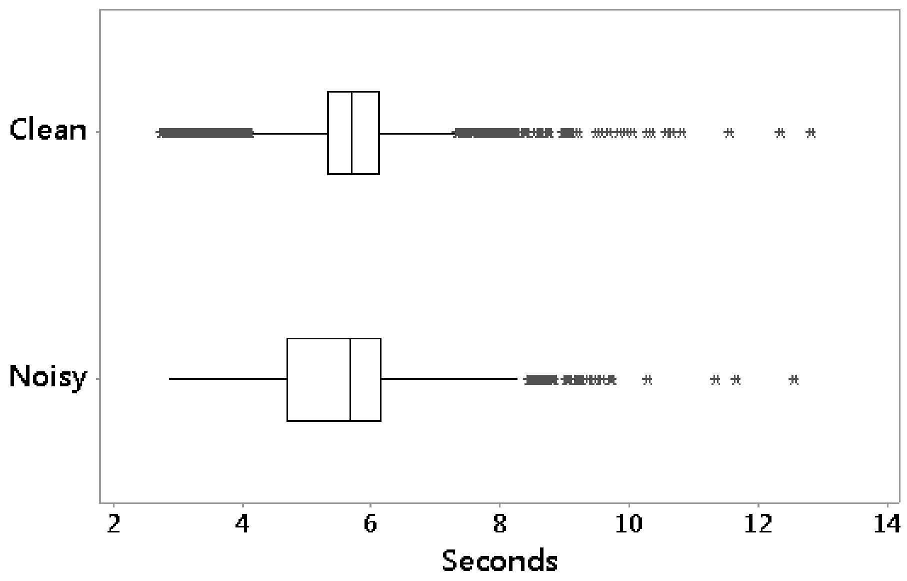


Figure 4.12. Box Plot of leaf image recognition times simulating a mobile app back-end, for Costa Rican noisy and clean subsets

Proportion Tests on the Clean Subset Table 4.6 shows the results obtained for all the proportion tests for the clean subset. Most combinations of HCoS and R1P8 & R3P16 for $1 <= k <= 10$ resulted in very low p-Values, which reject $H0$. However a few accuracy increases from 0.9 HCoS and 0.1 R1P8 & R3P16 did fail the test. This means that, as the weight increases for HCoS, it starts getting non-significant accuracy increases, which makes sense since it is almost equal to HCoS alone.

Proportion Tests on the Noisy Subset Table 4.7 shows the results obtained for all the proportion tests for the noisy subset. All combinations of HCoS and R1P8 & R3P16 resulted in very low p-Values, which reject $H0$.

Proportion Tests on the Complete Dataset Table 4.8 shows the results obtained for all the proportion tests on the complete dataset of leaf images from Costa Rica. Almost every single test rejected $H0$. For $k = 1$ the results are not significant.

In all Proportion Tests, by adding texture with a bigger factor the model improves significantly the accuracy. As the factor assigned to texture declines, the improvement becomes statistically insignificant.

6.4 Processing Time

As shown in Figure 4.12, times range from 2.76 to 12.81 seconds. However, the median of the elapsed time is 5.70 seconds for the clean subset and 5.66 seconds for the noisy subset. These are suitable times even for mobile applications that use the developed back-end.

Table 4.6. Proportion Test results over the Costa Rican Clean Subset

Costa Rica Clean Subset
Confidence Level=0.95
Sample Size=1468
H0: HCoS=HCoS & R1P8 & R3P16
H1: HCoS<HCoS & R1P8 & R3P16

k	HCoS	HCoS=0.1, R1P8 & R3P16=0.9	p-Value	Reject H0?	Accuracy Improvement
1	0.311	0.567	5.65023E-21	YES	0.255
2	0.446	0.702	4.99997E-19	YES	0.255
3	0.535	0.766	2.00608E-15	YES	0.231
4	0.587	0.816	1.21109E-19	YES	0.230
5	0.631	0.857	2.81689E-21	YES	0.225
6	0.674	0.875	9.64321E-21	YES	0.201
7	0.710	0.890	2.70704E-18	YES	0.180
8	0.740	0.903	4.32615E-17	YES	0.163
9	0.768	0.918	6.49779E-16	YES	0.151
10	0.790	0.931	1.14726E-14	YES	0.141

k	HCoS	HCoS=0.5, R1P8 & R3P16=0.5	p-Value	Reject H0?	Accuracy Improvement
1	0.311	0.563	4.32788E-06	YES	0.251
2	0.446	0.702	6.56883E-09	YES	0.256
3	0.535	0.785	1.09341E-11	YES	0.251
4	0.587	0.822	5.88439E-16	YES	0.235
5	0.631	0.854	2.42945E-19	YES	0.223
6	0.674	0.881	4.19306E-23	YES	0.207
7	0.710	0.909	1.18899E-21	YES	0.198
8	0.740	0.924	1.62723E-20	YES	0.185
9	0.768	0.937	7.26426E-20	YES	0.170
10	0.790	0.945	7.84393E-20	YES	0.155

k	HCoS	HCoS=0.9, R1P8 & R3P16=0.1	p-Value	Reject H0?	Accuracy Improvement
1	0.311	0.386	0.976355356	NO	0.075
2	0.446	0.520	0.823819993	NO	0.074
3	0.535	0.610	0.840833982	NO	0.075
4	0.587	0.668	0.26158887	NO	0.082
5	0.631	0.706	0.0201783	YES	0.074
6	0.674	0.748	0.017077481	YES	0.074
7	0.710	0.779	0.002586312	YES	0.069
8	0.740	0.812	0.000201496	YES	0.072
9	0.768	0.832	5.92221E-05	YES	0.065
10	0.790	0.845	3.63353E-06	YES	0.055

Table 4.7. Proportion Test results over the Costa Rican Noisy Subset

Costa Rica Noisy Subset
Confidence Level=0.95
Sample Size=2345
H0: HCoS=HCoS & R1P8 & R3P16
H1: HCoS<HCoS & R1P8 & R3P16

k	HCoS	HCoS=0.1, R1P8 & R3P16=0.9	p-Value	Reject H0?	Accuracy Improvement
1	0.151	0.519	1.7283E-129	YES	0.368
2	0.225	0.638	7.7632E-157	YES	0.413
3	0.277	0.701	1.3354E-165	YES	0.424
4	0.325	0.750	6.4313E-182	YES	0.425
5	0.364	0.783	5.3369E-191	YES	0.420
6	0.399	0.810	2.8814E-194	YES	0.411
7	0.435	0.830	5.1936E-187	YES	0.396
8	0.470	0.844	1.3596E-178	YES	0.374
9	0.496	0.858	2.6133E-177	YES	0.362
10	0.521	0.872	5.9603E-173	YES	0.352
k	HCoS	HCoS=0.5, R1P8 & R3P16=0.5	p-Value	Reject H0?	Accuracy Improvement
1	0.151	0.320	1.2405E-75	YES	0.169
2	0.225	0.435	2.2453E-116	YES	0.209
3	0.277	0.515	1.1237E-149	YES	0.238
4	0.325	0.574	7.6123E-168	YES	0.250
5	0.364	0.616	5.4143E-184	YES	0.252
6	0.399	0.660	1.5749E-202	YES	0.261
7	0.435	0.692	5.0885E-199	YES	0.258
8	0.470	0.721	8.0747E-191	YES	0.250
9	0.496	0.744	1.7097E-191	YES	0.248
10	0.521	0.771	2.0950E-191	YES	0.250
k	HCoS	HCoS=0.9, R1P8 & R3P16=0.1	p-Value	Reject H0?	Accuracy Improvement
1	0.151	0.177	2.4494E-26	YES	0.025
2	0.225	0.257	1.9667E-50	YES	0.032
3	0.277	0.311	1.9949E-63	YES	0.035
4	0.325	0.364	4.4262E-79	YES	0.040
5	0.364	0.408	1.5164E-96	YES	0.044
6	0.399	0.455	6.3080E-102	YES	0.055
7	0.435	0.484	8.9291E-112	YES	0.050
8	0.470	0.516	6.4232E-118	YES	0.046
9	0.496	0.546	4.2650E-125	YES	0.049
10	0.521	0.574	9.9417E-134	YES	0.054

Table 4.8. Proportion Test results over the Costa Rican Complete Dataset

Costa Rica All Dataset
Confidence Level=0.95
Sample Size=2345
H0: HCoS=HCoS & R1P8 & R3P16
H1: HCoS<HCoS & R1P8 & R3P16

k	HCoS	HCoS=0.1, R1P8 & R3P16=0.9	p-Value	Reject H0?	Accuracy Improvement
1	0.070	0.145	4.3210E-01	NO	0.075
2	0.119	0.209	6.2947E-06	YES	0.090
3	0.148	0.252	1.8102E-10	YES	0.105
4	0.176	0.295	3.2827E-12	YES	0.119
5	0.204	0.326	1.6580E-12	YES	0.122
6	0.228	0.350	6.0361E-13	YES	0.122
7	0.253	0.377	8.0774E-12	YES	0.124
8	0.273	0.400	4.1983E-11	YES	0.126
9	0.295	0.417	5.8190E-11	YES	0.122
10	0.318	0.439	1.0927E-10	YES	0.121
k	HCoS	HCoS=0.5, R1P8 & R3P16=0.5	p-Value	Reject H0?	Accuracy Improvement
1	0.070	0.120	8.2576E-01	NO	0.050
2	0.119	0.178	6.0228E-03	YES	0.059
3	0.148	0.216	1.3785E-05	YES	0.069
4	0.176	0.251	4.9141E-09	YES	0.075
5	0.204	0.277	2.9011E-10	YES	0.072
6	0.228	0.304	1.0408E-11	YES	0.076
7	0.253	0.328	9.4610E-11	YES	0.075
8	0.273	0.353	8.2167E-12	YES	0.080
9	0.295	0.371	2.6311E-11	YES	0.076
10	0.318	0.393	1.6020E-10	YES	0.075
k	HCoS	HCoS=0.9, R1P8 & R3P16=0.1	p-Value	Reject H0?	Accuracy Improvement
1	0.070	0.084	6.9915E-01	NO	0.014
2	0.119	0.133	4.7461E-03	YES	0.014
3	0.148	0.165	3.6781E-04	YES	0.018
4	0.176	0.201	7.3870E-06	YES	0.025
5	0.204	0.224	4.0212E-06	YES	0.020
6	0.228	0.249	7.8066E-07	YES	0.021
7	0.253	0.277	2.8185E-06	YES	0.024
8	0.273	0.299	2.0626E-07	YES	0.026
9	0.295	0.320	1.0903E-07	YES	0.025
10	0.318	0.336	1.6458E-06	YES	0.017

6.5 Statistical Analysis of Noise Affectation, Best Algorithms per Species, and best value of k

Table 4.9 shows the results of each GLM per species. Each species has the accuracy maximum, mean and median. Also, a cluster has been assigned regarding the best algorithms resulted from the Tukey test per species. Table 4.10 depicts the algorithms in each cluster for reference. Additionally, column "Best Without Noise" indicates if noise does affect or not the accuracy for each species. Finally, column "$\hat{k}$" indicates the threshold value $\hat{k}$ per species. As indicated before, any $k > \hat{k}$ will be slightly better, but this is not statistically significant.

Noise Affectation. As Table 4.9 shows, most species are affected negatively by noise in the data. However, four species do show no significant difference between noisy and clean data. *Blackea maurafernandesiana*, *Brosimum alicastrum*, *Hura crepitans*, and *Picramnia antidesma* seem to be fairly resilient to noise with these algorithms. Table 4.9 shows some species which got low accuracy values at the bottom. *Annona mucosa* and *Dendropanax arboreus* got a median accuracy of 0.48, and *Aegiphila valerioi* of 0.45. Figure 4.13 shows 4 images of these 3 species. We suspect the reasons behind the low accuracy for these species are the shadows present inside the leaf and outside, against the paper sheet. Also it can be noticed some leaves also have physical damage. *Dendropanax arboreus* in Figure 4.13a and Figure 4.13b also shows how different both sides of the same species are, suggesting we need to separate the dataset in both sides of the leaves.

Value of threshold $\hat{k}$. The best $\hat{k}$ is achieved by species *Muntingia calabura*, with $\hat{k} = 3$. *Bauhinia purpurea* also shows a low value of $\hat{k} = 4$. *Eugenia hiraeifolia*, *Genipa americana*, *Hura crepitans*, *Quercus corrugata* and *Urera caracasana* have $\hat{k} = 5$. Overall, 13 species show a $\hat{k} = 6$ value, while 20 species have $\hat{k} = 7$, and the rest result in $8 <= \hat{k} <= 10$. As $\hat{k}$ is lower, then the potential maximum accuracy for that species tends to be very high.

Best Algorithms per species. Several clusters were identified based on algorithms that showed the best accuracy per species. Table 4.10 shows the list of clusters. Each cluster contains one to three algorithms that had the same statistical significance during our experiments per species. We have in total 10 clusters based on the best, second best and third best algorithm per species. Table 4.9 shows that most of the species belong to clusters that had the combination of 0.1 HCoS and 0.9 R1P8 & R3P16. Some also have R1P8 & R3P16 which is texture alone without curvature.

Figure 4.14 shows the accuracy distribution across the 10 clusters formed after carrying out Tukey tests on the different algorithms. The best algorithms have the biggest factor for texture. Cluster 3, Cluster 8 and Cluster 10, have as the best algorithm the combination 0.1 HCoS and 0.9 R1P8 & R3P16, and have the second best accuracy across all species. The very best cluster is Cluster 9, reaching an Accuracy of 1.

Figure 4.15 shows the distribution of species count per cluster. The cluster with the most species is Cluster 5, with more than 25 species. This cluster, as shown in Table 4.10, contains two best algorithms: 0.1 HCoS and 0.9 R1P8 & R3P16 combination, and 0 HCoS and 1 R1P8 & R3P16 combination. This means both of them are statistically equivalent for these species.

6.6 Statistical Analysis of Best Algorithms for $k = 5$

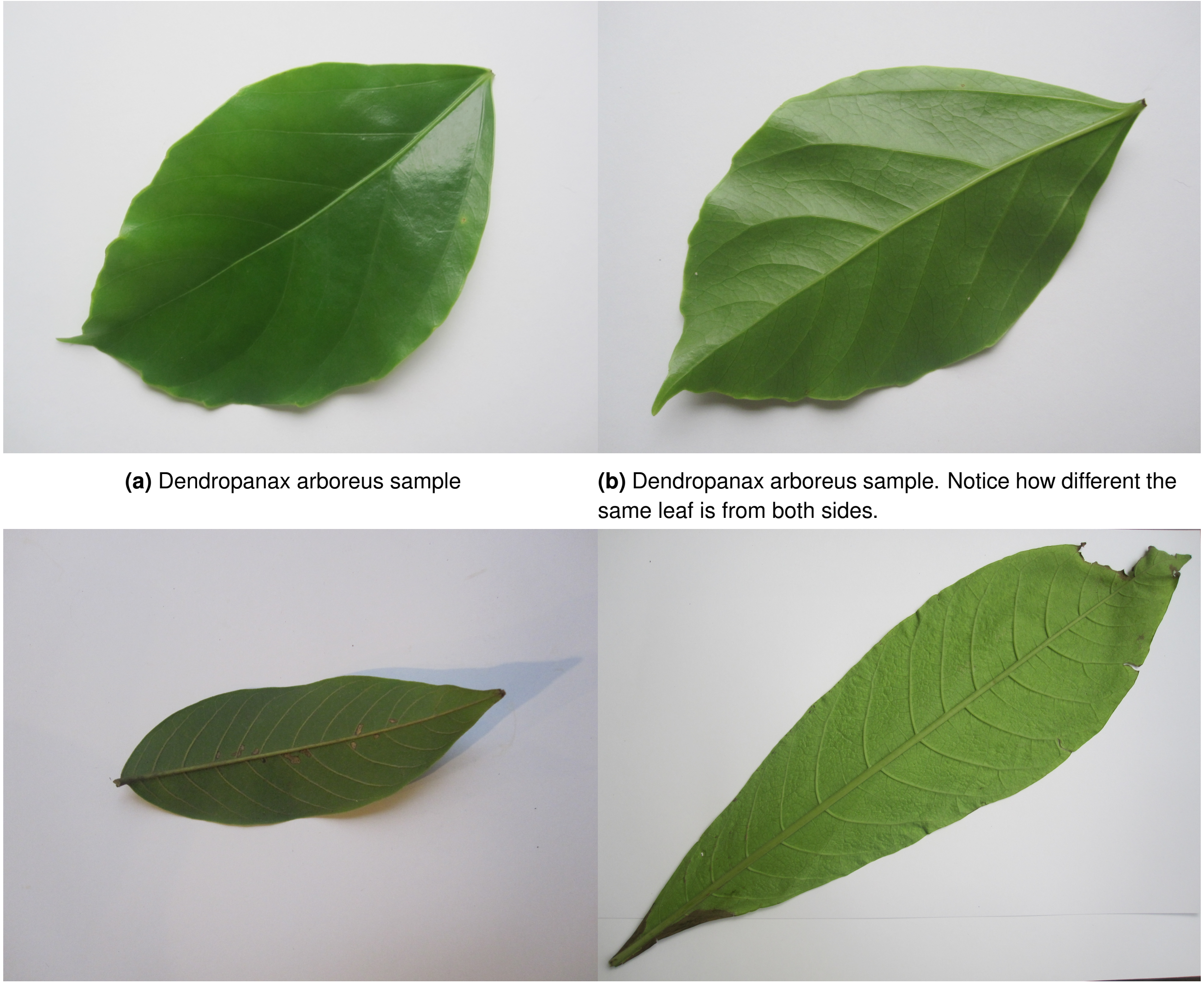
Best Algorithms for $k = 5$ and noisy dataset. Table 4.11 shows what algorithms maximize the probability of a good identification, given that $k = 5$ and the noisy dataset is used. Most algorithms are combinations, from pure texture, to a 0.5 HCoS and 0.5 R1P8 & R3P16 combination. No combination gets near a pure

Table 4.9. Per Species Table with Accuracy Mean, Maximum, Best Algorithms, Affectation by Noise and $\hat{k}$

Species	Maximum Accuracy	Median Accuracy	Mean Accuracy	Cluster	Best without Noise	$\hat{k}$	Number of Images
Bauhinia purpurea	1	0.94	0.92	3	Yes	4	53
Bauhinia ungulata	1	0.83	0.83	2	Yes	7	48
Blackea maurafernandesiana	1	0.87	0.83	4	No	7	42
Calycophyllum candidissimum	1	0.76	0.7	5	Yes	8	90
Cedrela odorata	1	0.79	0.73	5	Yes	6	62
Cestrum tomentosum	1	0.75	0.66	5	Yes	6	68
Citharexylum donnell-smithii	1	0.6	0.6	6	Yes	7	44
Colubrina spinosa	1	0.8	0.75	7	Yes	7	54
Croton draco	1	0.78	0.77	10	Yes	7	63
Dipteryx panamensis	1	0.77	0.7	5	Yes	7	103
Eugenia hiraeifolia	1	0.95	0.86	4	Yes	5	50
Ficus cotinifolia	1	0.7	0.64	5	Yes	7	58
Genipa americana	1	0.87	0.8	9	Yes	5	42
Guaiacum sanctum	1	0.85	0.74	9	Yes	8	68
Guazuma ulmifolia	1	0.89	0.85	5	Yes	6	54
Heliocarpus appendiculatus	1	0.84	0.78	1	Yes	6	56
Hura crepitans	1	0.83	0.83	5	No	5	53
Hymenaea courbaril	1	0.82	0.72	1	Yes	7	80
Muntingia calabura	1	0.96	0.94	9	Yes	3	61
Picramnia antidesma	1	0.77	0.72	1	No	6	52
Platymiscium parviflorum	1	0.6	0.56	5	Yes	8	58
Platymiscium pinnatum	1	0.56	0.6	5	Yes	6	67
Posoqueria latifolia	1	0.66	0.63	5	Yes	7	48
Quercus corrugata	1	0.9	0.82	8	Yes	5	50
Robinsonella lindeniana var. divergens	1	0.83	0.8	9	Yes	7	48
Samanea saman	1	0.74	0.69	8	Yes	8	78
Stemmadenia donnell-smithii	1	0.65	0.61	3	Yes	7	56
Tabebuia impetiginosa	1	0.85	0.79	5	Yes	7	58
Tabebuia ochracea	1	0.81	0.74	5	Yes	7	66
Tabebuia ochracea CR	1	0.81	0.71	4	Yes	7	36
Terminalia oblonga	1	0.7	0.67	8	Yes	8	64
Urera caracasana	1	0.71	0.72	5	Yes	5	28
Vernonia patens	1	0.71	0.64	5	Yes	6	36
Zygia longifolia	1	0.76	0.66	1	Yes	8	60
Astronium graveolens	0.97	0.65	0.62	1	Yes	7	78
Croton niveus	0.96	0.71	0.65	5	Yes	6	34
Terminalia amazonia	0.96	0.73	0.68	8	Yes	9	110
Trichilia havanensis	0.96	0.62	0.58	5	Yes	8	76
Acnistus arborescens	0.95	0.7	0.61	1	Yes	9	47
Ardisia revoluta	0.95	0.55	0.53	8	Yes	8	60
Erythrina poeppigiana	0.95	0.55	0.54	1	Yes	8	50
Sapium glandulosum	0.95	0.6	0.61	6	Yes	6	50
Tabebuia rosea	0.95	0.6	0.57	7	Yes	6	40
Anacardium excelsum	0.94	0.72	0.62	1	Yes	7	58
Calophyllum brasiliense	0.94	0.66	0.61	5	Yes	7	61
Cordia eriostigma	0.94	0.55	0.5	5	Yes	6	38
Hyeronima alchorneoides	0.94	0.72	0.61	5	Yes	8	50
Simarouba glauca	0.94	0.71	0.65	5	Yes	8	121
Swietenia macrophylla	0.94	0.54	0.53	5	Yes	8	60
Persea americana	0.93	0.55	0.54	5	Yes	7	42
Manilkara chicle	0.91	0.64	0.6	5	Yes	8	65
Pimenta dioica	0.91	0.62	0.59	4	Yes	8	58
Tabernaemontana littoralis	0.91	0.62	0.53	5	Yes	9	56
Clusia croatii	0.9	0.75	0.62	5	Yes	6	50
Ocotea sinuata	0.89	0.56	0.52	5	Yes	7	46
Sideroxylon capiri	0.88	0.6	0.57	1	Yes	8	55
Brosimum alicastrum	0.87	0.61	0.58	5	No	7	60
Cretra costaricense	0.87	0.44	0.47	1	Yes	9	48
Psidium guajava	0.87	0.55	0.48	8	Yes	8	40
Pachira quinata	0.86	0.6	0.57	8	Yes	8	79
Solanum rovirosanum	0.86	0.57	0.54	1	Yes	8	56
Aegiphila valerioi	0.81	0.5	0.45	8	Yes	6	44
Dendropanax arboreus	0.81	0.5	0.48	8	Yes	8	54
Annona mucosa	0.77	0.5	0.48	2	Yes	8	55

Table 4.10. Cluster definition and most significant Algorithms per Cluster

Cluster Name	Algorithm
1	0.1 HCoS and 0.9 R1P8 & R3P16
2	0.5 HCoS and 0.5 R1P8 & R3P16
3	0.1 HCoS and 0.9 R1P8 & R3P16 0 HCoS and 1 R1P8 & R3P16 0.5 HCoS and 0.5 R1P8 & R3P16
4	0.1 HCoS and 0.9 R1P8 & R3P16 0.5 HCoS and 0.5 R1P8 & R3P16 0 HCoS and 1 R1P8 & R3P16
5	0.1 HCoS and 0.9 R1P8 & R3P16 0 HCoS and 1 R1P8 & R3P16
6	0.5 HCoS and 0.5 R1P8 & R3P16 0.1 HCoS and 0.9 R1P8 & R3P16 0.9 HCoS and 0.1 R1P8 & R3P16
7	0 HCoS and 1 R1P8 & R3P16 0.1 HCoS and 0.9 R1P8 & R3P16
8	0.5 HCoS and 0.5 R1P8 & R3P16 0.1 HCoS and 0.9 R1P8 & R3P16
9	0.1 HCoS and 0.9 R1P8 & R3P16 0.5 HCoS and 0.5 R1P8 & R3P16
10	0.5 HCoS and 0.5 R1P8 & R3P16 0.1 HCoS and 0.9 R1P8 & R3P16 0 HCoS and 1 R1P8 & R3P16


(a) Dendropanax arboreus sample

(b) Dendropanax arboreus sample. Notice how different the same leaf is from both sides.

(c) Annona mucosa sample showing lots of shadows

(d) Damaged leaf of Aegiphila valerioi species

Figure 4.13. Leaf samples of species with low accuracy

curvature algorithm. Similar results are noted in Figure 4.16 where the best probabilities are around the 0.2 HCoS and 0.8 R1P8 & R3P16 combination. In general the distribution is very homogenous.

Best Algorithms for $k = 5$ and clean dataset. Table 4.12 shows what algorithms maximize the probability of a good identification, given that $k = 5$ and the clean dataset is used. In this case the best algorithms per species are more spread across most combinations. This is due the lack of noise in the data and the lesser affectation of the curvature algorithms. This, compared with the data of Table 4.11 confirms how texture seems to be more robust with noise. Figure 4.17 shows the distribution of probabilities of a good identification per algorithm. On clean data it seems the biggest probabilities are near the center with combinations around the 0.3 HCoS and 0.7 R1P8 & R3P16 combination.

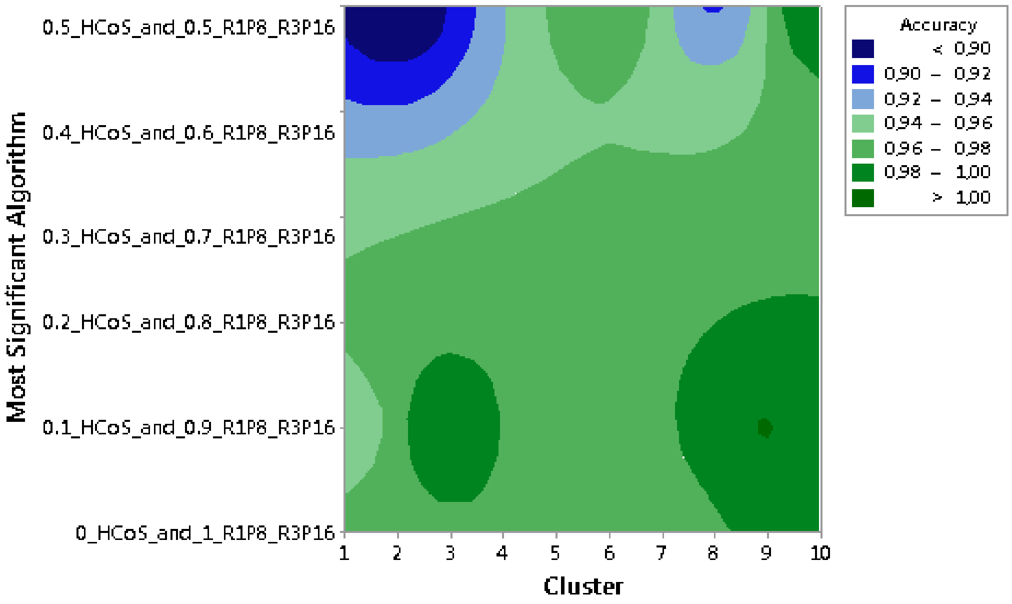


Figure 4.14. Accuracy distribution across different clusters found on the species

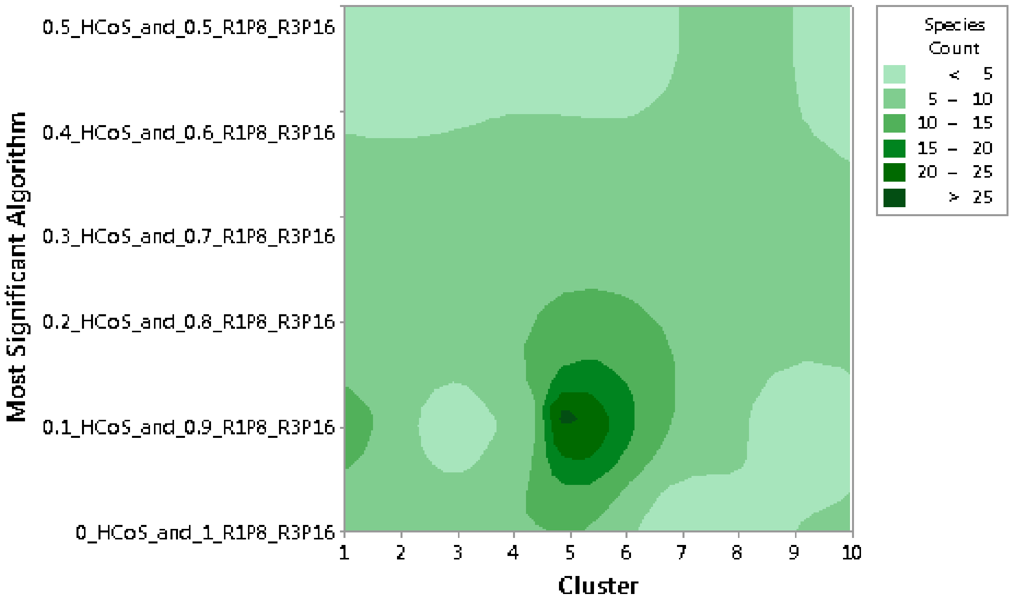


Figure 4.15. Species count distribution across different clusters

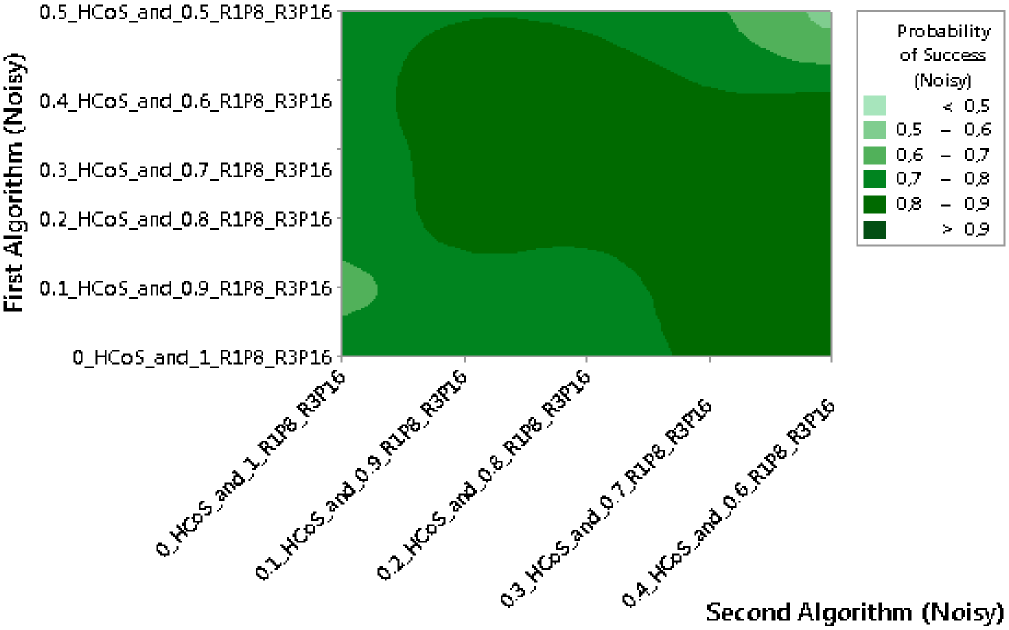


Figure 4.16. Distribution of Probability of successful identification with noisy data and $k = 5$

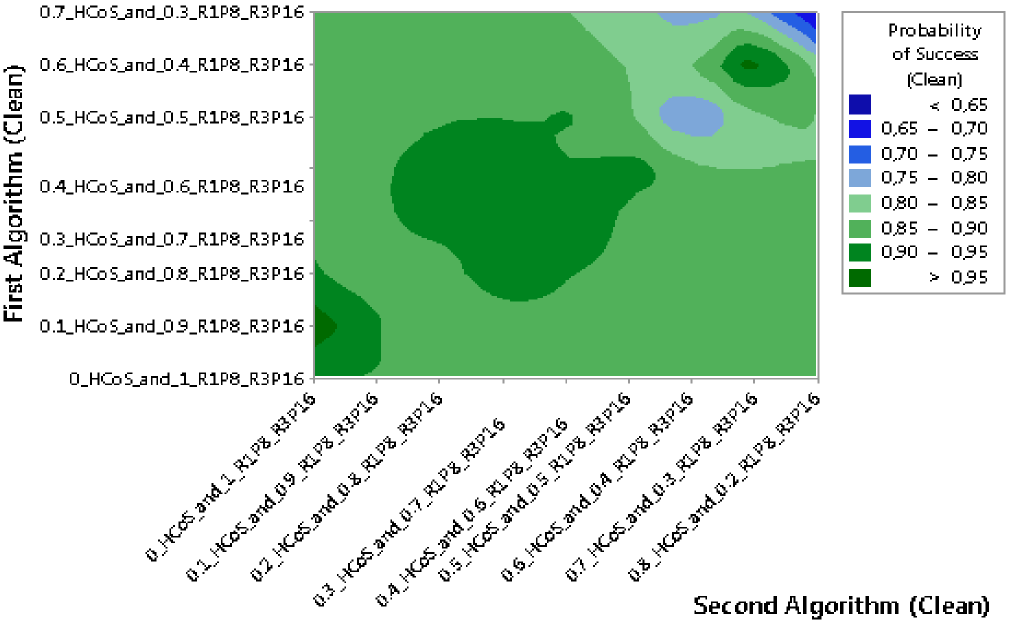


Figure 4.17. Distribution of Probability of successful identification with clean data and $k = 5$

Table 4.11. Algorithms that maximize the probability of a good identification for all species on noisy data, with a fixed $k = 5$

Species	First Algorithm	Probability of Success
Acnistus arborescens	0.1 HCoS and 0.9 R1P8 & R3P16	0.903
Aegiphila valerioi	0 HCoS and 1 R1P8 & R3P16	0.484
Anacardium excelsum	0.1 HCoS and 0.9 R1P8 & R3P16	0.725
Annona mucosa	0.1 HCoS and 0.9 R1P8 & R3P16	0.436
Ardisia revoluta	0.2 HCoS and 0.8 R1P8 & R3P16	0.592
Astronium graveolens	0.2 HCoS and 0.8 R1P8 & R3P16	0.729
Bauhinia purpurea	0.2 HCoS and 0.8 R1P8 & R3P16	0.943
Bauhinia ungulata	0.3 HCoS and 0.7 R1P8 & R3P16	0.858
Blackea maurafernandesiana	0.1 HCoS and 0.9 R1P8 & R3P16	0.889
Brosimum alicastrum	0 HCoS and 1 R1P8 & R3P16	0.749
Calophyllum brasiliense	0.1 HCoS and 0.9 R1P8 & R3P16	0.808
Calycophyllum candidissimum	0.1 HCoS and 0.9 R1P8 & R3P16	0.844
Cedrela odorata	0.2 HCoS and 0.8 R1P8 & R3P16	0.858
Cestrum tomentosum	0.1 HCoS and 0.9 R1P8 & R3P16	0.867
Citharexylum donnell-smithii	0.5 HCoS and 0.5 R1P8 & R3P16	0.591
Clusia croatii	0 HCoS and 1 R1P8 & R3P16	0.797
Colubrina spinosa	0.2 HCoS and 0.8 R1P8 & R3P16	0.906
Cordia eriostigma	0.1 HCoS and 0.9 R1P8 & R3P16	0.704
Cretra costaricense	0.1 HCoS and 0.9 R1P8 & R3P16	0.609
Croton draco	0.2 HCoS and 0.8 R1P8 & R3P16	0.876
Croton niveus	0 HCoS and 1 R1P8 & R3P16	0.863
Dendropanax arboreus	0 HCoS and 1 R1P8 & R3P16	0.452
Dipteryx panamensis	0.2 HCoS and 0.8 R1P8 & R3P16	0.855
Erythrina poeppigiana	0.1 HCoS and 0.9 R1P8 & R3P16	0.767
Eugenia hiraeifolia	0.2 HCoS and 0.8 R1P8 & R3P16	0.99
Ficus cotinifolia	0.1 HCoS and 0.9 R1P8 & R3P16	0.849
Genipa americana	0.2 HCoS and 0.8 R1P8 & R3P16	0.89
Guaiacum sanctum	0.3 HCoS and 0.7 R1P8 & R3P16	0.905
Guazuma ulmifolia	0 HCoS and 1 R1P8 & R3P16	0.95
Heliocarpus appendiculatus	0.2 HCoS and 0.8 R1P8 & R3P16	0.957
Hura crepitans	0 HCoS and 1 R1P8 & R3P16	0.902
Hyeronima alchorneoides	0 HCoS and 1 R1P8 & R3P16	0.743
Hymenaea courbaril	0.2 HCoS and 0.8 R1P8 & R3P16	0.874
Manilkara chicle	0.1 HCoS and 0.9 R1P8 & R3P16	0.696
Muntingia calabura	0.2 HCoS and 0.8 R1P8 & R3P16	0.98
Ocotea sinuata	0 HCoS and 1 R1P8 & R3P16	0.807
Pachira quinata	0.1 HCoS and 0.9 R1P8 & R3P16	0.631
Persea americana	0 HCoS and 1 R1P8 & R3P16	0.796
Picramnia antidesma	0 HCoS and 1 R1P8 & R3P16	0.94
Pimenta dioica	0.1 HCoS and 0.9 R1P8 & R3P16	0.638
Platymiscium parviflorum	0.1 HCoS and 0.9 R1P8 & R3P16	0.709
Platymiscium pinnatum	0.1 HCoS and 0.9 R1P8 & R3P16	0.858
Posoqueria latifolia	0.1 HCoS and 0.9 R1P8 & R3P16	0.844
Psidium guajava	0.1 HCoS and 0.9 R1P8 & R3P16	0.477
Quercus corrugata	0.3 HCoS and 0.7 R1P8 & R3P16	0.942
Robinsonella lindeniana var. divergens	0.3 HCoS and 0.7 R1P8 & R3P16	0.888
Samanea saman	0.3 HCoS and 0.7 R1P8 & R3P16	0.827
Sapium glandulosum	0.2 HCoS and 0.8 R1P8 & R3P16	0.589
Sideroxylon capiri	0 HCoS and 1 R1P8 & R3P16	0.648
Simarouba glauca	0 HCoS and 1 R1P8 & R3P16	0.734
Solanum rovirosanum	0.1 HCoS and 0.9 R1P8 & R3P16	0.731
Stemmadenia donnell-smithii	0.2 HCoS and 0.8 R1P8 & R3P16	0.708
Swietenia macrophylla	0.1 HCoS and 0.9 R1P8 & R3P16	0.762
Tabebuia impetiginosa	0.1 HCoS and 0.9 R1P8 & R3P16	0.93
Tabebuia ochracea	0.2 HCoS and 0.8 R1P8 & R3P16	0.9
Tabebuia ochracea (Costa Rica)	0 HCoS and 1 R1P8 & R3P16	0.892
Tabebuia rosea	0.1 HCoS and 0.9 R1P8 & R3P16	0.728
Tabernaemontana littoralis	0 HCoS and 1 R1P8 & R3P16	0.716
Terminalia amazonia	0.2 HCoS and 0.8 R1P8 & R3P16	0.726
Terminalia oblonga	0.2 HCoS and 0.8 R1P8 & R3P16	0.731
Trichilia havanensis	0.1 HCoS and 0.9 R1P8 & R3P16	0.687
Urera caracasana	0 HCoS and 1 R1P8 & R3P16	0.963
Vernonia patens	0 HCoS and 1 R1P8 & R3P16	0.932
Zygia longifolia	0.1 HCoS and 0.9 R1P8 & R3P16	0.781

Table 4.12. Algorithms that maximize the probability of a good identification for all species on clean data, with a fixed $k = 5$

Species	First Algorithm	Probability of Success
Acnistus arborescens	0.2 HCoS and 0.8 R1P8 & R3P16	0.948
Aegiphila valerioi	0.7 HCoS and 0.3 R1P8 & R3P16	0.692
Anacardium excelsum	0.4 HCoS and 0.6 R1P8 & R3P16	0.877
Annona mucosa	0.7 HCoS and 0.3 R1P8 & R3P16	0.646
Ardisia revoluta	0.5 HCoS and 0.5 R1P8 & R3P16	0.82
Astronium graveolens	0.4 HCoS and 0.6 R1P8 & R3P16	0.873
Bauhinia purpurea	0.5 HCoS and 0.5 R1P8 & R3P16	0.981
Bauhinia ungulata	0.6 HCoS and 0.4 R1P8 & R3P16	0.961
Blackea maurafernandesiana	0.4 HCoS and 0.6 R1P8 & R3P16	0.946
Brosimum alicastrum	0.1 HCoS and 0.9 R1P8 & R3P16	0.825
Calophyllum brasiliense	0.2 HCoS and 0.8 R1P8 & R3P16	0.903
Calycophyllum candidissimum	0.3 HCoS and 0.7 R1P8 & R3P16	0.923
Cedrela odorata	0.4 HCoS and 0.6 R1P8 & R3P16	0.938
Cestrum tomentosum	0.3 HCoS and 0.7 R1P8 & R3P16	0.934
Citharexylum donnell-smithii	0.7 HCoS and 0.3 R1P8 & R3P16	0.875
Clusia croatii	0.1 HCoS and 0.9 R1P8 & R3P16	0.857
Colubrina spinosa	0.3 HCoS and 0.7 R1P8 & R3P16	0.957
Cordia eriostigma	0.2 HCoS and 0.8 R1P8 & R3P16	0.833
Cretra costaricense	0.2 HCoS and 0.8 R1P8 & R3P16	0.737
Croton draco	0.3 HCoS and 0.7 R1P8 & R3P16	0.95
Croton niveus	0.2 HCoS and 0.8 R1P8 & R3P16	0.913
Dendropanax arboreus	0.7 HCoS and 0.3 R1P8 & R3P16	0.717
Dipteryx panamensis	0.4 HCoS and 0.6 R1P8 & R3P16	0.941
Erythrina poeppigiana	0.1 HCoS and 0.9 R1P8 & R3P16	0.747
Eugenia hiraeifolia	0.2 HCoS and 0.8 R1P8 & R3P16	0.998
Ficus cotinifolia	0.2 HCoS and 0.8 R1P8 & R3P16	0.918
Genipa americana	0.3 HCoS and 0.7 R1P8 & R3P16	0.952
Guaiacum sanctum	0.4 HCoS and 0.6 R1P8 & R3P16	0.967
Guazuma ulmifolia	0.1 HCoS and 0.9 R1P8 & R3P16	0.97
Heliocarpus appendiculatus	0.3 HCoS and 0.7 R1P8 & R3P16	0.981
Hura crepitans	0.2 HCoS and 0.8 R1P8 & R3P16	0.943
Hyeronima alchorneoides	0.5 HCoS and 0.5 R1P8 & R3P16	0.846
Hymenaea courbaril	0.3 HCoS and 0.7 R1P8 & R3P16	0.942
Manilkara chicle	0.3 HCoS and 0.7 R1P8 & R3P16	0.847
Muntingia calabura	0.3 HCoS and 0.7 R1P8 & R3P16	0.99
Ocotea sinuata	0.1 HCoS and 0.9 R1P8 & R3P16	0.873
Pachira quinata	0.3 HCoS and 0.7 R1P8 & R3P16	0.806
Persea americana	0 HCoS and 1 R1P8 & R3P16	0.844
Picramnia antidesma	0.1 HCoS and 0.9 R1P8 & R3P16	0.961
Pimenta dioica	0.5 HCoS and 0.5 R1P8 & R3P16	0.823
Platymiscium parviflorum	0.3 HCoS and 0.7 R1P8 & R3P16	0.829
Platymiscium pinnatum	0.2 HCoS and 0.8 R1P8 & R3P16	0.918
Posoqueria latifolia	0.2 HCoS and 0.8 R1P8 & R3P16	0.92
Psidium guajava	0.5 HCoS and 0.5 R1P8 & R3P16	0.687
Quercus corrugata	0.4 HCoS and 0.6 R1P8 & R3P16	0.982
Robinsonella lindeniana var. divergens	0.5 HCoS and 0.5 R1P8 & R3P16	0.964
Samanea saman	0.4 HCoS and 0.6 R1P8 & R3P16	0.934
Sapium glandulosum	0.7 HCoS and 0.3 R1P8 & R3P16	0.843
Sideroxylon capiri	0.2 HCoS and 0.8 R1P8 & R3P16	0.767
Simarouba glauca	0.3 HCoS and 0.7 R1P8 & R3P16	0.85
Solanum rovirosanum	0.2 HCoS and 0.8 R1P8 & R3P16	0.844
Stemmadenia donnell-smithii	0.4 HCoS and 0.6 R1P8 & R3P16	0.872
Swietenia macrophylla	0.2 HCoS and 0.8 R1P8 & R3P16	0.852
Tabebuia impetiginosa	0.2 HCoS and 0.8 R1P8 & R3P16	0.966
Tabebuia ochracea	0.3 HCoS and 0.7 R1P8 & R3P16	0.957
Tabebuia ochracea (Costa Rica)	0.2 HCoS and 0.8 R1P8 & R3P16	0.938
Tabebuia rosea	0.3 HCoS and 0.7 R1P8 & R3P16	0.857
Tabernaemontana littoralis	0.1 HCoS and 0.9 R1P8 & R3P16	0.793
Terminalia amazonia	0.4 HCoS and 0.6 R1P8 & R3P16	0.871
Terminalia oblonga	0.5 HCoS and 0.5 R1P8 & R3P16	0.895
Trichilia havanensis	0.3 HCoS and 0.7 R1P8 & R3P16	0.831
Urera caracasana	0.1 HCoS and 0.9 R1P8 & R3P16	0.976
Vernonia patens	0 HCoS and 1 R1P8 & R3P16	0.95
Zygia longifolia	0.2 HCoS and 0.8 R1P8 & R3P16	0.884

7 Conclusions

The addition of texture increases significantly the accuracy of our implementation of the HCoS. When comparing HCoS versus the combination of 0.1 HCoS and 0.9 R1P8 & R3P16, for the Costa Rican clean subset, the improvement ranges from 14.1% to 25.5%, depending on the value of k. Similarly, with the noisy subset, the improvement ranges from 35.5% to 42.5%. These improvements were proved to be statistically significant in our experiments.

The complete dataset experiments demonstrated that poor accuracy levels are achieved when noisy images are classified against clean images. We speculate that this is due to the many enhancements that leaf images underwent before being added to the clean dataset. First, leaves were pressed for 24 hours in order to flatten them and thus minimize the presence of shadows. Secondly, Photoshop was used to manually remove artifacts. Finally, image enhancement algorithms (e.g., stem removal) were applied. This result has important implications if a mobile application is developed, given that users will take noisy pictures. As a result we are left with two alternatives. The first one is to use a noisy dataset to train the classifier. Alternatively a clean dataset could be used but user images would need to undergo further automated image enhancements comparable to those performed manually with Photoshop.

Experiments for individual species provided some interesting results. Concerning minimal values of k, i.e., the size of the set of candidates that are considered best possibilities in an identification process, good levels of accuracy were obtained for $\hat{k} = 7$ in 63% of the species. Working with noisy images had a negative effect on levels of accuracy on 61 out of 65 species studied, as compared to clean images and a clean dataset. Finally, texture also stands out in most individual cases as the determining factor for high accuracy levels as compared to leaf shape.

Our statistical analysis of best algorithms for $k = 5$ did not render a clear winner but highlighted that the best combination of algorithms should use weigths smaller than 0.2 to HCoS.

8 Future Work

A natural next step in this research is to develop a mobile app that uses the georeference of photographs of leaves as an additional criterion to classify species. Most modern mobile phones already include excellent cameras and provide the option of automatically georeferencing any picture taken with these cameras. In addition to the reference image dataset such as the one developed for this research, maps of potential distribution of species of Costa Rican trees would be needed. *Atta*, a comprehensive and fully georeferenced database of thousands of species of organisms from Costa Rica developed by the Instituto Nacional de Biodiversidad, Costa Rica (INBio) [2] and GBIF's database [3] are excellent foundations to generate these potential distribution maps of species. In addition to curvature, texture, and georeferencing as discriminating factors, morphological measures of leaves are also frequently used by specialists to identify plant species. Some of these measures are: aspect ratio, which is the ratio of horizontal width to vertical length; form coefficient, which is a numerical value that grades the leaf shape as between circular (shortest perimeter for a given area) and filliform (longest perimeter for a given area); and blade and petiole length. Algorithms to calculate these measures have already been developed (e.g., WinFOLIA). However, they have not been integrated in computer vision systems for automatic identification of plant species.

A crowd sourcing approach could be a very efficient way to increase the size of the image dataset that currently comprises 66 plant species from Costa Rica. In addition, crowdsourcing could also be used to clean noisy pictures as a citizen science project.

[2] http://www.inbio.ac.cr

[3] http://www.gbif.org

Finally, the individual contribution of texture features such as venation, porosity, and reflection in characterizing a plant species has not been formally established. A more elaborate analysis of the leaf texture that disaggregates it into a separate layer for each these features would help understand and quantify their individual contribution.

Acknowledgement

To the National Biodiversity Institute of Costa Rica (INBio) and Nelson Zamora for their help with the leaf sample recollection and expert feedback during this research.

Chapter 5

On the Significance of Leaf Sides in Automatic Leaf-based Plant Species Identification

Reference Jose Carranza-Rojas and Erick Mata-Montero (2016b). "On the significance of leaf sides in automatic leaf-based plant species identification". In: *2016 IEEE 36th Central American and Panama Convention (CONCAPAN XXXVI)*, pp. 1–6. DOI: 10.1109/CONCAPAN.2016.7942341

Keywords Biodiversity Informatics, Computer Vision, Image Processing, Plant Identification

1 Abstract

Because the front side of a leaf and the underside are functionally very different – the former captures sunlight to produce photosynthesis and the latter absorbs carbon dioxide and releases oxygen and vapor – they typically have different visual features. In this paper we study the significance of leaf sides in visual recognition systems for automatic plant species identification. We measure the accuracy of species identifications with a dataset of 63 species of trees from Costa Rica that includes pictures of both, front sides and undersides of tree leaves. The dataset is used as a global dataset and is also partitioned as two datasets: one of front side pictures and one of underside pictures. Training and testing of different algorithms is performed and their accuracies computed for the group of species and for each individual species. For the tested dataset, leaf side is a significant factor for automatic plant species identification. On the average, and for most cases, underside pictures lead to more accurate identifications.

2 Introduction

Automatic identification of organisms has not only been a dream among systematists for centuries (MacLeod 2007), but also a current need to understand, sustainably use, and save biodiversity. Several automatic and semi-automatic approaches have been used in the past. For example, dichotomous keys, multi-access keys, morphometrics, DNA barcoding, and image-based identification, among others (Mata-Montero et al. 2016). A number of computer vision and machine learning techniques use leaf images to identify plant species (Carranza-Rojas et al. 2016a; Kumar et al. 2012; Nguyen et al. 2013; Wu et al. 2007). It is usually assumed that a user takes a picture P of the front side of a specimen's leaf, which is then used by an algorithm or model M to establish a ranking of the best k candidate species for P, for some "small" value of k (, $1 \leq k \leq 5$). Supervised training techniques are typically used to train model M with leaf image datasets that often include pictures of the front side and the underside of leaves of specimens that have been previously identified. Because research that aims to identify plants based on leaf images alone tries to get the best out of the leaf visible features, it is important to consider as many leaf discriminant factors as possible. Nevertheless, to our knowledge, previous research on automatic visual plant species identification based on leaf images use front side and underside pictures indiscriminately.

From a functional point of view, the front side and the underside of a leaf are in charge of two different critical tasks. The front side surface gathers energy from sunlight while apertures (stomata) on the the cooler shady underside bring in carbon dioxide and release oxygen and vapor. As a result, the front side and the underside of a leaf tend to have a different appearance. The front side tends to be glossy and has more vivid colors while the underside may have more trichomes (hairs) to keep the surface cool, could be duller, and veins could be more visible.

In this paper we study the significance of leaf sides in automatic visual plant species identification based on leaf images. Our hypothesis is that an automated leaf image-based plant identification system benefits from having the training dataset split into two subsets: one that comprises front side pictures only and one that consists of back side pictures only, which leads to two different plant identification models that we call $Model_F$ and $Model_B$, respectively. We postulate that $Model_F$ and $Model_B$ would be more accurate than M when the image P corresponds to the front and back side of a leaf, respectively. As a pragmatic consequence, when a user provides a picture P for an automatic identification, they should indicate the leaf side so that either $Model_F$ or $Model_B$ is used. However, even if the hypothesis does not hold true, it may still be significant if P is a front side or an underside picture when a general model M is used. Therefore, our experiments also address this issue.

Because of the rich diversity of plant and even tree species in Costa Rica, we realize that the results of this research are affected by the subset of species used. Some species may have front sides of leaves that are very distinctive while others may have undersides that are more discriminating. The accuracy achieved globally for the dataset used in this research may not reflect the importance of leaf sides in automatic plant species identification for individual species. Thus, our experiments also assess, for each of the 63 species in the dataset, the accuracy of models $Model_F$, $Model_B$, and M when picture P corresponds to either the front side or the underside of a leaf.

The rest of this manuscript is organized as follows: Section 3 summarizes relevant related work. Section 4 and Section 5 cover methodological aspects and experiment design, respectively. Section 6 describes the results obtained. Section 1 presents the conclusions and, finally, Section 8 summarizes future work.

3 Related Work

Previous research on leaf image-based identification of plant species has been reported in (Kumar et al. 2012), (Herdiyeni et al. 2013), (Nguyen et al. 2013), and (Carranza-Rojas et al. 2016a). LeafSnap (Kumar et al. 2012) uses a curvature model and similarity search using kNN with an image dataset of North American trees that comprises 184 species. Herdiyeni et al. 2013 use LBP features to classify medicinal and house plants from Indonesia based also on leaf images, for a total of 30 species. Nguyen et al. Nguyen et al. 2013 use SURF to develop an Android application for mobile plant species recognition based on leaf images of 32 species. Finally, Carranza-Rojas et al. 2016a extends work in (Kumar et al. 2012) along two lines. First, LeafSnap's underlying algorithms are applied to a set of 66 tree species from Costa Rica. Secondly, texture is used as an additional criterion to measure the level of improvement achieved in the automatic identification of Costa Rica tree species. None of these studies address the issue of significance of leaf sides in automatic leaf-based plant species identification.

4 Methodology

We used the same approach as Kumar et al. 2012 and Carranza-Rojas et al. 2016a to classify leaves into species of plants. The dataset of images is a subset of the one used in (Carranza-Rojas et al. 2016a). However, a first step was to add metadata indicating the leaf side of each image. Then, leaf segmentation was carried out by using EM. After that, two leaf features were extracted, namely (visual) texture and curvature. Then, classification was done using kNN with $3 \leq k \leq 5$ and using histogram intersection as distance metric. Finally, the accuracy achieved by the classifier was calculated.

Th following subsections provide more details about the image data used, the segmentation approach, and the algorithms used for feature extraction.

4.1 Image Data

The dataset created by Carranza-Rojas et al. 2016a was used almost in its entirety; it includes images of 63 species of randomly picked trees from Costa Rica's central plateau region. Labels were added to logically separate leaf front side from leaf underside images, which allowed us to experiment with each image dataset separately or in combined form. Following the notation presented in Section 2, $Train_F$ is the subset that comprises all 998 front side leaf images, $Train_B$ is the subset that contains all 991 back side leaf images, and finally $Train_C$ is the complete dataset with all 1989 images combined.

4.2 Segmentation

For segmentation we used the HSV color space to cluster pixels into two clusters using EM. However, we discarded the Hue channel since it often contains too much noise. One cluster corresponds to the *lamina* (leaf blade) and the other one to the background.

4.3 Features

We extracted two different feature sets, one for texture and one for margin or curvature. The following subsections explain briefly both algorithms.

LBPU

As mentioned in Section 2, the front and back side of a leaf typically display different textures. LBPU is a feature extraction algorithm that is rotation invariant and has been proved to be excellent for texture pattern matching (Ojala et al. 2002). The following three different variations of LBPU are used:

- Radius of 1 pixel, 8 pixels of sampling. We call it R1P8.
- Radius of 3 pixels, 16 pixels of sampling. We call it R3P16.
- The concatenation of the previous 2 into a single histogram. We call it R1P8P3P16.

A LBPU descriptor is applied to each pixel c in the image and its circular neighborhood $Neighborhood(c)$ that has radius R and P pixels. For each pixel p in $Neighborhood(c)$, p has a gray level $gray(p)$. A boolean threshold function is applied to the difference of gray value between each pixel p from the neighborhood of c and the central pixel c, to form a binary number of length P. To achieve rotation invariance, right shifts are applied to the binary number and then the minimum number is selected.

HCoS

This descriptor was developed by the authors in (Kumar et al. 2012). First, a disk of radius $1 \leq r \leq 25$ is defined at every contour pixel of the leaf. Then, two different histograms are created by measuring the pixel area of the intersection of the disk with the leaf and the length of the arc defined by the intersection of the circumference of the disk and the leaf. This is calculated for all 25 values of radius r and concatenated together into a single histogram called HCoS.

This curvature descriptor, as well as the LBPU variants described, are levels of the factor named *Algorithm*, as explained in Section 5, which describes the experiments. Even though the curvature of the front side and the back side of a leaf are mirror images of each other, this feature was included in the analysis just to determine if it is relevant or should be discarded in future analysis.

4.4 Trained Models and Classification

Classification was carried out by using kNN with $3 \leq k \leq 5$, which, from a user point of view, is a reasonable range of "small" values of k. To calculate the distance between histograms, we used *histogram intersection* as described in (Kumar et al. 2012).

Three algorithms or trained models were defined. $Model_F$ is the model trained with only front side images, $Model_B$ is the model trained with back side images, and $Model_C$ is the model trained with with the complete image dataset.

We calculated the accuracy of the different models. Let E be an identification experiment that consists of a model M, a set S that contains n images of leaves of n (not necessarily different) unknown tree species to be identified, and an integer value k, $k \geq 1$. We define $hit(M, k, x)$ as a boolean function that indicates if model M generates a ranking in which one of the top k candidate species is a correct identification of sample x. Equation 5.1 formally defines $Accuracy(M, S, k)$.

$$Accuracy(M, S, k) = \sum_{x \in S} \frac{hit(M, k, x)}{n} \qquad \textbf{(5.1)}$$

Table 5.1. Levels for *Training+Model* factor

Level	Description
$Test_B + Model_B$	Model tested with back side images and trained with back side images
$Test_B + Model_C$	Model tested with back side images and trained with complete dataset
$Test_C + Model_C$	Model tested with complete dataset and trained with complete dataset
$Test_F + Model_C$	Model tested with front side images and trained with complete dataset
$Test_F + Model_F$	Model tested with front side images and trained with front side images
$Test_B + Model_F$	Model tested with back side images and trained with front side images
$Test_F + Model_B$	Model tested with front side images and trained with back side images

5 Experiments

We ran the classifier over the two datasets $Train_F$ and $Train_B$ to get the accuracy related to front side and back side leaf images. We also ran it for the complete, combined dataset $Train_C$. Additionally, we used a GLM to test if the leaf side was actually a significant factor during classification, with a confidence level of 95%. The three factors used in the GLM are: *Algorithm*, *k*, and *Training+Model*. Factor *Training+Model* represents the combination of a particular trained model, and the dataset used for testing. Table 5.1 shows the seven levels related to this factor. We used $3k5$ only, since those values would be suitable for a species ranking for a mobile app or similar.

After finding if the *Training+Model* factor was significant, a Tukey test was run to assess if the difference between levels for the *Training+Model* was statistically significative, with a confidence level of 95%. This would tell us how relevant leaf side are across the tests.

We ran this globally for all species, but we also ran the GLM for each species separately. This would tell us the role of the leaf side for each species.

6 Results

6.1 Global significance of leaf side

Table 5.2 summarizes the obtained P-Values for each of the three factors and their interactions. All datasets and all feature extraction algorithms (texture and curvature) were used, for $3k5$. The most important factor to our experiments is *Training+Model* which obtained a p-Value of 0%, suggesting leaf side significance on both training and testing. Notice also that *Training+Model* is significant together with *Algorithm*, which means that some feature extraction algorithms may work better or worse depending on the leaf side images used for training and testing.

Table 5.3 summarizes the results of running the Tukey test for *Training+Model*. The mean is computed over the accuracy obtained for all feature extraction algorithms and $3 \leq k \leq 5$, for each *Training+Model* level.

Table 5.2. Global GLM results at a 95% confidence. R-sq = 99.96%

Source	P-Value
k	0.000
Algorithm	0.000
Training+Model	0.000
*k*Algorithm*	0.000
*k*Training+Model*	0.022
*Algorithm*Training+Model*	0.000

Table 5.3. Tukey Pairwise Comparisons at a 95% confidence, for factor *Training+Model*

Training+Model	Accuracy Mean	Grouping
$Test_B + Model_B$	0.80	A
$Test_B + Model_C$	0.79	B
$Test_C + Model_C$	0.76	C
$Test_F + Model_C$	0.74	D
$Test_F + Model_F$	0.73	E
$Test_B + Model_F$	0.37	F
$Test_F + Model_B$	0.31	G

Group A, which uses $Test_B$ (tested with back images) and $Model_B$ (trained with back images), achieves the best average accuracy. Group B, which is closely related to Group A, but slightly inferior, also uses $Test_B$ for testing, but the combined $Model_C$ for training. This suggests that globally, the identification of back side images P is better than when P is a front side image (except if the model used is $Model_F$).

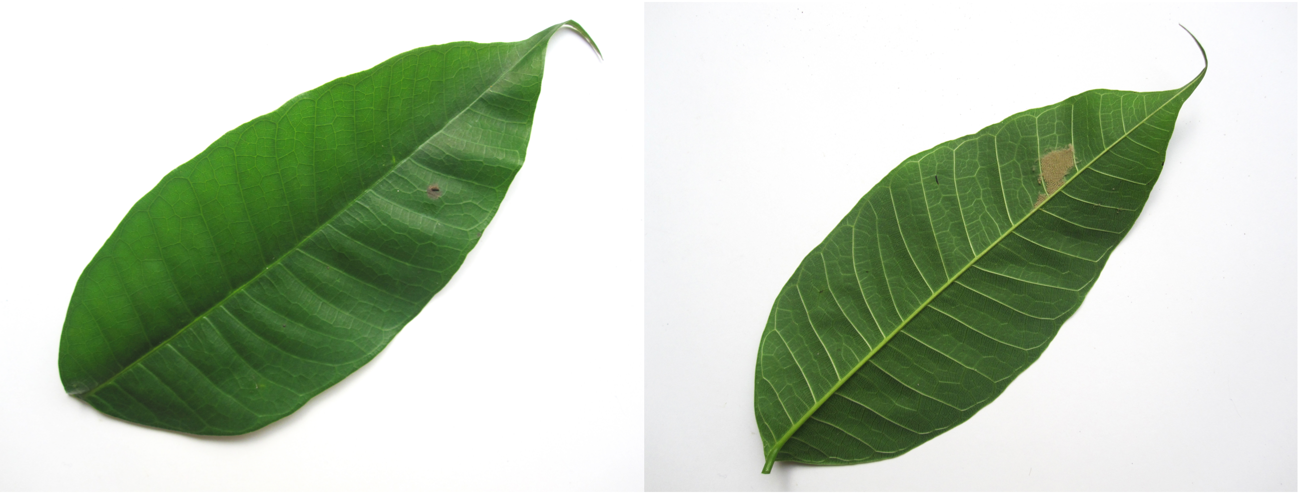
It is interesting to note that when P is a front side image, the combined $Model_C$ is slightly better than using a more specialized $Model_F$.

Additionally, it is worth noting that, consistent with intuition, testing with $Test_B$ but training with $Model_F$, and vice-versa, is not a good idea.

For the sake of completeness, we also ran tests for the curvature algorithm alone. Not surprisingly, the worst cases are also $Test_B - Model_F$ and $Test_F - Model_B$, but with a higher accuracy of 67% in both cases. Compared to the Tukey test that contains both curvature and texture in Table 5.3, which was as low as 37%, this 67% is much better. This shows that internal texture patterns differ between leaf sides for classification and that curvature does not suffer as much when one side or the other of the leaf is used.

6.2 Significance of leaf side per species

Table 5.4 shows the results of the GLM applied to each of the 63 species. For 39 species (61.9%) the best accuracy is obtained when back side images P are used. For 16 species (23.8%) the highest accuracy is obtained when P is a front side image. Finally, for 9 species (14.2%) there is no clear winner. This means that a large group of species are better classified when P is a back side image, but there is also another group of species that have better results when P is a front side image. In the context of a software tool, this individualized analysis is important for use cases in which the user is trying to determine if image P corresponds to a given species. For example, if we want to determine if P is an image of *Brosimum alicastrum*, we may get better accuracy in the automatic identification if P is a back side image and the model was trained



(a) Front side image of a *Brosimum alicastrum* sample. **(b)** Back side image of a *Brosimum alicastrum* sample.

Figure 5.1. Difference between sides of the same leaf specimen of *Brosimum alicastrum*.

with a dataset of back side images (although a general $Model_C$ would not be too bad). However, if we want to determine if P is an image of *Quercus insignis*, we may get better accuracy if P is a front side image and the model was trained with a dataset of front side images (although a general $Model_C$ would not be too bad either).

A visual example of the difference between leaf sides is shown in Figure 5.1 for species *Brosimum alicastrum*. Visually, both images show how images of the leaf side of a single individual differ. For this particular species, the accuracy ranges from 91% for the back side subset, down to 74% for the front side subset, according to Table 5.4. For the combined or complete dataset the obtained accuracy is 80%.

7 Conclusions

For the tested dataset, leaf side is a significant factor for automatic plant species identification. On the average, and for most cases, underside pictures lead to more accurate identifications. For most species (61.9%), classification is better if the sample P to be identified is a back side leaf image; in a smaller number of cases (23.8%) a front side image P gives better results.

In agreement with intuition, the worst accuracy is obtained when the model is trained with back side images and tested with front side images and vice-versa.

However, it should be noticed that the above conclusions are due to the differences in texture displayed in the back and front sides of leaves. Because the curvature of the front side and the back side of a leaf are mirror images of each other, this feature is not sensibly affected by which side of the leaves are used. Thus, tools based on curvature analysis alone such as LeafSnap (Kumar et al. 2012) would not be affected by the indiscriminate use of leaf front and back side images.

8 Future Work

Other feature extraction algorithms such as point of interest should undergo a similar type of analysis. Additionally, it is important to understand if different leaf regions such as the apex, base, or petiole have

Table 5.4. Accuracy mean per species for the *Training+Model* factor. Highlighted values belong to the most significant group according to the Tukey tests

Species	$Test_B + Model_B$	$Test_B + Model_C$	$Test_F + Model_F$	$Test_F + Model_C$	$Test_C + Model_C$
Acnistus arborescens	0.75	***0.80***	0.69	0.77	0.79
Aegiphila valerioi	0.63	***0.64***	0.47	0.55	0.6
Annona mucosa	***0.64***	***0.60***	0.47	0.5	0.55
Ardisia revoluta	***0.75***	***0.75***	0.55	0.6	0.68
Blakea maurofernandeziana	0.94	***0.98***	0.81	0.81	0.9
Brosimum alicastrum	***0.91***	***0.90***	0.74	0.69	0.8
Calophyllum brasiliense	***0.87***	***0.85***	0.68	0.67	0.76
Calycophyllum candidissimum	***0.83***	***0.83***	0.71	0.68	0.75
Cestrum tomentosum	***0.80***	***0.80***	0.75	0.77	0.78
Citharexylum donnell-smithii	0.77	***0.82***	0.68	0.67	0.75
Clethra costaricensis	***0.75***	0.65	0.64	0.7	0.67
Clusia croatii	0***.95***	0.89	0.81	0.76	0.83
Coccoloba floribunda	***0.64***	***0.63***	0.49	0.52	0.58
Cordia eriostigma	***0.68***	***0.67***	0.55	0.55	0.61
Croton draco	***0.98***	***0.87***	0.74	0.77	0.82
Croton niveus	***0.85***	***0.84***	0.79	0.79	0.81
Dalbergia retusa	***0.85***	***0.80***	0.65	0.65	0.73
Ficus cotinifolia	***0.91***	0.87	0.85	0.87	0.87
Guazuma ulmifolia	***0.93***	***0.93***	0.85	0.83	0.88
Hyeronima alchorneoides	***0.8***	***0.81***	0.71	0.71	0.76
Manilkara chicle	***0.86***	***0.85***	0.75	0.78	0.81
Ocotea sinuata	0.84	***0.86***	0.83	0.82	0.84
Persea americana	***0.78***	***0.76***	0.64	0.62	0.69
Pimenta dioica	***0.9***	***0.9***	0.58	0.58	0.74
Platymiscium pinnatum	***0.70***	***0.71***	0.6	0.57	0.64
Posoqueria latifolia	***0.72***	***0.66***	0.48	0.5	0.58
Quercus corrugata	***0.97***	0.95	0.85	0.88	0.91
Robinsonella lindeniana var. divergens	***1***	***1***	0.93	0.94	0.97
Sapium glandulosum	***0.82***	***0.80***	0.73	0.73	0.76
Sideroxylon capiri	***0.80***	0.76	0.54	0.52	0.65
Simarouba glauca	***0.97***	***0.95***	0.65	0.63	0.79
Swietenia macrophylla	***0.73***	***0.71***	0.67	0.65	0.68
Tabebuia ochracea	***0.79***	***0.82***	0.67	0.65	0.73
Tabebuia rosea	***0.83***	***0.83***	0.5	0.5	0.66
Tabernaemontana litoralis	***0.77***	***0.77***	0.67	0.67	0.72
Terminalia amazonia	0.86	***0.89***	0.84	0.83	0.86
Terminalia oblonga	***0.81***	***0.81***	0.58	0.64	0.72
Trichilia havanensis	***0.77***	0.67	0.68	0.7	0.68
Vernonia patens	***0.94***	0.93	0.87	0.84	0.89
Anacardium excelsum	0.72	0.75	0.75	***0.80***	0.78
Bauhinia purpurea	0.83	0.84	0.85	***0.87***	0.85
Colubrina spinosa	0.69	0.7	***0.89***	***0.88***	0.79
Dendropanax arboreus	0.53	0.47	***0.56***	***0.57***	0.52
Dipteryx panamensis	0.69	0.68	0.73	***0.76***	0.72
Eugenia hiraeifolia	0.83	0.79	0.87	***0.95***	0.87
Genipa americana	0.56	0.59	0.69	***0.80***	0.7
Heliocarpus appendiculatus	0.84	0.86	0.89	***0.97***	0.92
Hymenaea courbaril	0.61	0.62	***0.82***	***0.82***	0.72
Pachira quinata	0.79	0.76	***0.8***	0.78	0.77
Platymiscium parviflorum	0.65	0.61	0.62	***0.7***	0.65
Quercus insignis	0.8	0.82	***0.94***	***0.93***	0.87
Samanea saman	0.8	0.78	***0.9***	0.86	0.82
Stemmadenia donnell-smithii	0.35	0.38	***0.75***	***0.73***	0.57
Urera caracasana	0.94	0.94	***0.97***	0.85	0.94
Astronium graveolens	***0.84***	***0.85***	0.77	***0.83***	***0.84***
Erythrina poeppigiana	0.65	***0.7***	0.68	***0.71***	***0.7***
Hura crepitans	***0.83***	***0.84***	0.79	***0.84***	***0.84***
Psidium guajava	***0.75***	0.66	***0.73***	0.65	0.66
Solanum rovirosanum	0.68	***0.70***	0.65	***0.69***	***0.69***
Tabebuia impetiginosa	***0.82***	0.8	0.84	***0.83***	0.82
Bauhinia ungulata	0.79	0.79	0.81	0.79	0.79
Cedrela odorata	0.89	0.87	0.87	0.91	0.89
Muntingia calabura	0.95	94	0.93	0.93	0.94

significant features. Understanding this could also help in classifying species even when the leaf is partially damaged or only a portion of it is available. Because the results of this type of research are affected by the subset of species used, it is very important to create a national level or global level leaf image dataset with as many species as possible. As more leaf image data becomes available, analysis by geographic regions, higher level taxa, special interest taxa (such as endangered species and species of economic interest), and other groups would be extremely useful for biodiversity conservation. Also, as more leaf data are gathered and made available, approaches such as ConvNets (Simard et al. 2003) would be more feasible for identification even with complex backgrounds.

Acknowledgement

Thanks to the National Museum of Costa Rica and to Jose David Sánchez for their help in setting up the $Train_F$ and $Train_B$ image datasets. Additional thanks to Nelson Zamora and the National Biodiversity Institute of Costa Rica (INBio) for collecting and identifying the leaf samples that are included in the complete dataset $Train_C$. This image dataset is available under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International license upon request to the authors.

Chapter 6

Automated Plant Species Identification: Challenges and Opportunities

Reference Erick Mata-Montero and Jose Carranza-Rojas (2016). “Automated Plant Species Identification: Challenges and Opportunities”. In: *ICT for Promoting Human Development and Protecting the Environment: 6th IFIP World Information Technology Forum, WITFOR 2016, San José, Costa Rica, September 12-14, 2016, Proceedings*. Springer International Publishing, pp. 26–36. ISBN: 978-3-319-44447-5. DOI: 10.1007/978-3-319-44447-5_3

Keywords Biodiversity Informatics, Computer Vision, Image Processing, Machine Learning, Leaf Recognition, Plant Identification, Citizen-Science, Species Identification, Cybertaxonomy

1 Abstract

The number of species of macro organisms on the planet is estimated at about 10 million. This staggering diversity and the need to better understand it led inevitably to the development of classification schemes called biological taxonomies. Unfortunately, in addition to this enormous diversity, the traditional identification and classification workflows are both slow and error-prone; classification expertise is in the hands of a small number of expert taxonomists; and to make things worse, the number of taxonomists has steadily declined in recent years. Automated identification of organisms has therefore become not just a long time desire but a need to better understand, use, and save biodiversity. This paper presents a survey of recent efforts to use computer vision and machine learning techniques to identify organisms. It focuses on the use of leaf images to identify plant species. In addition, it presents the main technical and scientific challenges as well as the opportunities for herbaria and cybertaxonomists to take a quantum leap towards identifying biodiversity efficiently and empowering the general public by putting in their hands automated identification tools.

2 Introduction

The word "biodiversity" is a synonym of "biological diversity". The Convention on Biological Diversity (CBD) defines biodiversity as: "the variability among living organisms from all sources including, inter alia, terrestrial, marine and other aquatic ecosystems and the ecological complexes of which they are a part; this includes diversity within species, between species, and of ecosystems." [1] Therefore, there are three levels of biodiversity: intra-specific (genetic), inter-specific, and ecosystemic. Even though a full understanding of all three levels is indispensable to guide biodiversity conservation efforts, this paper focuses on inter-specific biodiversity and some associated taxonomic challenges.

The CBD Strategic Plan 2011-2020 has explicitly stated twenty ambitious targets known as the Aichi Targets [2]. Aichi Target 19 specifically proposes that "knowledge, the science base and technologies relating to biodiversity, its values, functioning, status and trends, and the consequences of its loss, are improved, widely shared and transferred, and applied"; but in fact biodiversity informatics will be fundamental to the achievement of all of the Aichi Targets.

It is estimated that about 10 million species of macro organisms inhabit the earth. This vast inter-specific biodiversity and the need to better understand it led to the development of classification schemes called biological taxonomies. Even since Aristotle's times, when only approximately 500 species of animals had been identified, Aristotle himself established a classification method. In the XVIII century, Carl Linnaeus "father of modern taxonomy" formalized a system of naming organisms called binomial nomenclature which is used to this day.

Unfortunately, in addition to the enormous biodiversity of the earth, current identification and classification workflows are both slow and error-prone. Furthermore, classification expertise is in the hands of a small, decreasing number of expert taxonomists. This has been identified as a serious problem and is known as the "global taxonomic impediment" [3]. Automated identification of organisms has therefore become not just a dream among systematists for centuries (MacLeod 2007) but a need to better understand, use, and save biodiversity.

Even though the number of plant species (about 400,000) is considerably smaller than the number of animal species, taxonomic work on them is still a monumental task. However, plant species identification is particularly important for biodiversity conservation. It is critical to conduct studies of biodiversity richness of a region, monitoring of populations of endangered plants and animals, climate change impact on forest coverage, bioliteracy, payment for environmental services, and weed control, among many other major challenges.

The rest of this paper is organized as follows: Section 3 summarizes progress made to automate the identification of taxa in systematics. It starts with a description of the traditional dichotomous keys approach, and then presents interactive keys, morphometric approaches, briefly describes DNA barcoding, and concludes with recent approaches based on machine learning and computer vision techniques. Section 4 summarizes the state of the art of leaf-based plant species identification using computer vision. Finally, Section 5 concludes with current challenges and opportunities.

3 Automated Taxon Identification in Systematics

Traditionally, systematists have not relied on quantitative data alone to identify taxa. They prefer the visual inspection of morphology, the (mostly) qualitative assessment of characters, and the comparison of these to reference specimens and/or images. While this process works, it is not quick, efficient or reliable (MacLeod

[1] https://www.cbd.int/convention/articles/default.shtml?a=cbd-02

[2] https://www.cbd.int/sp/targets/

[3] https://www.cbd.int/gti/

2007). The following subsections describe some attempts to automate or at least define algorithms that can be followed either manually (e.g., dichotomous keys and morphometrics) or translated into software that fully or partially automates the taxa identification process. In some cases, the resulting software guides a human user (e.g., interactive keys) who actually makes the decisions. In other cases, it fully automates the taxa identification process by extracting additional data from specimens (e.g., molecular and chemical data) or multimedia information such as digital images and sound.

3.1 Single-access keys

In biology, an *identification key* is a document or software that takes the form of a decision tree that offers a fixed sequence of identification steps. If each step has only two alternatives, the key is said to be *dichotomous*, otherwise it is *polytomous*. These keys are possibly the oldest attempt to designing algorithms for organismal identification long before computers were available. They aim at reducing the rate of errors, making explicit and objective the rules to be followed, and selecting optimal or semi-optimal sequences of questions.

This approach has several drawbacks even when those algorithms have been programmed. Among them are the difficulty to accommodate new species descriptions and the assumption that a user has all the information available to proceed from the top question (the *single-access key*) to the following levels. The latter means that when only partial information is available about the organism (e.g., only leaves or flowers of a plant), a user might not be able to go past the very first question.

3.2 Multiple-access keys

These are decision trees that have multiple starting points that allow users to follow different paths, possibly because he/she has partial morphological information. In its computerized version, they are also called *interactive keys*. They start with a full domain of candidates (e.g., all plants from a country), and proceed to gradually discard candidates as the user proceeds answering questions in an arbitrary order. The final result could be a unitary set of candidates (full identification achieved), an empty set (a new species or an incomplete key), or a set with cardinality greater than 1 (some questions remain to be answered).

3.3 Morphometric approaches

Morphometrics is the study of shape variation and its co-variation with other variables (Bookstein 1997). Three general approaches are usually distinguished: traditional morphometrics, landmark-based morphometrics and outline-based morphometrics. Traditional morphometrics is the application of multivariate statistical analysis to sets of quantitative variables such as length, width, and height. Geometric morphometrics emphasizes methods that capture the geometry of the morphological structures of interest and preserve this information throughout the analyses. Outline-based morphometrics focuses on shape variation along the contour of an object. These three approaches are not necessarily mutually exclusive. Adams et al. 2004 provide an excellent survey on this subject.

3.4 DNA barcoding

DNA barcoding is a taxonomic method that uses a short genetic marker in an organism's DNA to identify it as belonging to a particular species (Hebert et al. 2003). The gene region that is being used as the standard barcode for almost all animal groups is a 648 base-pair region in the mitochondrial cytochrome c oxidase 1 gene ("CO1"). For plants, two gene regions in the chloroplast, matK and rbcL, have been approved as the barcode region. DNA barcoding has met with a strong reaction from scientists, especially systematists, who either express their enthusiastic support or vehement opposition (Ebach et al. 2010; Rubinoff et al. 2006). The current trend appears to be that DNA barcoding should be used alongside traditional taxonomic tools and alternative forms of molecular systematics so that problem cases can be identified and errors detected.

3.5 Crowd sourcing (collective intelligence)

Crowd sourcing approaches to species identification is neither a quantitative nor an automated method. However, it is included it in this survey because it uses computer technology to gather georeferenced multimedia information (e.g., images) and a community of citizen scientists and biologists who jointly tackle the challenge of identifying an organism based on an image, collective knowledge, and interactive keys or other forms of computer-based tools. Besides, it is a low-cost high impact approach to empower and engage the general public in cibertaxonomy and biodiversity conservation. iNaturalist [4] and Pl@ntNET [5] (Joly et al. 2015a) are two excellent examples of this approach. On the negative side, high levels of quality control are imperative because the community involved does not necessarily comprise domain experts.

3.6 Computer vision and machine learning

In spite of enormous progress in the application of computer vision algorithms in other areas such as medical imaging, OCR, and biometrics (Andreopoulos et al. 2013), only recently have they been applied to identify taxa. Images of plant leaves and insect wings have been particularly attractive because they are flat and their morphology is used in most identification keys. Thus, in the last decade, research in computer vision has produced algorithms to help botanists and non-experts classify plants based on images of their leaves (Bhardwaj et al. 2013; Herdiyeni et al. 2012; M. Z. Rashad 2011; R.D et al. 2011; Wu et al. 2007). However only a few studies have resulted in efficient systems that are used by the general public, such as LeafSnap (Kumar et al. 2012).

Computer vision and machine learning are two highly related artificial intelligence fields. In a *supervised learning* scenario, the general approach for organismal identification using computer vision comprises two general steps. First, digital images of identified species are fed to an algorithm that cleans them, segments them, and extracts relevant features. As a result, source images are typically transformed from the bitmap domain to a more tractable domain (e.g., histograms) and stored in a *training dataset* D. The second step consists of using the training dataset D to train an algorithm A. *Unsupervised learning* (e.g., cluster analysis) can also be used when a dataset of images is available but the associated species have not been identified.

Once algorithm A has been trained and tested, it is ready to try to identify species based on images of organisms. In the typical scenario, algorithm A has two inputs, namely, an image I of the unidentified organism and the dataset D. Algorithm A applies to image I the same filters used to create the dataset D and outputs a ranking of k *candidate species*. The larger the number k is, the better the chance of including

[4] http://www.inaturalist.org

[5] http://www.plantnet-project.org/page:projet?langue=en

the correct identification in the ranking is. However, most users would expect k to be a small value to be useful. Details on the use of computer vision and machine learning to identify plants based on images of leaves are presented in the following section.

4 Automated Leaf-based Plant Species Identification

Several surveys regarding leaf-based identification of plants have been published in the past. *Survey on Techniques for Plant Leaf Classification* covers most classification methods such as kNN, PNN, and SVM, as well as their accuracy and precision. In (Vishakha Metre 2013), Metre and Ghorpade survey different texture-only techniques, provide a comparison schema for them, and pinpoint how important it is to create a centralized dataset of leaf images.

Most researchers agree on a general workflow to identify species based on images of their leaves (Bhardwaj et al. 2013; Herdiyeni et al. 2012; M. Z. Rashad 2011; R.D et al. 2011; Wu et al. 2007). The first step is data acquisition. Acquiring leaf images is a time consuming task. Because of the lack of standards and centralized repositories, researchers have typically generated isolated datasets for their projects. *Segmentation* of the leaf is then executed to explicitly separate leaf from non-leaf pixels. Afterwards, different techniques are used to extract features based on venation (Li et al. 2006), curvature (Kumar et al. 2012) and morphometrics (Bhardwaj et al. 2013). Finally, machine learning techniques are used to generate the trained algorithm (Bhardwaj et al. 2013; Herdiyeni et al. 2012; M. Z. Rashad 2011; R.D et al. 2011; Wu et al. 2007).

4.1 Data Acquisition

Existing leaf recognition datasets use images of individual leaves on uniformly colored backgrounds for easier leaf segmentation. There are several datasets publicly available but, to our knowledge, there is not yet a centralized dataset which can grow as researchers and citizen scientists add more images and data. The following are examples of datasets from different projects:

- The Flavia Dataset (Wu et al. 2007) encompasses 32 species and a total of 3,621 fresh leaf images on white backgrounds. Leaves were collected in Nanking, China.

- Kumar et al. 2012 created a dataset for 184 tree species from Northeastern USA that includes 23,916 images of fresh leaves with uniform backgrounds. It is used by the LeafSnap mobile app.

- Mata-Montero and Carranza-Rojas (Mata-Montero et al. 2015) from the Costa Rica Institute of Technology created a dataset that comprises 2,345 noisy and 1,468 clean leaf images from 67 Costa Rican tree species, all with uniform background.

- ImageCLEF is a leaf classification competition that has created its own dataset (Joly et al. 2015a). It currently includes 1,000 plant species from West Europe. It has more than 100,000 images of leaves, as well as flowers, fruits, stem and the whole plant pictures. It comprises both images with white background and images taken directly in the field with complex backgrounds and noise (Joly et al. 2015a).

4.2 Leaf Segmentation

Leaf segmentation can act on images with uniform backgrounds, such as a white piece of paper, or complex backgrounds. The former is simpler although artifacts such as shadows and light gradients still generate some problems. Most researchers use uniform backgrounds to simplify this phase. In leafsnap, 7360026 EM is used to cluster pixels. This produces fairly good segmentation but shadows tend to generate false positives. Similarly, in (Soares et al. 2013) the authors study how a semi-controlled light environment affects clustering algorithms. They perform color clustering and then apply Grab-Cut to find the global optimal segmentation solution.

Very few studies have tackled the problem of segmenting leaves with complex backgrounds (Cerutti et al. 2013; Le et al. 2015). This feature is highly desirable for at least two types of leaves: leaves of tall trees from which it is difficult to take a sample and then photograph it with a uniform background, and leaves of plants that have been mounted on herbarium sheets. In the former case, it would be ideal to zoom-in with the camera and take a picture of the leaf in its tree. In the latter, the background may not be as complex as a natural setting but overlapping of leaves and other plant elements in the herbarium sheet makes the automated extraction of leaves and their subsequent segmentation very challenging.

We are not aware of any research that aims at generating leaf image datasets from herbarium sheets. The benefit of doing this is twofold. First of all, herbaria all over the world have invested millions of dollars over long periods of time to collect samples of plants. Rather than going again to the field to take pictures or collect more samples, it would be considerably less expensive to use leaves of plants that have already been identified and conserved in herbaria. Secondly, it would help demonstrate the value of herbaria collections.

4.3 Feature Extraction and Identification

Segmentation of the input image I produces a segmented image I' to which feature extraction is applied. This subsection briefly surveys approaches that use curvature, texture, venation, leaf morphometrics, or combinations of them.

Curvature.

Kumar et al. 2012 create what they call a HCoS, which consists on measuring the leaf area and arc length of the intersection of the leaf and disks of radius r, where $1 <= r <= 25$ pixels, and the disks are centered at every leaf contours pixel of the leaf in I'. All calculations are then added into a unique histogram that describes the contour of the leaf. Using kNN and histogram intersection, a list of the k species whose leaves more closely match the leaf in I is presented to the user. Another method applied on both simple and complex leaves is the one described in (Zhao et al. 2015). Their method captures both global and local shape features and uses them separately during identification. This allows to discriminate leaves with similar shape but different margin patterns, and viceversa. Similarly to Kumar et al. 2012, several scales are explored by convolving the contour against a Gaussian filter with different values σ. This is particularly useful for serration of the margin.

Texture.

LBP descriptors are used in (Herdiyeni et al. 2013) to identify medicinal and house plants from Indonesia. Different LBP descriptors were extracted from different sample points and radius, and concatenated into histograms. Then a four layer PNN classifier was used. For complex background images the achieved precision was 77% and for uniform background images 86.67%. In (Nguyen et al. 2013) SURF features were used to develop an Android application for leaf recognition. The reported precision was 95.94% on the Flavia dataset (Wu et al. 2007). In (M. Z. Rashad 2011) authors identify plants based only on a portion of the leaf, allowing botanists to identify damaged plants. The reported precision is 98.7% when using ANN for classification on their own small dataset.

Venation.

Very few studies have used venation extraction as the basis for taxa identification. Venation extraction is not trivial, since veins are often merged with other leaf features. Some authors have simplified the task by using special equipment or treatments that render images with more clearly identified veins (Lu et al. 2012; Sun et al. 2011). However, this defeats the goal of having users get an automated identification for specimens that they have photographed with ordinary digital cameras.

In (Sun et al. 2011), vein pixels are extracted from laser scanned images in 3D. The laser scans a 3D point cloud in which veins are 3D-convex. A curvature threshold is then used to obtain potential vein pixels. Finally a squared linear fitting is applied to approximate the vein contour lines. In (Li et al. 2006) researchers developed a tool to help botanists extract veins of leaves with minimum human interaction. They used a patch-based approach where a set of linear functions are learned from patches of images containing veins using Independent Component Analysis (ICA). Then these learned functions are used as a pattern map for vein detection.

Leaf morphometrics.

Leaves display very rich morphology. Traditional leaf measurements include aspect ratio, leaf area, rectangularity, circularity, convexity, and solidity, among others (Bhardwaj et al. 2013). Additionally, color moments for gray scale intensities such as mean, variance, kurtosis, skewness have also been used (Bhardwaj et al. 2013). Traditional, landmark-based, and outline-based morphometrics have been used both separately and in combined form.

Multimodal approaches.

In (R.D et al. 2011), a multimodal system composed of 38 morphological features and a Principal Component Analysis (PCA) approach for texture were used. The PCA training phase took all the dataset pictures and put them in a matrix, where a small number of characteristic features called eigenpictures were generated. Then, each image was represented as a linear combination of these eigenpictures. Their reported precision on the Flavia dataset (Wu et al. 2007) for the morphological features was 91.9%, for the PCA algorithm 85.4%, and for both combined 89.2%.

In (Herdiyeni et al. 2012) a combination of shape, texture and color was used to recognize Indonesian medicinal plants. As a classifier they used PNN with a reported precision of 72.16% over 51 medicinal species, with a total of 2,448 images. The authors created a mobile app which runs on Android OS called Medleaf (Herdiyeni et al. 2012). Their best precisions were achieved by using LBPV as a feature base and not morphological features.

In (Mata-Montero et al. 2015) texture extraction of the whole leaf using LBP was compared with the HCoS curvature method developed by Kumar et al. 2012. In the experiments it was proved that texture is more resilient to noise on leaf images. Better accuracy was achieved by assigning a small importance factor to curvature (10%) and a larger one to texture (90%). This result also matches results of (Herdiyeni et al. 2013) with regard to the usefulness of LBP for identification based on images of damaged leaves.

Deep Learning Approaches.

Deep learning has become a huge success in computer vision research (Krizhevsky et al. 2012). In (Lee et al. 2015) a CNN was applied to a dataset with 44 species. The CNN was not coded with layers for specific features (e.g., curvature or texture), but the authors could infer that a layer was related to shape/curvature and another one to patterns similar to texture/venation. With this interpretation, the authors conclude that shape/curvature is not as discriminating as texture/venation, which is consistent with Mata-Montero et al. 2015.

5 Challenges and Opportunities

Biodiversity conservation presents several monumental challenges. At the political and management level, it requires information and a deep understanding of living nature. However, about 80% of the organisms on the planet do not even have a name. The scientific task of naming and classifying those organisms is gigantic, not only because of the large number of species to identify and describe, but also because it is tedious, slow, and error-prone. The global taxonomic impediment adds to the complexity of these challenges. Finally, access to this knowledge is limited by the scientific and non-digital nature of large amounts of literature.

Fortunately, computer vision and machine learning techniques that have been very effective in other realms are now being used to identify organisms, in particular plants, with high levels of accuracy (90% or more). This could have an important impact in concrete conservation actions such as control of trade of endangered species and the execution of rapid biodiversity inventories. The following paragraphs, summarize some opportunities we currently have to cope with the above mentioned challenges.

Building a global dataset: Global biodiversity informatics initiatives such as GBIF [6], EOL [7], and BHL [8], have successfully built large global databases of biodiversity information that is freely available on the web. GBIF currently provides more than 600 million specimen-level records, EOL over a million species level descriptions, and BHL more than 50 million pages of literature. An analogous dataset of digital images of plant elements (e.g., leaves) does not exist. However, there are several opportunities that should be taken. First of all, digital cameras are now very inexpensive and powerful. Secondly, even though data sharing protocols and standards need to be in place, organizations such as TDWG [9] are devoted to precisely this endeavor. Finally, crowd sourcing offers now excellent opportunities to both, generate large repositories of information, and raise awareness of the general public through citizen science projects. iNaturalist and Pl@ntNET (Joly et al. 2015a) have been very successful and deserve being emulated. The PlantCLEF dataset already demonstrates that this can be done at the European level.

Work with Herbaria: Herbaria hold treasures of information that should be critical to scale up the size and impact of a global dataset of digital images of elements of plants. Herbaria maintain large collections of plants that have been carefully mounted on sheets, could be digitized, and whose elements (e.g., leaves) could be extracted to feed a global dataset. Because herbaria sheets contain juxtaposed leaves, flowers,

[6] http://www.gbif.org

[7] http://www.eol.org

[8] http://www.bhl.org

[9] http://www.tdwg.org

and other plant elements, research on detection and extraction of leaves needs to be further developed. In addition, more research is needed to deal with noisy images, complex backgrounds, damage detection and *digital image repair*, along with leaf identifications based on portions of the leaf (in case it is damaged). Landmark-based morphometrics research should help with the latter. Finally, as a very important herbaria financial sustainability side effect, herbaria around the world would have more arguments to demonstrate the value and impact of maintaining and investing in their collections. However, it is very critical for herbaria to supplement their collections with digital images through crowd sourcing and changes in their traditional workflows.

Deep Learning: Deep Learning, particularly using CNNs, is a very hot topic in computer vision. The exciting results obtained in events such as ImageNet (Krizhevsky et al. 2012) have generated a lot of expectation. As more data and computational power are now available, this technique has become the most widely used, without substantial algorithmic changes since its inception. Instead of following a gradual path that aims at using images of elements of an organism first (e.g., leaves or flowers of a plant), and then pictures of the whole organism, CNN tackles directly the challenge of identifying organisms by using pictures of the whole or parts of the organism. However, this approach has at least two important limitations. First, it tends to work better with very large sets of images (Simard et al. 2003). Secondly, it lacks the explanatory power of other approaches such as landmark-based morphometrics. Nevertheless, as global data sets are developed, it is just a matter of time to overcome the former. Additionally, research work is already under way to overcome the latter (Lee et al. 2015).

Chapter 7

Going Deeper in the Automated Identification of Herbarium Specimens

Reference Jose Carranza-Rojas, Herve Goeau, Pierre Bonnet, Erick Mata-Montero, and Alexis Joly (2017b). "Going deeper in the automated identification of Herbarium specimens". In: *BMC Evolutionary Biology* 17.1, p. 181. ISSN: 1471-2148. DOI: 10.1186/s12862-017-1014-z. URL: https://doi.org/10.1186/s12862-017-1014-z

Keywords Biodiversity Informatics, Computer Vision, Deep Learning, Plant Identification, Herbaria

1 Abstract

Background Hundreds of herbarium collections have accumulated a valuable heritage and knowledge of plants over several centuries. Recent initiatives started ambitious preservation plans to digitize this information and make it available to botanists and the general public through web portals . However, thousands of sheets are still unidentified at the species level while numerous sheets should be reviewed and updated following more recent taxonomic knowledge. These annotations and revisions require an unrealistic amount of work for botanists to carry out in a reasonable time. Computer vision and machine learning approaches applied to herbarium sheets are promising but are still not well studied compared to automated species identification from leaf scans or pictures of plants in the field.

Results In this work, we propose to study and evaluate the accuracy with which herbarium images can be potentially exploited for species identification with deep learning technology. In addition, we propose to study if the combination of herbarium sheets with photos of plants in the field is relevant in terms of accuracy, and finally, we explore if herbarium images from one region that has one specific flora can be used to do transfer learning to another region with other species; for example, on a region under-represented in terms of collected data.

Conclusions This is, to our knowledge, the first study that uses deep learning to analyze a big dataset with thousands of species from herbaria. Results show the potential of Deep Learning on herbarium species identification, particularly by training and testing across different datasets from different herbaria. This could potentially lead to the creation of a semi, or even fully automated system to help taxonomists and experts with their annotation, classification, and revision works.

2 Introduction

For several centuries, botanists have collected, catalogued and systematically stored plant specimens in herbaria. These biological specimens in research collections provide the most important baseline information for systematic research (Tschöpe et al. 2013). These physical specimens ensure reproducibility and unambiguous referencing of research results relating to organisms. They are used to study the variability of species, their phylogenetic relationship, their evolution, and phenological trends, among others. The estimated number of specimens in Natural History collection is in the 2–3 billion range (Duckworth et al. 1993). There are approximately 3,000 herbaria in the world, which have accumulated around 350,000,000 specimens (Thiers 2017), *i.e.*, whole plants or plant parts usually in dried form and mounted on a large sheet of paper.

Large scale digitization of specimens is therefore crucial to provide access to the data that they contain (Ellwood et al. 2015). Recent national and international initiatives such as iDigBio (*iDigBio* 2017) or e-ReColNat started ambitious preservation plans to digitize and facilitate access to herbarium data through web portals accessible to botanists as well as the general public. New capacities such as specimen annotation (Suhrbier et al. 2017) and transcription (Mononen et al. 2014) are offered in these portals. However, it is estimated that more than 35,000 species not yet described and new to science have already been collected and are stored in herbaria (Bebber et al. 2010). These specimens, representing new species, remain undetected and undescribed because they may be inaccessible, their information is incomplete, or the necessary expertise for their analysis is lacking. These new species are then unnoticed, misplaced, or treated as unidentified material. Thousands and thousands of sheets are still not identified at the species level while numerous sheets should be reviewed and updated following more recent taxonomic knowledge. These annotations and revisions require such a large amount of work from botanists that it would be unfeasible to carry them out in a reasonable time.

Computer vision approaches based on the automated analyses of these sheets may be useful for such species identification tasks. Furthermore, such automated analysis could also help botanists in the processes of discovering and describing new species among the huge volume of stored herbarium specimens. As a result, evolutionary and ecological studies could be strongly accelerated due to the quick access to the most interesting specimens of a particular group of species. A tool that, based on herbarium sheet images across multiple collections world wide, finds the plant specimens more similar to a candidate would be of great help for taxonomists and botanists working at herbaria. However, this is still a very challenging objective. Because specimens are mounted on sheets assuming that they will be used and visually inspected by humans, the amount of visual noise present in this type of image is very high for fully automated computer vision processing. Nevertheless, in the last five years, deep learning has become a promising tool to handle extremely complex computer vision tasks. Additionally, online portals of ambitious initiatives such as iDigBio already provide access to more than 14 million herbarium images (Page et al. 2015) that are particularly useful for deep learning approaches (Goodfellow et al. 2016). Thus, it is now possible to use images of herbaria thanks to current advances in machine learning and initiatives such as iDigBio.

With this study we aim to answer three questions: (i) are herbarium images useful for herbaria-only classification using deep learning? (ii) Can a deep learning model learn relevant features from herbarium images and be successfully used for transfer learning to deal with field images? (iii) And finally, can herbarium images from one region of the world, be used for transfer learning on a herbarium dataset from another region, especially for a region under-represented in terms of collected data?

The following are the main contributions of this research:

- New datasets of herbaria properly curated for machine learning purposes, including one small dataset (255 species, 7.5$\sim$k images) and one large dataset (1,204 species, 260$\sim$k images).

- Demonstration of the feasibility of implementing an identification system for herbarium data at a realistic scale, *i.e.*, with 30 times more species than previous studies in the literature (Unger et al. 2016).

- Experiments to study the usage of herbaria for transfer learning to field photos.
- Demonstration of the potential of using herbaria from one region of the world for transfer learning to another region, with different species.

To our knowledge, this is the first study on the automated analysis of herbarium collections with a large number of sheets and the first one using deep learning techniques. The rest of this manuscript is organized as follows: Section 3 presents relevant related work. Section 4 and Section 5 cover experiment design and the results obtained, respectively. Section 1 presents the conclusions and summarizes future work.

3 Related Work

Among the diverse methods used for species identification, Gaston et al. 2004 discussed in 2004 the potential of automated approaches typically based on machine learning and multimedia analysis methods. They suggested that, if the scientific community is able to (i) overcome the production of large training datasets, (ii) more precisely identify and evaluate the error rates, (iii) scale up automated approaches, and (iv) detect novel species, it will then be possible to initiate the development of a generic automated species identification system that could open opportunities for work in biological and related fields. Since the question raised by Gaston *et al.* ("Automated species identification: why not?"), considerable work has been done on the development of automated approaches for plant species identification, mostly based on computer vision techniques (*e.g.* Casanova et al. 2009; Goëau et al. 2013; Joly et al. 2014a, 2015a; Lee et al. 2015; Wilf et al. 2016; Yanikoglu et al. 2014). A recent and exhaustive review of plant identification using computer vision techniques has been published by Wäldchen et al. 2017. Some of these results were integrated in effective web or mobile tools and have initiated close interactions between computer scientists and end-users such as ecologists, botanists, educators, land managers and the general public. One remarkable system in this domain is the LeafSnap application (Kumar et al. 2012), focused on a few hundred tree species of North America and on the contour of leaves alone. This was followed a few years later by other applications such as Folia (Cerutti et al. 2013) and the popular Pl@ntNet application (Joly et al. 2016a) that now accounts for millions of users all around the world.

However, very few studies have attempted to use herbaria for automated plant classification. So far, most of the biodiversity informatics research related to herbaria has focused on digitization of their collections (Thiers et al. 2016). Wijesingha et al. 2012 use a small dataset of the genus Stemonoporus, endemic to Sri Lanka, that contains a total of 17 species and 79 images. They extracted morphometric features such as leaf length, width, area and perimeter. The reported accuracy for species identification is 85%. Unger et al. 2016 use SVM with Fourier features and morphometric measures to identify species in two test sets, one with 26 species, the other with 17, in each case using 10 images per species, with respective accuracy of 73.21% and 84%. In all these previous studies, the amount of data used was relatively small and restricted to few tens of species. To have more conclusive results and to plan more realistic scenarios, our work focuses on large datasets. Actually, for a given flora from one region, thousands of species can potentially be expected. Therefore, numerous confusions can be encountered not only among species related to a same genus, for instance, but also across genera that share some similar visual patterns.

Besides species identification, some other studies have attempted to automatically extract characters or attributes from herbarium data. It was demonstrated in (Corney et al. 2012) that leaf characters can be automatically extracted using a hand-crafted workflow of state-of-the-art image analysis techniques. It is likely that such ad-hoc workflow would not generalize well to other herbarium data. Moreover, it is not applicable to the other parts of the plant such as flowers, fruits, etc. More recently, Tomaszewski et al. 2016 aimed at determining whether leaf shape changes during the drying process (using elliptic Fourier analysis combined with principal component analysis as well as manual measurements). The results indicate that the preservation process of pressing and drying plants for herbarium purposes causes changes in leaf shape so that they suggest that shape analyses should be performed on datasets containing only one of the leaf types (dried and fresh leaves).

On the deep learning side, Yosinski et al. 2014 study the effects of progressive transfer learning. They conclude that the first layers of the model relate to generic features and help a lot during the transfer itself. However, this is not focused on a particular domain, leaving open the question of how much transfer learning changes if the dataset used for it is from a specific domain or of a similar domain. In particular for plant recognition, it remains to be seen if a very specific domain dataset, such as herbaria, can be used to learn and fine tune with other similar, related datasets, such as field images of plants.

4 Methodology

The following subsections describe the deep learning model used in the experiments, the transfer learning approach, the datasets, and the provisions made to avoid biases and to pre-process all datasets.

4.1 Deep Learning Model

We focused our experiments on the use of Convolutional Neural Networks (CNNs) (LeCun et al. 1995), which have been shown to considerably improve the accuracy of automated plant species identification compared to previous methods (Goëau et al. 2015; Joly et al. 2016b; Wäldchen et al. 2017). More generally, CNNs recently received much attention because of the impressive performance they achieved in the ImageNet classification task (Krizhevsky et al. 2012). The main strength of these technologies comes from their ability to learn discriminant visual features directly from the raw pixels of the images without falling into the trap of the curse of dimensionality, referring to the exponentially increase of the model variables as the dimensionality grows (Goodfellow et al. 2016). This is achieved by stacking multiple *convolutional layers*, *i.e.*, the core building blocks of a CNN. A convolutional layer basically takes images as input and produces as output *feature maps* corresponding to different convolution kernels, while looking for different visual patterns.

Looking at the impressive results achieved by CNNs in the 2015 and 2016 edition of the international PlantCLEF challenge (Goëau et al. 2015; Goëau et al. 2016) on species identification, there is no doubt that they are able to capture discriminant visual patterns of the plants in a much more effective way than previously engineered visual features. In particular, we used an extended version of the GoogleNet model (Szegedy et al. 2015) that is a very deep CNN that stacks several so-called inception layers. We extended the base version with Batch Normalization (Ioffe et al. 2015) which has been proven to speed up convergence and limits overfitting and with a PRELU activation function (He et al. 2015) instead of the traditional RELU.

Table 10.2 shows the modified GoogleNet model with the batch normalization added outside the Inception modules. Just like the original GoogleNet, the model is comprised of several inception modules, however Batch Normalization is added inside each inception module for faster convergence right after each pooling layer. Figure 7.1 shows how the modified Inception module is comprised. The model was implemented by using the Caffe framework (Jia et al. 2014). A batch size of 16 images was used for each iteration, with a learning rate of 0.0075 with images of 224×224 resolution. Simple crop and resize data augmentation was used with the default settings of Caffe.

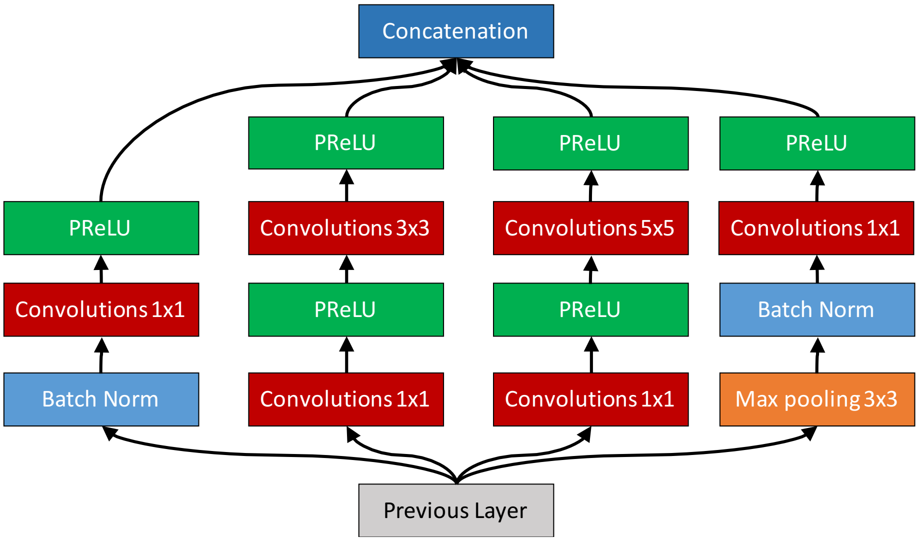


Figure 7.1. Modified Inception module using PRELU and Batch Normalization.

Table 7.1. GoogleNet architecture modified with Batch Normalization.

Type	Patch size / Stride	Output Size	Depth	Params	Ops
convolution	7x7/2	112x112x64	1	2.7K	34M
max pool	3x3/2	56x56x64	0		
batch norm		56x56x64	0		
LRN		56x56x64	0		
convolution	3x3/1	56x56x192	2	112K	360M
max pool	3x3/2	28x28x192	0		
batch norm		28x28x192	0		
LRN		28x28x192	0		
inception (3a)		28x28x256	2	159K	128M
inception (3b)		28x28x480	2	380K	304M
max pool	3x3/2	14x14x480	0		
batch norm		14x14x480	0		
inception (4a)		14x14x512	2	364K	73M
inception (4b)		14x14x512	2	437K	88M
inception (4c)		14x14x512	2	463K	100M
inception (4d)		14x14x528	2	580K	119M
inception (4e)		14x14x832	2	840K	170M
max pool	3x3/2	7x7x832	0		
batch norm		7x7x832	0		
inception (5a)		7x7x832	2	1072K	54M
inception (5b)		7x7x1024	2	1388K	71M
avg pool	7x7/1	1x1x1024	0		
batch norm		1x1x1024	0		
linear		1x1x10000	1	1000K	1M
softmax		1x1x10000	0		

4.2 Transfer Learning

Transfer learning is a powerful paradigm used to overcome the the lack of sufficient domain-specific training data. Deep learning models actually have to be trained on thousands of pictures per class to converge to accurate classification models. It has been shown that the first layers of deep neural networks deal with generic features (Yosinski et al. 2014) so that they are generally usable for other computer vision tasks. Consequently they can be trained on arbitrary training image data. Moreover, the last layers themselves contain more or less generic information transferable from one classification task to another one. These layers are expected to be more informative for the optimization algorithm than a random initialization of the weights of the network. Therefore, a common practice is to initialize the network by pre-training it on a big available dataset and then fine-tune it on the scarcer domain-specific data. Concretely, the methodology we used in our experiment for transferring knowledge from dataset *A* to dataset *B* is the following:

1. The network is first trained from scratch on dataset *A* by using a multinomial logistic regression on top of the SOFTMAX layer and the linear classification layer.

2. The linear classification layer used for dataset *A* is then replaced by a new one aimed at classifying the classes in *B*. It is initialized with random weights.

3. The other layers are kept unchanged so as to initialize the learning of dataset *B* with the weights learned from *A*.

4. The network is trained on the images in *B*.

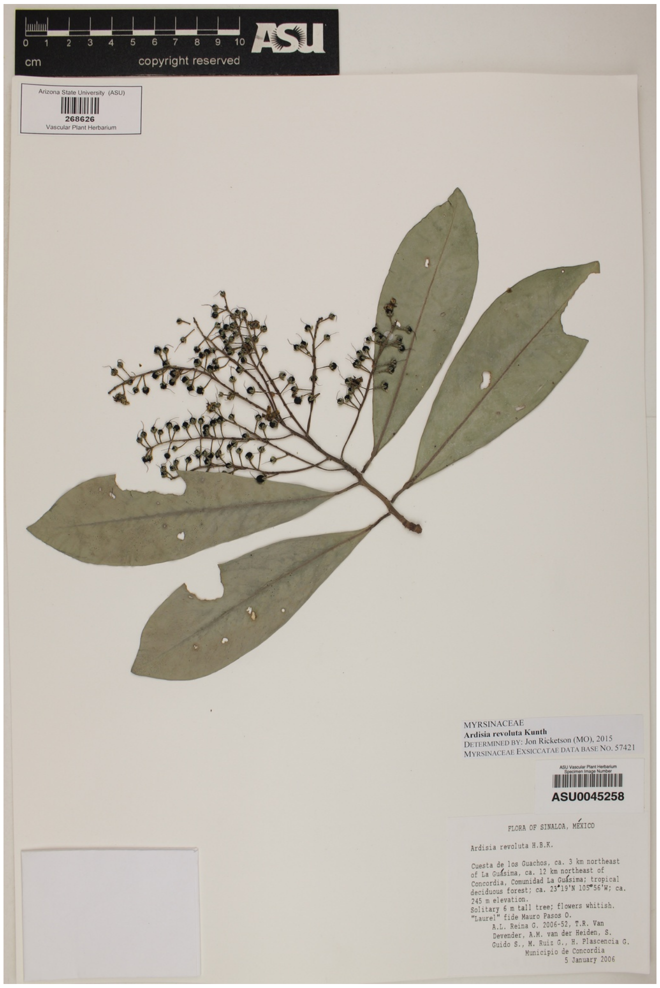
4.3 Herbarium Data

Herbarium data used in the experiments comes from the iDigBio portal, which aggregates and gives access to millions of images for research purposes. As illustrated in Figure 10.1, typical herbarium sheets result in a significantly affected visual representation of the plant, with a typical monotonous aspect of brown and dark green content and a modified shape of the leaves, fruits or flowers due to the drying process and aging. Moreover, the sheets are surrounded by handwritten/typewritten labels, institutional stamps, bar codes and even reference colour bar patterns for the most recent ones. Whereas all of these items are very useful for botanists, they generate a significant level of noise from a machine learning point of view. This research aims at assessing if these images can be handled by deep learning algorithms as suggested in (Mata-Montero et al. 2016). We focus on species classification.

4.4 Datasets

We used five datasets in this research. Two of them use herbarium sheet images from iDigBio; two more use non-dried plant pictures from Costa Rica and France; additionally, ImageNet weights were used to pre-train the deep learning model. We only used the weights of a pre-trained model on ImageNet, not the dataset itself. ImageNet is a well known generalist dataset which is not dedicated to plants, for this reason we didn't not use directly the data of this dataset. Table 10.1 shows the different datasets. The following paragraph explains each dataset and the associated acronyms used throughout this paper:

- CR: the Costa Rica Leaf Scan Dataset (CRLeaves) includes a total of 255 species from the Central Plateau in Costa Rica. It consists of 7,262 images digitized by the National Museum of Costa Rica and



Arizona State University (ASU)
268626
Vascular Plant Herbarium

MYRSINACEAE
Ardisia revoluta Kunth
DETERMINED BY: Jon Ricketson (MO), 2015
MYRSINACEAE EXSICCATAE DATA BASE NO. 57421

ASU Vascular Plant Herbarium
Specimen Image Number
ASU0045258

FLORA OF SINALOA, MÉXICO

Ardisia revoluta H.B.K.

Cuesta de los Guachos, ca. 3 km northeast of La Guásima, ca. 12 km northeast of Concordia, Comunidad La Guásima; tropical deciduous forest; ca. 23°19'N 105°56'W; ca. 245 m elevation.
Solitary 6 m tall tree; flowers whitish.
"Laurel" fide Mauro Pasos O.
A.L. Reina G. 2006-52, T.R. Van Devender, A.M. van der Heiden, S. Guido S., M. Ruiz G., H. Plascencia G.
Municipio de Concordia
5 January 2006

Figure 7.2. *Ardisia revoluta* Kunth herbarium sheet sample taken from Arizona State University Herbarium.

Table 7.2. Datasets used in this research

Name	Acronym	Source	Type	# of Images	# of Species/Classes
CRLeaves	CR	Costa Rica Central Plateau	Leaf Scans	7,262	255
Herbarium255	H255	iDigBio	Herbarium Sheets	11,071	255
PlantCLEF2015	PC	French Mediterranean	In-The-Wild / All organs	113,205	1000
Herbarium1K	H1K	iDigBio	Herbarium Sheets	253,733	1,204
ImageNet	I	ImageNet Challenge	Generic Images	1M	1000

the Costa Rica Institute of Technology (Mata-Montero et al. 2015). Figure 9.2 shows a random sample of this dataset. This is an unbalanced dataset.

- H255: this dataset includes 255 species that match 213 of the species present in the CRLeaves dataset. It uses the iDigBio (*iDigBio* 2017) database and has a total of 11,071 images. Figure 7.4 shows a random sample of pictures from this dataset. This is an unbalanced dataset.

- PC: this is the dataset used in the 2015 PlantCLEF competition. It includes 1,000 species, 91,759 images for training, and 21,446 images for testing (Goëau et al. 2015). Images are from the field and have many organs present. Most images are from the French Mediterranean region. Figure 7.5 shows a random sample of this dataset. This is also an unbalanced dataset.

- H1K: this dataset covers 1,204 species, 918 of which are included in the 1,000 species of the PlantCLEF dataset. Obtained through iDigBio, the dataset contains 202,445 images for training and 51,288 for testing. All images have been resized to a width of 1,024 pixels and their height proportionally, given the huge resolutions used in herbarium images. Figure 7.6 shows a random sample taken from this dataset. This is an unbalanced dataset.

- I: ImageNet is arguably the image dataset most used by the machine learning research community. It contains 1,000 generalist classes and more than a million images (Russakovsky et al. 2015). It is the de facto standard for pre-training deep learning models. We use only the weights of a trained model with this dataset for transfer learning proposes.

4.5 Avoiding Bias

To avoid biases in the experiments, we separated the datasets in a special way for training and testing. For herbarium datasets H255 and H1K, data was separated so that sheets of the same species that were collected by the same collector were not permitted to enter both the training and testing sets. For the CR dataset, we separated the data so that images of different leaves from each specimen are present in either the training or the testing set, but not in both. For the PlantCLEF (PC) dataset, we did this too at the observation level. So, no same observation is present in both training and testing subsets. These measures lead to more realistic and unbiased training/testing scenarios although they also lead to lower accuracy rates.

4.6 Image pre-processing

All datasets were normalized to an uniform size of 256 by 256 pixels without any other type of pre-processing. This is the current state-of-the-art resolution as deep learning models are intensive in computing.

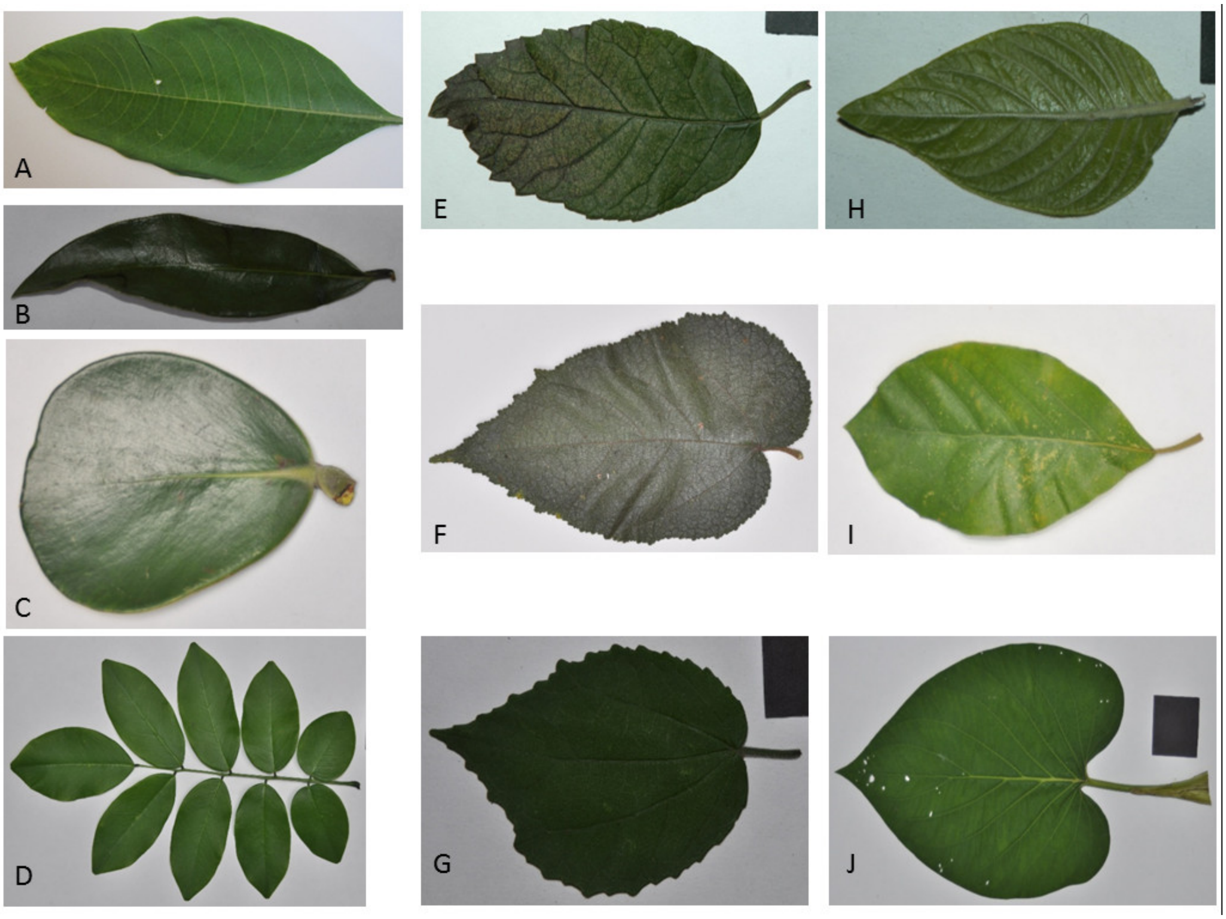


Figure 7.3. 10 leaf-scan images of different species used in the CRLeaves (CR) dataset: A) *Acnistus arborescens* (L.) Schltdl., B) *Brunfelsia nitida* Benth., C) *Clusia rosea* Jacq., D) *Dalbergia retusa* Hemsl., E) *Ehretia latifolia* Loisel. ex A.DC., F) *Guazuma ulmifolia* Lam., G) *Malvaviscus arboreus* Cav., H) *Pentas lanceolata* (Forssk.) Deflers, I) *Persea americana* Mill., J) *Piper auritum* Kunth.

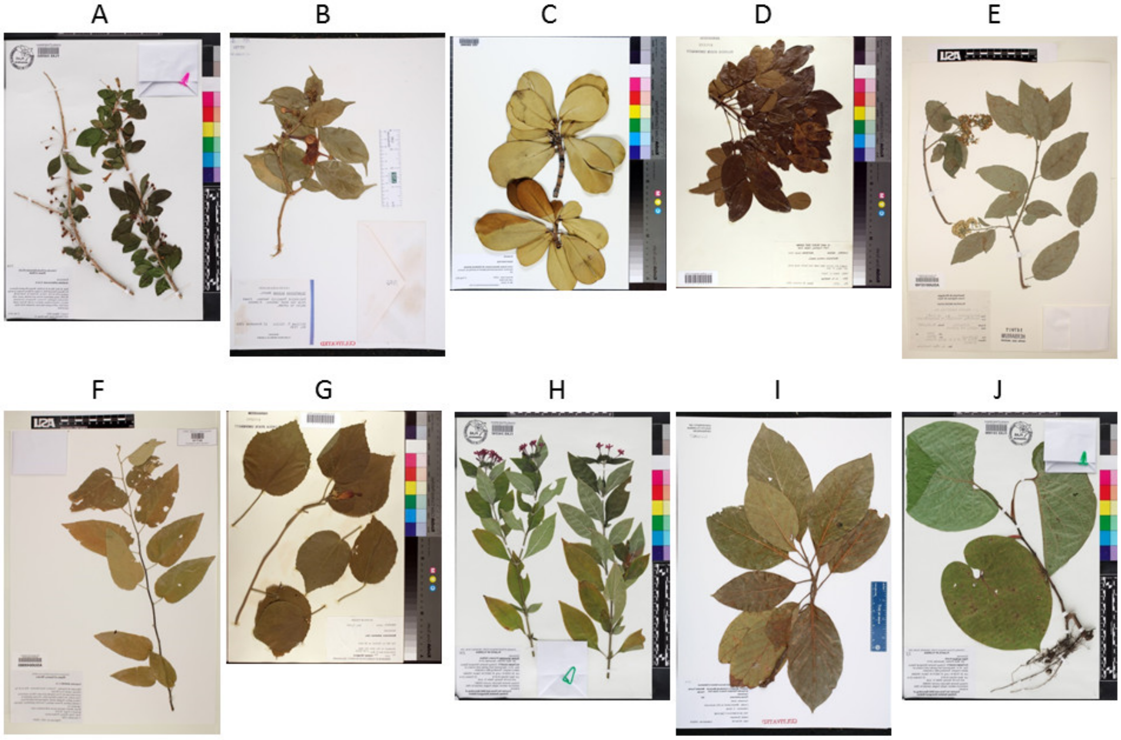


Figure 7.4. 10 herbarium sheet images of different species used in the H255 dataset: A) *Acnistus arborescens* (L.) Schltdl., B) *Brunfelsia nitida* Benth., C) *Clusia rosea* Jacq., D) *Dalbergia retusa* Hemsl., E) *Ehretia latifolia* Loisel. ex A.DC., F) *Guazuma ulmifolia* Lam., G) *Malvaviscus arboreus* Cav., H) *Pentas lanceolata* (Forssk.) Deflers, I) *Persea americana* Mill., J) *Piper auritum* Kunth.

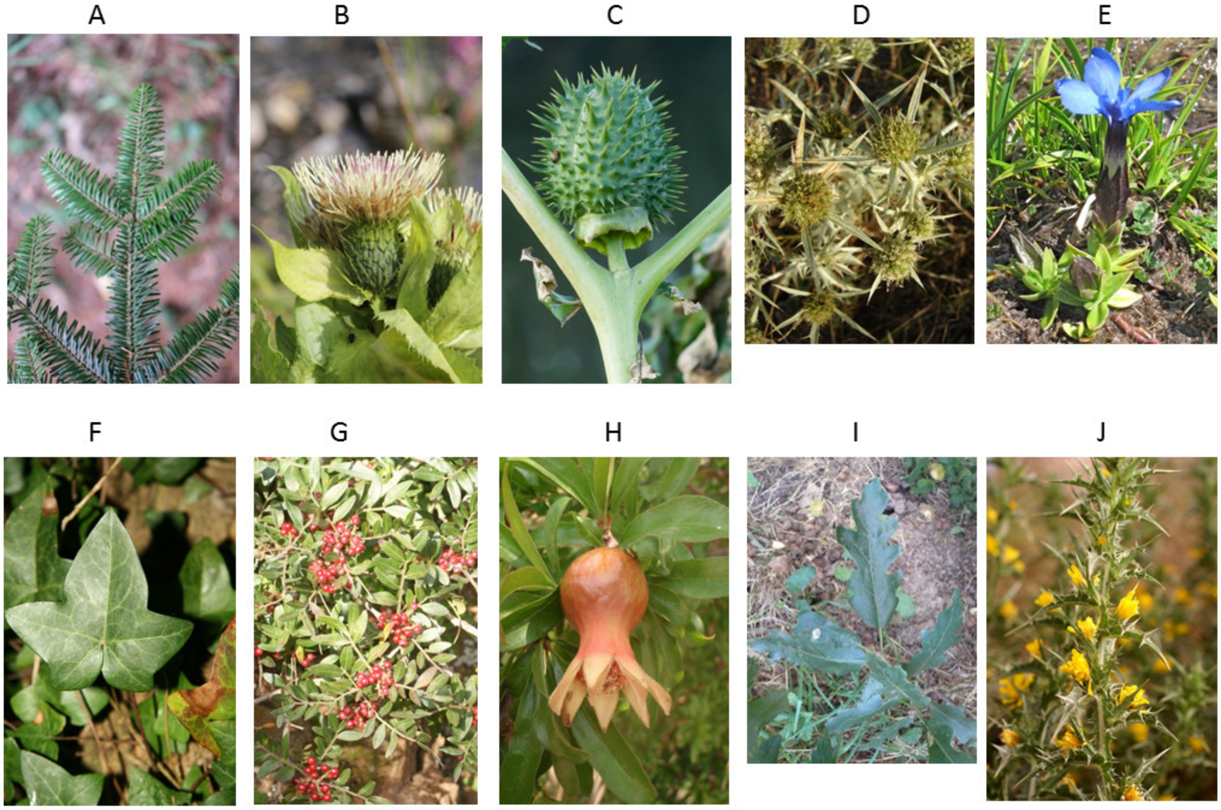


Figure 7.5. Images of different species used in the PlantCLEF (PC) dataset: A) *Abies alba* Mill., B) *Cirsium oleraceum* (L.) Scop., C) *Datura stramonium* L., D) *Eryngium campestre* L., E) *Gentiana verna* L., F) *Hedera helix* L., G) *Pistacia lentiscus* L., H) *Punica granatum* L., I) *Quercus cerris* L., J) *Scolymus hispanicus* L.

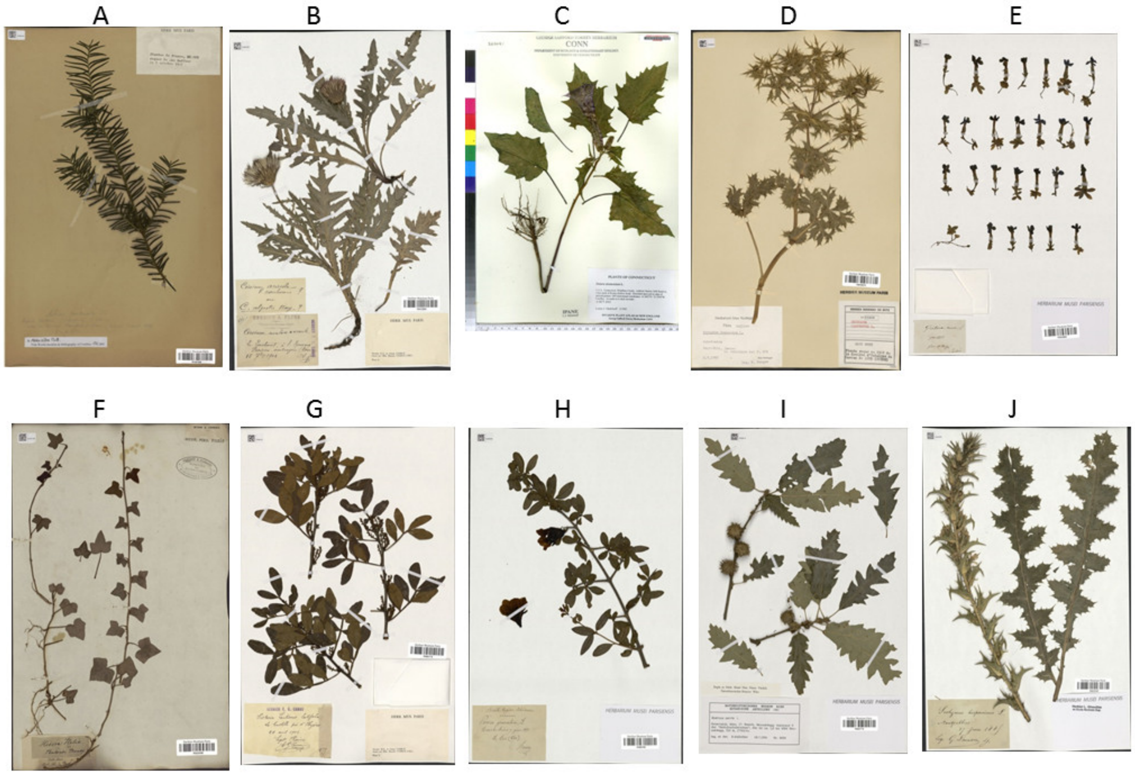


Figure 7.6. 10 herbarium sheet images used in the PlantCLEF (PC) dataset: A) *Abies alba* Mill., B) *Cirsium oleraceum* (L.) Scop., C) *Datura stramonium* L., D) *Eryngium campestre* L., E) *Gentiana verna* L., F) *Hedera helix* L., G) *Pistacia lentiscus* L., H) *Punica granatum* L., I) *Quercus cerris* L., J) *Scolymus hispanicus* L.

5 Experiments and Results

All experiments measured the top-1 and top-5 accuracy of the trained deep learning model under different circumstances, *i.e.*, herbarium specimens classification (section 5.1, Table 7.3), transfer learning across herbarium data from different regions (section 5.2, Table 7.4), and transfer learning from herbarium data to non-dried plant images (section 5.3, Table 7.5).

For each of these experiments, table columns are defined as follows:

- *Experiment*: the name of the experiment. It follows the $\langle Initialization \rangle . \langle Training \rangle . \langle Testing \rangle$ pattern, using the dataset acronyms already discussed. For example, $I.PC.PC$ means the initialization of weights was done by pre-training the network on ImageNet, then fine-tuning it on PlantCLEF training set, and finally testing it with PlantCLEF test set. Similarly, $R.PC.PC$ has almost the same meaning, but the initialization was Random (*i.e.*, no tranfer learning was used). Also, we use index I to mean that at the very beginning the weights of ImageNet were used. For example, $IH1K.PC.PC$ means the transfer learning was progressive, done from ImageNet, to Herbarium1K, to PlantCLEF, and tested with PlantCLEF data.

- *Initialization*: weights used to initialize the model.

- *Training*: training set used (*e.g.*, Herbarium255 training set, PlantCLEF training set, etc.)

- *Testing*: test set used (*e.g.*, Herbarium255 test set, PlantCLEF test set, etc.)

- *Top-1/Top-5*: accuracy achieved with top-1 and top-5 best predictions, respectively.

5.1 Herbarium specimen classification

These experiments aim at assessing the feasibility of using a deep learning system dedicated to herbarium specimen identification at a realistic scale (255 species from Costa-Rica in Herbarium255 and 1K species from France in Herbarium1K). Herbarium255 was divided in 70% training data and the rest 30% as test data used for computing the top-1 and top-5 classification accuracy. Herbarium1K was divided in 80% and 20% respectively, to keep the proportion of the data provided by the PC challenge. The separation was done by species, and within each species, no collector was shared by the training and testing sets to avoid bias in the data. The following four experiments were conducted:

- $R.H255.H255$: The neural network was initialized randomly, trained on the Herbarium255 training set (70%), and tested on the Herbarium255 test set (30%).

- $I.H255.H255$: The neural network was pre-trained on the generalist dataset ImageNet to initialize the weights, fine-tuned on the Herbarium255 training set (70%), and tested on the Herbarium255 test set (30%).

- $R.H1K.H1K$: The neural network was initialized randomly, trained on the Herbarium1K training set (80%), and tested on the Herbarium1K test set (20%).

- $I.H1K.H1K$: The neural network was pre-trained on the generalist dataset ImageNet to initialize the weights, fine-tuned on the Herbarium1K training set (80%), and tested on the Herbarium1K test set (20%).

Table 7.3. Results of the experiments related to herbarium specimens classification.

Experiment	Initialization	Training	Testing	Top-1 Accuracy	Top-5 Accuracy
			Costa-Rica Flora		
$R.H255.H255$	Random	Herbarium255	Herbarium255	0.585	0.771
$I.H255.H255$	ImageNet	Herbarium255	Herbarium255	**0.703**	**0.852**
			France Flora		
$R.H1K.H1K$	Random	Herbarium1K	Herbarium1K	0.726	0.871
$I.H1K.H1K$	ImageNet	Herbarium1K	Herbarium1K	**0.796**	**0.903**

Table 7.3 synthesizes the results of these experiments. A first clear result is that the best accuracies are achieved when ImageNet was used for the initialization step rather than using random weights. This means that herbarium data alone is not sufficient to train the neural network from scratch and that transfer learning from another dataset is significant.

Secondly, when using transfer learning, the achieved accuracies are impressive compared to previous work. We actually obtain similar top-1 accuracies than the recent study of Unger et al. 2016 (73% and 84% on 26 and 17 species, respectively) whereas our classifier is tested (and trained) on one to two orders of magnitude more species. In particular, with a 90% top-5 accuracy for the Herbarium1K dataset, these experiments show that a real-world system to help with herbarium sheet classification is clearly doable.

Thirdly, the slightly better performance on the Herbarium1K dataset compared to to the Herbarium255 dataset is probably related to the fact that the average number of images per species in the training set is much higher (207.13 images per species in Herbarium1K vs. 43.42 images per species in Herbarium255). This would also explain why the gain due to transfer learning is higher for Herbarium255. As the targeted classes (*i.e.* species) are illustrated by less images, the low-level layers of the network benefit more from training on more visual contents beforehand.

5.2 Cross-Herbaria transfer learning

Experiments $H1K.H255.H255$ and $IH1K.H255.H255$, as shown in Table 7.4, compare how prediction works on Herbarium255 (Costa Rica) after transfer learning from Herbarium1K (France). This is important because it provides insights on the possibility of training a deep learning model on a region of the world and use that knowledge in predictions for a different region, particularly for regions where there are not that many herbarium specimen images. In summary, we conducted the following two experiments:

- $H1K.H255.H255$: The neural network was pre-trained on the Herbarium1K dataset to initialize the weights, fine-tuned on the Herbarium255 training set (70%), and tested on the Herbarium255 test set (30%).

- $IH1K.H255.H255$: The neural network was pre-trained on ImageNet and then on Herbarium1K before being fine-tuned on the Herbarium255 training set (70%), and finally tested on the Herbarium255 test set (30%).

As shown in Table 7.4 the results are very promising. By comparing experiment $IH1K.H255.H255$ with experiment $I.H255.H255$ (replicated from Table 7.3), Herbarium255 prediction improves by 4.1% on top-1 accuracy and by 1.9% for top-5 if Herbarium1K is used for transfer learning. It is likely that using the whole iDigBio repository for transfer learning instead of Herbarium1K could give even better results but this is beyond the scope of this paper.

Table 7.4. Results of the experiments related to cross-herbarium transfer learning.

Experiment	Initialization	Training	Testing	Top-1 Accuracy	Top-5 Accuracy
Cross-herbaria Transfer learning (France to Costa-Rica)					
$H1K.H255.H255$	Herbarium1K	Herbarium255	Herbarium255	0.693	0.842
$IH1K.H255.H255$	ImageNet+Herbarium1K	Herbarium255	Herbarium255	**0.745**	**0.872**

If we compare experiment $H1K.H255.H255$ with $I.H255.H255$, the accuracy is almost the same, suggesting that transfer learning from ImageNet only performs similarly to transfer learning from Herbarium1K only. This is good news in the sense that Herbarium1K has much less images than ImageNet, which proves that a dataset smaller than ImageNet but specialized in a specific domain can be as effective in terms of transfer learning.

Finally, by comparing experiment $H1K.H255.H255$ with $R.H255.H255$ (replicated from Table 7.3), we also get an improvement in the accuracy of 10.7% for top-1 and 7% for top-5, suggesting it is way better to use a herbarium dataset from another region for transfer learning instead of just doing random weights initially.

5.3 Transfer learning from herbarium to non-dried plant images

These experiments are meant to measure if using herbarium images for progressive transfer learning is useful on other data types, in particular field images and non-dried leaf scans. Therefore, we conducted the following experiments:

- $R.CR.CR$: The neural network was initialized randomly, trained on the Costa-Rica leaf scans training set (70%) and tested on the Costa-Rica leaf scans test set (30%).
- $I.CR.CR$: The neural network was pre-trained on the generalist dataset ImageNet to initialize the weights, fine-tuned on the Costa-Rica leaf scans training set (70%) and tested on the Costa-Rica leaf scans test set (30%).
- $H255.CR.CR$: The neural network was pre-trained on the Herbarium255 dataset to initialize the weights, fine-tuned on the Costa-Rica leaf scans training set (70%) and tested on the Costa-Rica leaf scans test set (30%).
- $IH255.CR.CR$: The neural network was pre-trained on ImageNet and then on Herbarium255 before being fine-tuned on the Costa-Rica leaf scans training set (70%) and finally tested on the Costa-Rica leaf scans test set (30%).
- $R.PC.PC$: The neural network was initialized randomly, trained on the PlantCLEF training set (80%) and tested on the PlantCLEF test set (20%).
- $I.PC.PC$: The neural network was pre-trained on the generalist dataset ImageNet to initialize the weights, fine-tuned on the PlantCLEF training set (80%) and tested on the PlantCLEF test set (20%).
- $H1K.PC.PC$: The neural network was pre-trained on the Herbarium1K dataset to initialize the weights, fine-tuned on the PlantCLEF training set (80%) and tested on the PlantCLEF test set (20%).
- $IH1K.PC.PC$: The neural network was pre-trained on ImageNet and then on Herbarium1K before being fine-tuned on the PlantCLEF training set (80%) and finally tested on the PlantCLEF test set (20%).

Table 7.5 synthesizes the results of these experiments. The main conclusion is that initializing the models with ImageNet always results in better accuracy for all experiments. If we compare experiments $R.CR.CR$ and $H255.CR.CR$, fine tuning over herbaria against the randomly initialized baseline offers an accuracy increase of 4.7% and 3.8% for top-1 and top-5 respectively. By comparing experiments $R.CR.CR$ and $IH255.CR.CR$, the increase goes up to 12.1% and 8.6% respectively, but still, it is less effective than fine-tuning directly from the ImageNet dataset ($I.CR.CR$). This result is aligned with previous evaluations in the literature (see *e.g.* Goëau et al. 2015; Joly et al. 2016b). It confirms that models trained on a big generalist dataset such as ImageNet can be used as generic feature extractors for any domain-specific task. On the contrary, the visual features learned on Herbarium255 are more specific to herbarium content and do generalize less well to the leaf scans classification task (even if Herbarium255 and CRLeaves cover the same species). This is coherent with the conclusions of Tomaszewski et al. 2016 that leaf shape changes during the drying process and that shape analyses should be performed on datasets containing only dried or fresh leaves.

The results obtained on the PlantCLEF dataset suggest that it is even less possible to transfer knowledge from herbarium to field images (in particular, wild flower images, which is the most represented type of view in the PlantCLEF dataset). By comparing the results of experiment $R.PC.PC$ and $H1K.PC.PC$, we can actually notice that the accuracy decreases by 6.1% for the top-1 and 6.8% for the top-5. If we compare $I.PC.PC$ with $IH1K.PC.PC$, the decrease reaches 9.8%. This means that the visual features learned from the herbarium data are even worse than random features for the initialization of the network. To better understand the reason for this phenomenon, we plotted in Figure 7.7 the evolution of the loss function of the network during training (for experiments $R.PC.PC$, $I.PC.PC$ and $H1K.PC.PC$). It shows that using the $H1K$-based initialization causes the network to converge quickly to a stable but worse solution than when using the random or the ImageNet-based initialization. Our interpretation is that the stochastic gradient descent is blocked into a saddle point close to a local minimum. This is probably due to the fact that the visual features learned on the herbarium data are somehow effective in classifying the field images, but far away from the optimal visual features that should be learned. The visual aspect of a herbarium image is indeed very different from a picture of a plant in natural conditions. Several phenomena affect the transformation of the plant sample during the drying process. There is first a strong variation of the colors of the plant, indeed most of the dry leaves have a brown instead of a green color when they are fresh, flower and fruit colors are also strongly impacted. Furthermore, herbarium specimens have often an overlap of their leaves with flowers and fruits that makes difficult the automated identification of the object of interest inn the herbarium image. 3D objects such as fruits and flowers are also completely transformed when they are pressed. These transformations are most probably the reasons why transfer learning from herbarium images to field data isn't effective.

6 Discussion and Conclusions

This study is, to our knowledge, the first one that analyzes a big dataset with thousands of specimen images from herbaria and uses deep learning. Results show the potential of deep learning on herbarium species identification, particularly by training and testing across different herbarium datasets. This could potentially lead to the creation of a semi, or even fully, automatic system to help taxonomists and experts do their annotation, classification, and revision work at herbarium.

In particular, we showed that is possible to use a herbarium image dataset from one region of the world to do transfer learning to another region, even when the species do not match. This indicates that a deep learning approach could be used in regions that do not have lots of herbarium images. On the negative side, we did show that it is not beneficial to do transfer learning from herbarium data to leaf scan pictures and it is even counterproductive to do transfer learning from herbarium data to field images. This confirms some previous studies in the literature that concluded that the observable morphological attributes can change

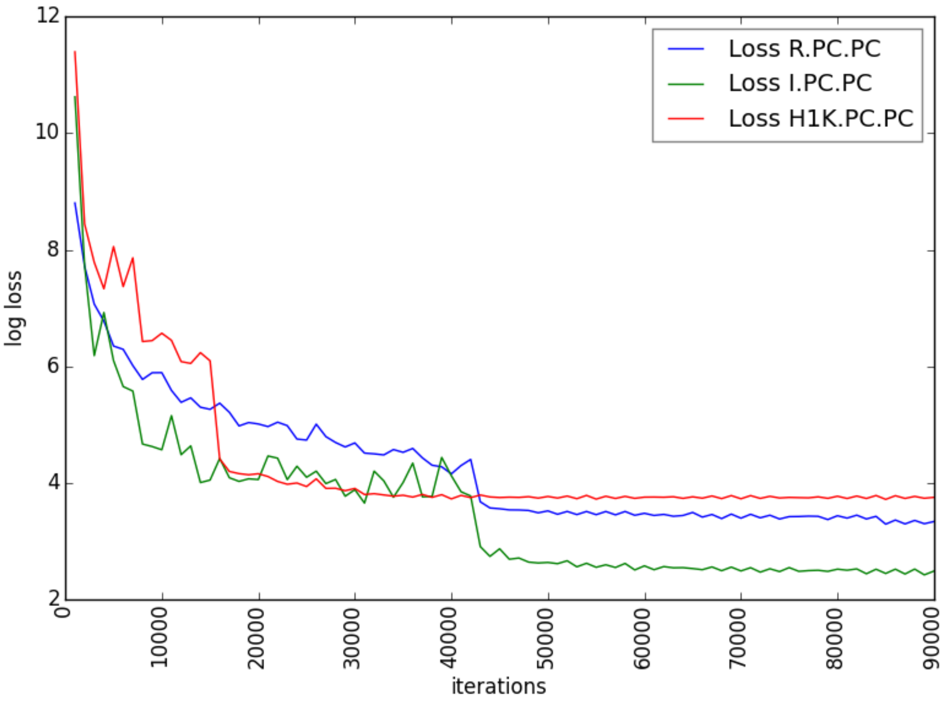


Figure 7.7. Comparison of losses of $R.PC.PC$, $I.PC.PC$ and $H1K.PC.PC$ experiments

Table 7.5. Results of the experiments related to transfer learning from Herbarium to non-dried plant images

Experiment	Initialization	Training	Testing	Top-1 Accuracy	Top-5 Accuracy
		CRLeaves Baselines			
$R.CR.CR$	Random	CRLeaves	CRLeaves	0.37	0.50
$I.CR.CR$	ImageNet	CRLeaves	CRLeaves	**0.51**	**0.61**
		CRLeaves using transfer learning from herbarium data			
$H255.CR.CR$	Herbarium255	CRLeaves	CRLeaves	0.416	0.542
$IH255.CR.CR$	ImageNet,Herbarium255	CRLeaves	CRLeaves	0.491	0.590
		PlantCLEF Baselines			
$R.PC.PC$	Random	PlantCLEF	PlantCLEF	0.334	0.566
$I.PC.PC$	ImageNet	PlantCLEF	PlantCLEF	**0.523**	**0.726**
		PlantCLEF using transfer learning from herbarium data			
$H1K.PC.PC$	Herbarium1K	PlantCLEF	PlantCLEF	0.273	0.498
$IH1K.PC.PC$	ImageNet,Herbarium1K	PlantCLEF	PlantCLEF	0.425	0.661

significantly with the drying process. Additionally, the particular unnatural layout of plants and their parts on herbarium sheets may also have a negative effect.

It is worth trying to apply some pre-processing on the herbarium datasets for further experimentation, particularly to get rid of handwritten tags and other visual noise present in the herbarium sheets. Additionally, as per results on only herbarium data, it would be a good idea to start working on a model whose hyperparameters, architecture and data augmentation are thought for herbarium in particular, to maximize accuracy for a system dedicated to herbarium in mind. More experiments with bigger leaf datasets are recommended, since some viability of using herbarium for fine tuning on leaf images was observed. Concerning the question of how herbarium data could be useful for field images classification, we believe we should rather try to model the drying process itself typically by learning a transfer function between a representation space dedicated to herbarium images and another one dedicated to field images. In order to improve the accuracy in future experiments, an option is to explore the taxonomy as a class hierarchy. Several others possibilities could potentially improve transfer learning between herbarium images and images of plants in the field. Herbarium annotation (with tags on what is possible to see in the image of the specimen) could be a first important step of progress for the computer vision community. Indeed, if we are able for the same species to use images of herbarium and plant in the field that contain the same visual information (both in flower, or with leaves for example), we will be able to better understand contexts in which transfer learning failed or potentially be improved. Herbarium visual quality evaluation could be also of a great interest. Indeed, some herbarium specimens can be really precious for the botanical community, but if the plant sample in the image is too old and damaged, this specimen will be of poor interest for automated species identification. The individual image quality evaluation could be very useful to weight the use of each images during the learning phase on training datasets.

Finally, based on our results, we believe that the development of deep learning technology based on herbarium data, together with the recent recognition of e-publication in the International Code of Nomenclature (Nicolson et al. 2017) will also contribute to significantly increase the volume of descriptions of new species in the following years.

7 Declarations

7.1 Ethics approval and consent to participate

This study included no data or analysis requiring special consent to conduct or to publish.

7.2 Consent to publish

Not applicable.

7.3 Availability of data and materials

The datasets used in this study will be permanently available here `http://otmedia.lirmm.fr/LifeCLEF/GoingDeeperHerbarium/`

7.4 Competing interests

The authors declare that they have no competing interests.

7.5 Funding

This work was partially supported by the Costa Rica Institute of Technology.

7.6 Authors' Contributions

JMC wrote the manuscript, with contributions from AJ, HG, EMM and PB. JMC did the experiments and developed the computational methods, supervised by HG, AJ, EM and PB. All authors read and approved the final manuscript.

7.7 Acknowledgements

Thanks to the National Museum of Costa Rica for their help with the collection, identification, and digitization of samples in the Costa Rican leaf-scan dataset. Special thanks to the Costa Rica Institute of Technology for partially sponsoring this research. We would also like to thank the large community that has actively engaged in iDigBio initiative for the valuable access to the herbarium data.

Chapter 8

Automated Herbarium Specimen Identification using Deep Learning

Reference Jose Carranza-Rojas, Alexis Joly, Pierre Bonnet, Hervé Goëau, and Erick Mata-Montero (2017a). "Automated Herbarium Specimen Identification using Deep Learning". In: *Biodiversity Information Science and Standards* 1, e20302. DOI: 10.3897/tdwgproceedings.1.20302. eprint: https://doi.org/10.3897/tdwgproceedings.1.20302

Keywords Biodiversity Informatics; Computer Vision; Deep Learning; Plant Identification; Herbaria

1 Abstract

Hundreds of herbarium collections have accumulated a valuable heritage and knowledge of plants over several centuries (Page et al. 2015). Recent initiatives, such as iDigBio [1] aggregate data from and images of vouchered herbarium sheets (and other biocollections) and make this information available to botanists and the general public worldwide through web portals. These ambitious plans to transform and preserve these historical biodiversity data into digital format are supported by the United States National Science Foundation (NSF) Advancing the Digitization of Natural History Collections (ADBC) and the digitization is done by the Thematic Collections Networks (TCN) funded under the ADBC program. However, thousands of herbarium sheets are still unidentified at the species level while numerous sheets should be reviewed and updated following more recent taxonomic knowledge. These annotations and revisions require an unrealistic amount of work for botanists to carry out in a reasonable time (Bebber et al. 2010). Computer vision and machine learning approaches applied to herbarium sheets are promising (Wijesingha et al. 2012) but are still not well studied compared to automated species identification from leaf scans or pictures of plants taken in the field.

In a recent study, we evaluate the accuracy with which herbarium images can be potentially exploited for species identification with deep learning technology (Carranza-Rojas et al. 2017b), particularly Convolutional Neural Networks (CNNs) (Szegedy et al. 2015). This type of network allows automatic learning of the most prominent visual patterns in the images since they are trainable end-to-end (thus, differentiable), as opposed to previous approaches that use custom, hand-made feature extractors. A first challenge is to use herbarium sheet images alone to automatically identify the species of plants mounted on herbarium sheets. Secondly, we

[1] https://www.idigbio.org

propose studying if the combination of herbarium sheet images with photos of plants in the field (Carranza-Rojas et al. 2016a; Joly et al. 2015a) is a viable idea to train models that provide accurate results during identification. Finally, we explore if herbarium images from one region with a specific flora can be used in transfer learning (a technique in deep learning that first allows training a model with a dataset and then once trained, uses the weighted results to train another model with that knowledge as the baseline) to another region with other species; for example, in a region under-represented in terms of collected data.

Our evaluation shows that the accuracy for species identification with deep learning technology, based on herbarium images, reaches 90.3% on a dataset of more than 1200 European plant species. This could potentially lead to the creation of a semi-, or even fully automated system to help taxonomists and experts with their annotation, classification, and revision works.

In this paper, we take a closer look at the accuracy levels achieved with respect to the first two challenges. We evaluate the accuracy levels for each species included in the dataset, which encompasses 253,733 images, 1,204 species.

2 Research Questions

As shown in (Carranza-Rojas et al. 2017b), the accuracy levels achieved are high for plant species identification using herbarium collection images. In this work we answer 2 additional questions:

- What are the accuracies reported per species/genus/family?
- Is there a relation between high accuracy per species and the number of images used for training?

The first question provides insights in the automatic plant identification at other taxonomic levels beyond species. This is useful for species which do not have high accuracy and perhaps having only the genus of family is enough. The second research question is important to understand if certain species have visual features that are very prominent, allowing easier identification even in the absence of lots of images. Additionally, it may allow new research focused on such species and what the deep learning models are learning from them. By comparing the internal of the models with the human taxonomic knowledge in some way, it can provide insights of how the models are actually classifying the species and if they are learning the same as human taxonomists.

3 Experiments & Results

We run the same experiments as the one in (Carranza-Rojas et al. 2017b), but we measure the accuracy not only at the species level, but also by genus and family levels. Additionally, we measure the accuracy per each species and compare it with the number of images available for each species. Figure 8.1 shows the results of the training on the $H1K$ dataset for Top-1 and Top-5 accuracy plus the loss of the model, for species.

Figure 8.2 shows the results of Top-1 accuracy per species class, same as Figure 8.3 for Top-5 accuracy. It can be noticed how there are some species that are very well identified, but also a big number of species is never identified at all. Same situation happens in Figure 8.3 for Top-5 accuracy, but of course, accuracy goes up overall. Table 8.1 shows the result for the 10 top identified species. There are even more that are perfectly identified given the dataset.

In Figure 8.4 we show the results of identification at the genus level, with a total of 501 genera. Similarly to species, there is a group of genera that are not identified at all, but also there are some that are identified

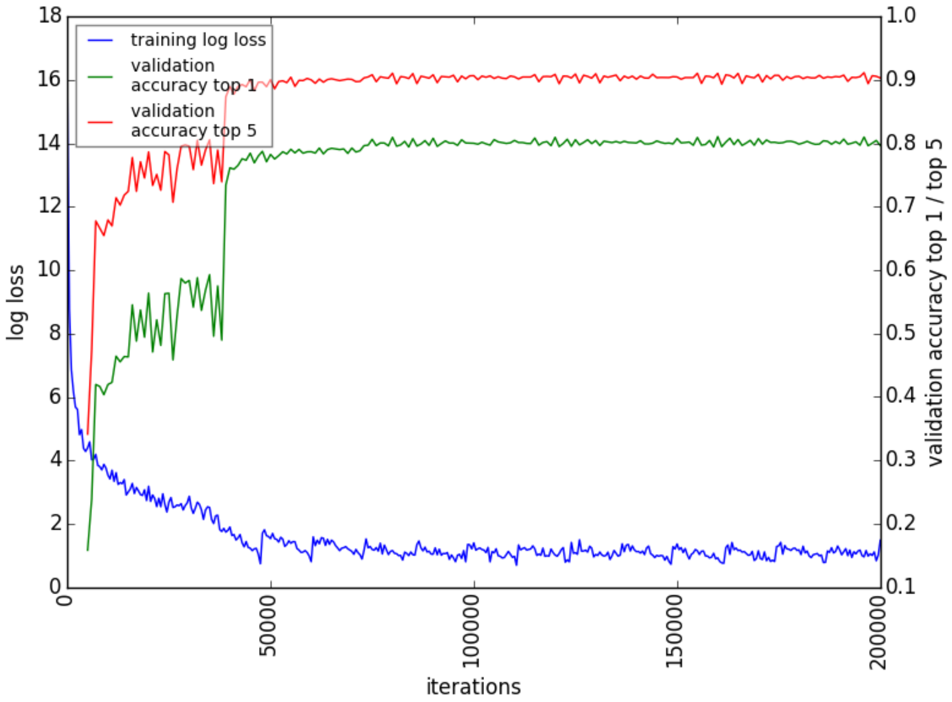


Figure 8.1. Loss, Top-1 and Top-5 accuracy for Experiment $H1K$

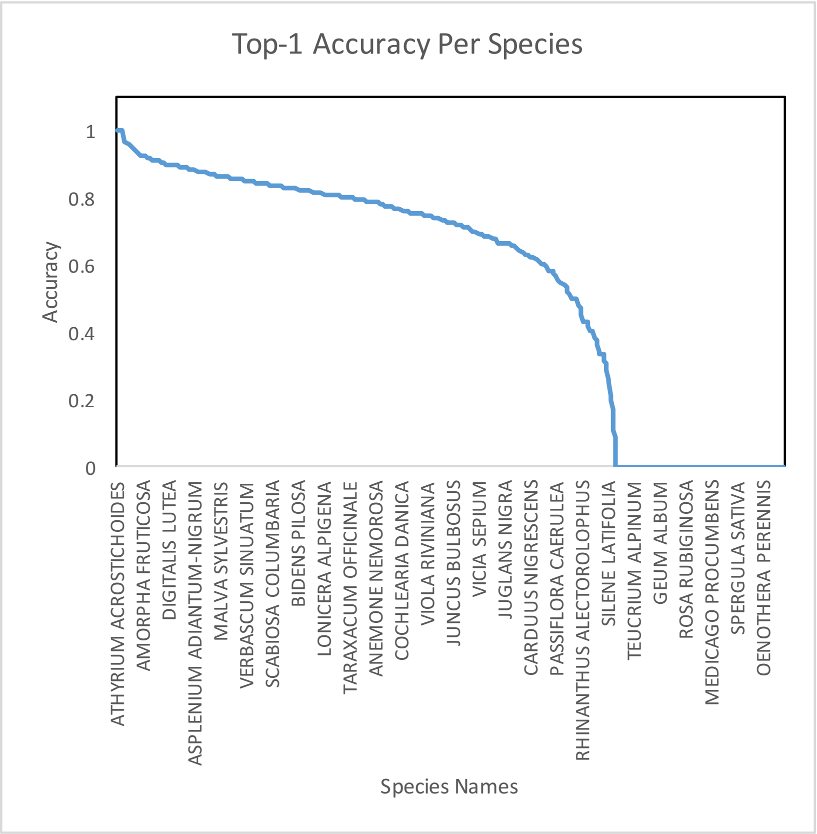


Figure 8.2. Top-1 Accuracy per species

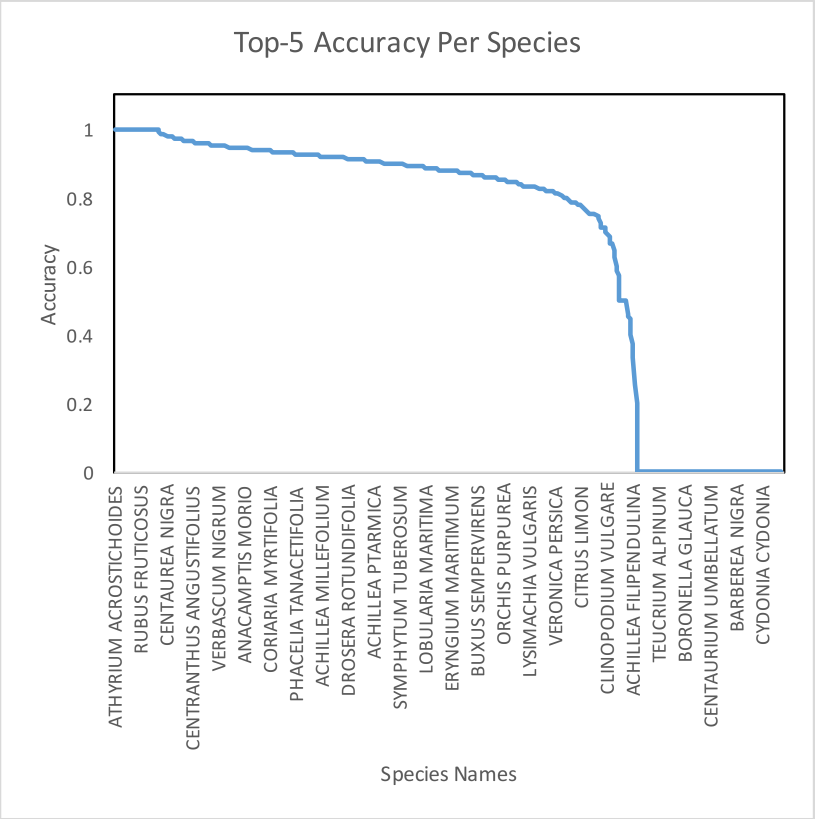


Figure 8.3. Top-5 Accuracy per species

Table 8.1. 10 of the best identified species from 1225

Species	Adjusted Top-1
PRUNUS AMERICANA	1
NONOTROPA HYPOPITYS	1
SAXIFRAGA PANICULATA	1
BLECHNUM SPICANT	1
TRIFOLIUM CAMPESTRE	1
OENOTHERA GLAZIOVIANA	1
RHINANTHUS CRISTA-GALLI	1
HOMOGYNE ALPINA	1
RIBES SATIVUM	1
MALVA ROTUNDIFOLIA	1

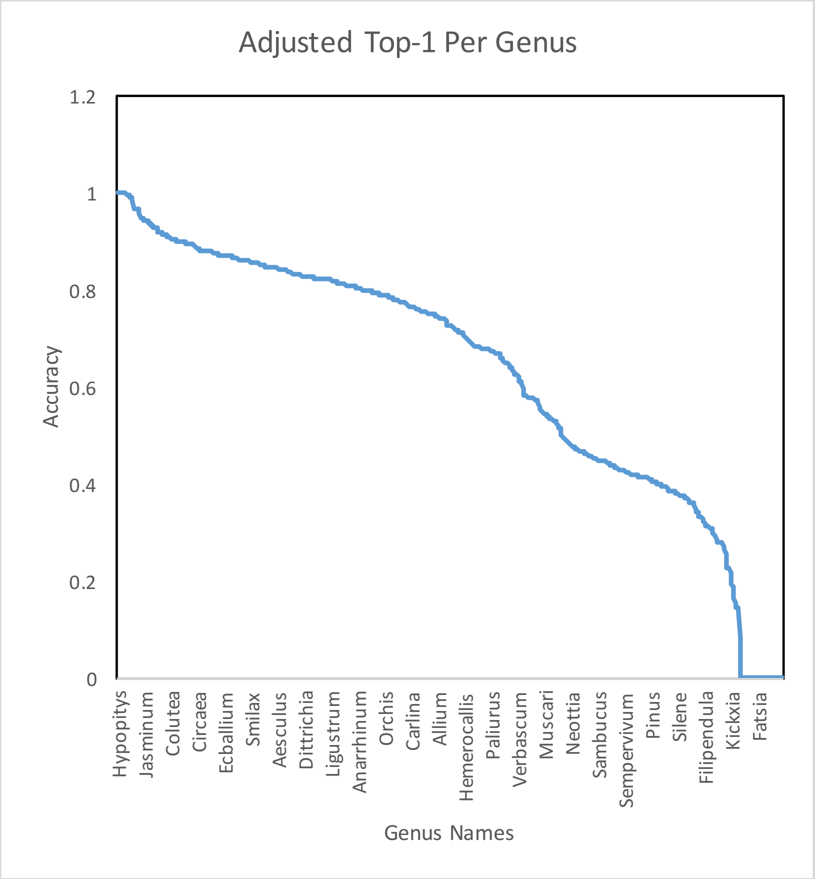


Figure 8.4. Adjusted Top-1 per genus, by averaging the adjusted accuracy per species

perfectly. The amount of unidentified genera is less than in the species experiment. Similarly, Table 8.2 shows the 10 top identified genera. It can be noticed that less genera are identified with perfect accuracy, at least compared to species.

Similar situation happens with the families. Figure 8.5 shows the behavior of accuracy per family with a total of 124 families. In this case, as the number of classes is fewer than genera and species, the number of unidentified families is less. Table 8.3 shows the best identified families.

Figure 8.6 reports the distribution of species accuracy compared with the amount if images per species. It can be noticed how some species get high accuracy regardless of the few amount of images available for them. This is important since that means those species may have certain highly distinguishable visual patterns that may match human taxonomists' knowledge. It can also be noticed how there is a conglomerate of species that possess higher accuracy that 80%, and a number of images above 100. This means a species is likely to be well identified if the number of images for training is around 100 or more.

4 Conclusions

This research shows that some species can be particularly well identified regardless of the number of images available. This is important to discover new insights in why the model is so robust for such species, even in the absence of lots of images which is a normal need in the deep learning domain.

Table 8.2. 10 of the best identified genera from 501

Species	Adjusted Top-1
Hypopitys	1
Blechnum	1
Amaroria	1
Homogyne	1
Nigella	1
Casuarina	1
Leucanthemopsis	1
Bryonia	0.9969
Kerria	0.9952
Gladiolus	0.9922
Bellevalia	0.9889

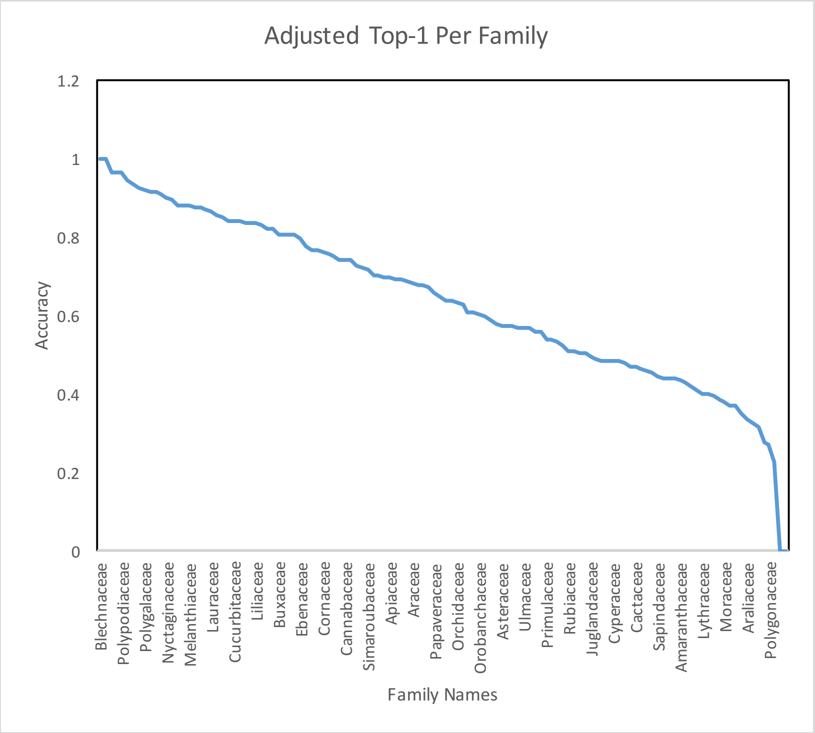


Figure 8.5. Adjusted Top-1 per family, by averaging the adjusted accuracy per species

Table 8.3. 10 of the best identified families from 124

Species	Adjusted Top-1
Blechnaceae	1
Casuarinaceae	1
Osmundaceae	0.9655
Equisetaceae	0.9645
Polypodiaceae	0.9640
Arecaceae	0.9453
Ginkgoaceae	0.9350
Garryaceae	0.9259
Polygalaceae	0.9171
Lentibulariaceae	0.9142
Meliaceae	0.9135

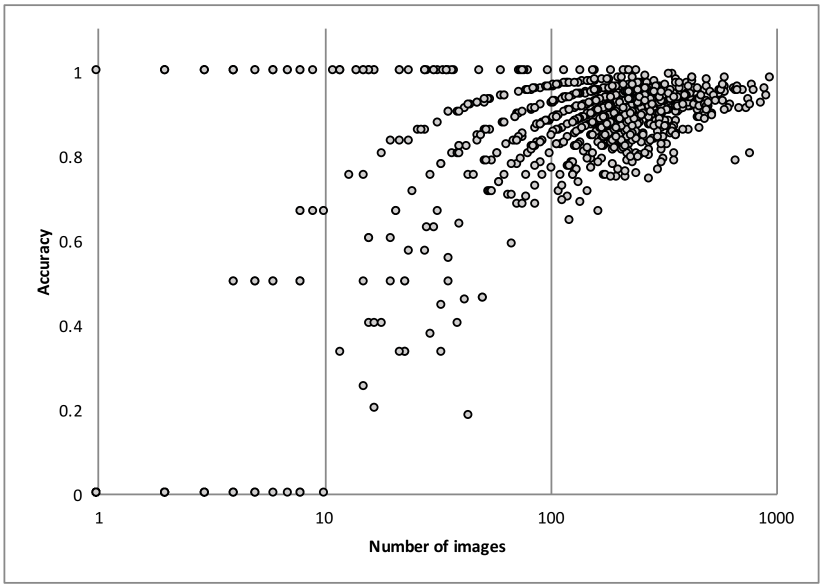


Figure 8.6. Species distribution per number of images and accuracy

5 Future Work

Now that we know that some species are identified very highly regardless of the amount of images, we can explore the patterns learned by the deep learning model by using DN at the latest layers of the model. Then, human taxonomists can point if those visual patterns of pixels are meaningful to them or not. In case they are, the models are learning similar patterns just like human experts. If they are not know, perhaps human taxonomists can learn new ways to classify species of plants. This could even lead to the creation of new taxonomic keys.

6 Acknowledgements

Thanks to the National Museum of Costa Rica for their help with the collection, identification, and digitization of samples in the Costa Rican leaf-scan dataset. Special thanks to the Costa Rica Institute of Technology for partially sponsoring this research. We would also like to thank the large community that has actively engaged in iDigBio initiatives, for the valuable access to their herbarium data.

7 Hosting institution

- Costa Rica Institute of Technology, Costa Rica
- Institut National de Recherche en Informatique et en Automatique (INRIA), France
- Centre de Cooperation Internationale en Recherche Agronomique pour le Developpement (CIRAD), France

Chapter 9

Hidden Biases in Automated Image-Based Plant Identification

Reference Jose Carranza-rojas, Erick Mata-Montero, and Herve Goeau (2018). "Hidden Biases in Automated Image-Based Plant Identification". In: *2018 IEEE International Work Conference on Bioinspired Intelligence (IWOBI)*, pp. 1–9. DOI: 10.1109/IWOBI.2018.8464187

Keywords Biodiversity Informatics, Computer Vision, Image Processing, Leaf Recognition, Automated Plant Identification, Deep Learning

1 Abstract

Plant identification is critical to support important biodiversity conservation actions such as biodiversity inventories, monitoring of populations of endangered organisms, and assessing climate change impact, among many others. Because deep learning has demonstrated impressive results in the field of computer vision in general, research on automatic plant identification has been shifting its attention towards deep learning approaches. However, some authors have noticed that an important methodological issue may have been overlooked in the design of many experiments, which may explain why, on one hand, some studies based on hand-crafted feature extraction approaches report very high accuracy levels, but, on the other hand, newer deep learning approaches used in events such as the PlantCLEF challenge report relatively lower accuracy levels. Because PlantCLEF uses same specimen photos exclusively in either the training dataset or the testing dataset, we postulate that this may explain the lower accuracies achieved. Specifically, we explore the following two questions: does using different images of the same specimen for training and testing introduce a significant bias in deep learning experiments as well as in those that use handcrafted features in classical computer vision techniques? Does it affect the accuracy of species identifications even in the more restricted domain of leaf-based automated species identifications? We also address the issue of scalability of accuracy results for both, a particular feature extraction approach and a deep learning approach. All experiments are conducted on a dataset of 7,262 photos of leaves of 255 species of plants from Costa Rica.

2 Introduction

Plant identification is key to support very important biodiversity conservation actions such as biodiversity inventories, monitoring of populations of endangered organisms, measuring climate change impact, and modelling of invasive species geographical distributions, among many others. However, the traditional approach used by taxonomists to identify species based on their morphology is tedious and very sensitive to errors (Carvalho et al. 2007). As a result, automatic plant identification has generated considerable of attention in recent years (Hebert et al. 2003; Joly et al. 2016a; Kumar et al. 2012; MacLeod 2007; Mata-Montero et al. 2016). Computer vision and machine learning techniques stand out as particularly effective to fully or partially automate the process of identifying plants and, in general, organisms. Some fully functional apps such as Leafsnap (Kumar et al. 2012) and, more recently, Pl@ntNET (Joly et al. 2016a) have already been developed and made available. Both apps put in the hands of millions of users around the globe the ability to identify species of plants by just taking pictures with their mobile phone.

Because deep learning has demonstrated enormous potential and very good results vis-a-vis computer vision in general, research on automatic plant identification has been shifting its attention towards deep learning approaches. However, it is interesting to note that numerous studies using approaches based on hand-crafted feature extraction report very high accuracy (Aggarwal et al. 2012; Arun et al. 2013; Bhardwaj et al. 2013; Herdiyeni et al. 2012; M. Z. Rashad 2011; Mata-Montero et al. 2015; Wijesingha et al. 2012; Wu et al. 2007). In contrast, newer deep learning approaches used in events such as the PlantCLEF 2015 and PlantCLEF 2016 challenges (Goëau et al. 2015; Goëau et al. 2016) did not report such high accuracy.

Throughout this paper, we use the term *Same-Specimen-Picture Bias (SSPB)* to refer to a particular characteristic of the datasets used for the training phase and the testing phase of a supervised learning experiment with a global dataset D. We say that SSPB is avoided (absent) if all plant images from dataset D are distributed so that, for each specimen S all its images are used exclusively in the training phase or in the testing phase. Otherwise, we say SSPB is (potentially) present.

It is critical to understand at this point the difference between a specimen and a species. Species are the basic units of taxonomic classification, ranking below a genus and denoted by a Latin binomial, e.g., *Homo sapiens*. A specimen is a particular instance of the species category. Therefore, because useful datasets always contain, for each species, one or more pictures of each of several specimens, avoiding SSPB in an experiment does not mean that all images of a species are used either for training or for testing. In fact, for each species, we assume that a percentage of all the images of that species are used for training and the rest for testing, but, at the specimen level, all images of a given specimen are either used for training or for testing. An *observation* is a set of pictures of the same specimen. Figure 9.1 illustrates two ways of splitting into a training and a test subsets a very small dataset with three observations, which are differentiated by the color of their frames. Each observation, in turn, consists of three pictures. On the left, the SSPB is avoided while on the right, pictures from a same specimen, sharing potentially very similar color and texture distributions, are indifferently in the training and the test datasets.

Our interest on the subject arises from the fact that PlantCLEF experiments avoid SSPB and this may explain the lower accuracies achieved as compared to other experiments where SSPB is not ignored. In this study, we specifically explore the following two questions: does the presence of SSPB really introduce a significant bias in deep learning experiments as well as in those that use hand-crafted feature extraction techniques? Does it affect the accuracy of species identifications even in the more restricted domain of leaf-based automated species identifications? Additionally, as mentioned (yet not studied deeply) in (Mata-Montero et al. 2016) we address the issue of scalability of accuracy results for both a particular feature extraction approach and a deep learning approach.

This paper is organized as follows: Section 3 presents related work. Section 4 covers methodological aspects and Section 5 experimental design. Section 6 discusses the results obtained in this research. Con-

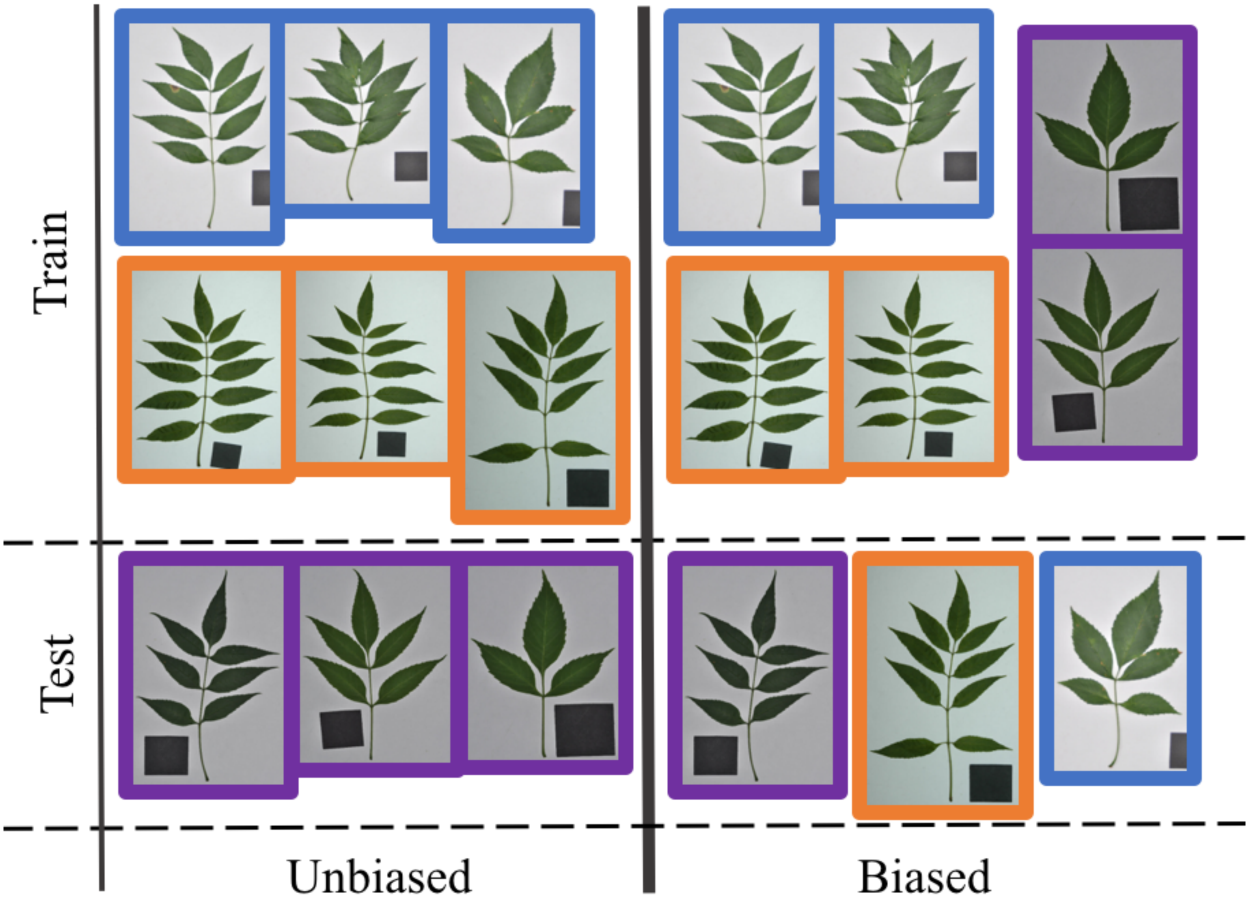


Figure 9.1. Unbiased and biased splits of a small dataset of 3 specimens of *Tecoma stans (L.) Juss. ex Kunth*. Each specimen contains 3 photos framed with the same color.

clusions are presented in Section 1. Finally, future work is summarized in Section 8.

3 Related Work

There has been extensive research based on traditional computer vision approaches to plant species identification, in particular, using leaf scan datasets, which consist of leaf images on a uniform background.

Kumar et al. 2012 created LeafSnap, a mobile app that uses curvature features to classify species of plants based on leaf images. They use kNN and similarity measures to compare the curvature histograms of each leaf. Their dataset consists on 184 species, and the reported top-5 accuracy is 96.8%.

Herdiyeni et al. 2013 use texture to classify medicinal and house plant species from Indonesia. They use LBP with different sample points and radius to calculate a histogram and then concatenate all histograms of different radious together. They run their tests on two datasets: the first one consists of 1,440 images with uniform background across 30 species of tropical plants. The second has 300 non-uniform images of 30 different species of house plants. The reported precision is 77% for medicinal species and 86.67% for house plants. SURF was used by Nguyen et al. 2013 on grascale images of leaves. A precision of 95.94% is reported using the Flavia dataset (Wu et al. 2007). This dataset comprises 3,621 leaf images and 32 species.

In (Mata-Montero et al. 2015) LBPU and kNN are used to classify species with a dataset of 66 species from Costa Rica. The best reported top-5 accuracy is of 85.4%. A very good survey on plant identification is (Wäldchen et al. 2017), which contains most of the reported accuracies for these traditional approaches. With the Flavia dataset (Wu et al. 2007) the reported top-1 accuracies go up to 97.80%.

Because of the excellent results obtained in computer vision challenges such as ImageNet (Russakovsky et al. 2015), the research community has been focusing on deep learning approaches in the last few years. During the 2016 PlantCLEF plant identification challenge, the organizers released a paper (Goëau et al. 2016) that summarizes the challenge results. The training set used in the challenge had 113,205 pictures obtained from 41,794 observations of 1,000 species, collected from 8,960 contributors. Images correspond to leaves (in complex, natural backgrounds), flowers, stems, the entire plant, fruits, and leaf scans. The test set was built from images taken from Pl@ntNet (Joly et al. 2014a). It comprises 8,000 images, 4,633 labeled as the known 1,000 species, 3,367 as new unknown classes. Teams from eight countries participated in the challenge and submitted results. The top 26 runs with best performance were all based on CNNs. The best average precision achieved was of 71.8% by the Japanese team (Hang et al. 2016) which used a VGGNet (Simonyan et al. 2014). An important approach by Hang et al. 2016 uses convolutional layers for species and organs separately and then they merge both into a single set of features.

The work in (Lee et al. 2015) is one of the first attempts of using deep learning for plant identification, particularly with CNNs. They also use DN to describe visually how the patterns are build, starting with generic blobs until some vein patterns emerge. Venation of different orders are chosen by the model for pattern recognition at different layers, which reflects how texture is key for plant identification (Mata-Montero et al. 2016). The model used is AlexNet (Krizhevsky et al. 2012), and transfer learning is applied from ImageNet (Russakovsky et al. 2015). The used dataset contains leaf images from England, with a total of 44 species. The achieved top-1 accuracy is 99.5%. Grinblat et al. 2016 use a custom CNN formed by two convolution layers and RELU non-linearities between them. In addition, they develop models that contain up to five layers, all with a softmax layer at the end. The reported mean accuracy is 98.8%. However the dataset used is very small. It comprises 866 pictures and has only three species of legumes, namely, white beans, red beans, and soy beans.

With the exception of PlantCLEF, none of the previously described research indicates in their publications that SSPB is considered in their experiments. SSPB is not an obvious issue as images from the same specimen are not obviously visually similar since they differ in sizes, angles, and distances, among others.

Even for the trained human eye, they may look as different specimens. Instead, previous research generally mentions how images from the global dataset D are distributed across the training and testing dataset; for example, 70% randomly selected images of each species go into the training dataset and the remaining 30% goes into the testing dataset. Multiple images from the same specimen (very common in plant datasets) are not reported as an issue when the training and testing datasets are defined. A "reasonable" assumption that we ourselves adopt in previous experiments (Mata-Montero et al. 2016) is that diversity of leaf images of a single specimen may be so large that SSPB is not really a significant bias. The experiments described in the following sections demonstrate that the opposite is true.

4 Methodology

This section describes the two set ups for the training and testing datasets. Then, as an orthogonal issue, it presents the two computer vision pipelines/approaches used to classify the leaf-scan images into species. As a result, we have the four scenarios depicted in Table 9.1

Table 9.1. Scenarios studied in this research.

	Data sets	
Experiments	Unbiased	Biased
Hand-crafted feature extraction	1	2
Deep learning	3	4

4.1 Datasets

We used a leaf-scan dataset D that contains a total of 7,262 photos of 255 tree species from the Central Plateau in Costa Rica. Leaf samples were collected during the first semester of 2016 by taxonomists from the Costa Rica National Museum[1] and digitized by a technician from the Costa Rica Institute of Technology. All 255 species were used for training. Images were re-sampled to a 224×224 resolution for the deep learning experiment and to a 700×525 resolution for the hand-crafted feature extraction experiment. Figure 9.2 shows a random sample taken from the dataset. It is very important to mention that, for this work, it is crucial to include as metadata no only the species associated with each photo but also a unique specimen identifier.

Biased Dataset

This dataset was created by randomly taking 70% of the data in D for training and 30% for testing. These are really approximate percentages because such distribution has to be attempted for each species in dataset D. How close to a 70%-30% distribution is achieved depends on how many photos per species there are in dataset D. For example, if a given species X has 20 photos, 14 randomly chosen photos will go to the training dataset and 6 to the testing dataset. However, if the number of photos for species X is 2, one will go to the training dataset and the other one to the testing dataset, resulting in a 50%-50% distribution. This is the approach most of previous plant identification studies have followed (particularly those based on leaf-scans).

[1] http://www.museocostarica.go.cr

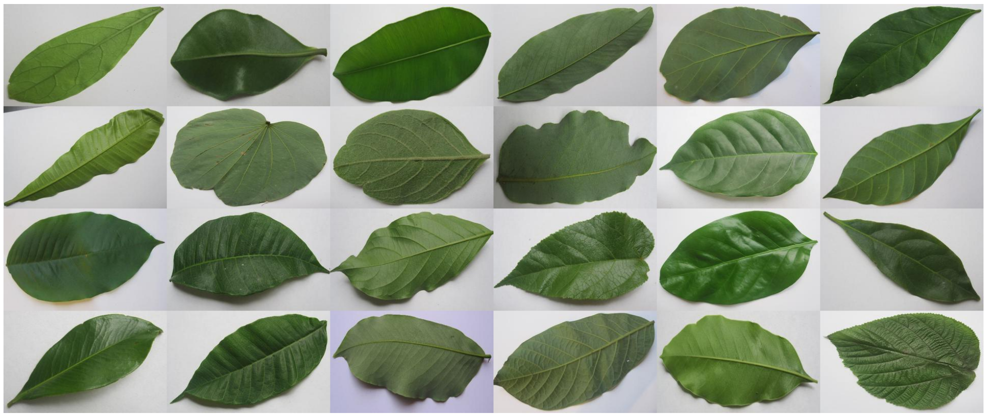


Figure 9.2. Random sample of the Costa Rican leaf-scan dataset.

Unbiased Dataset

For each species X in dataset D, we tried to approximate a 70%-30% distribution as in the Biased Dataset case. However, in order to avoid the SSPB, we used the unique specimen identifier to place images from the same specimen in eiher the training dataset or the testing dataset. This restriction alone could not be achievable if, for example, one species X has photos of several leaves but they are all from the same specimen. Our dataset D was created so that it precludes such scenario, as there are photos from at least two different specimens for each species X in dataset D. However, this restriction clearly makes it harder to approximate the ideal 70%-30% distribution both, at the species level and globally.

4.2 Computer Vision Approaches/Pipelines

We use two pipelines to classify leaf-scan images into species. In one, we use a traditional hand-crafted feature extraction approach, specifically, LBPU to extract texture features from the images, as done in (Carranza-Rojas et al. 2016a; Mata-Montero et al. 2015). We also use a deep learning approach as a second pipeline. Both are discussed in the following subsections.

Hand-crafted Feature Extraction

Based on the work described in (Mata-Montero et al. 2015), segmentation was done by using EM to get the binary image of the leaf and to ignore the background. HSV was used as the color domain to help on the segmentation, as explained in (Kumar et al. 2012). Then, LBPU features were extracted from the dataset and histograms based on the LBPU representations of pixels of each image were created. We used a radius of 3 pixels, and the sampling of circumference pixels was done with 16 pixels, forming binary numbers of 16 digits. Figure 9.3 illustrates the traditional pipeline applied to a *Bauhinia purpurea* leaf sample image. Finally, by using a histogram intersection function, similarity was calculated by applying a kNN classification algorithm to extract the best k candidates. In previous work (Mata-Montero et al. 2015), this approach gave very good results on a 66 species dataset of Costa Rican plant species. However, in those experiments SSPB was present. Because that dataset is fully included in our dataset D and we are using the same hand-crafted feature extraction algorithm, our first experiment, as described in the following section, allows us to assess the scalability of the accuracy results when going from dataset with 66 species to an extended dataset with

255 species.

Deep Learning

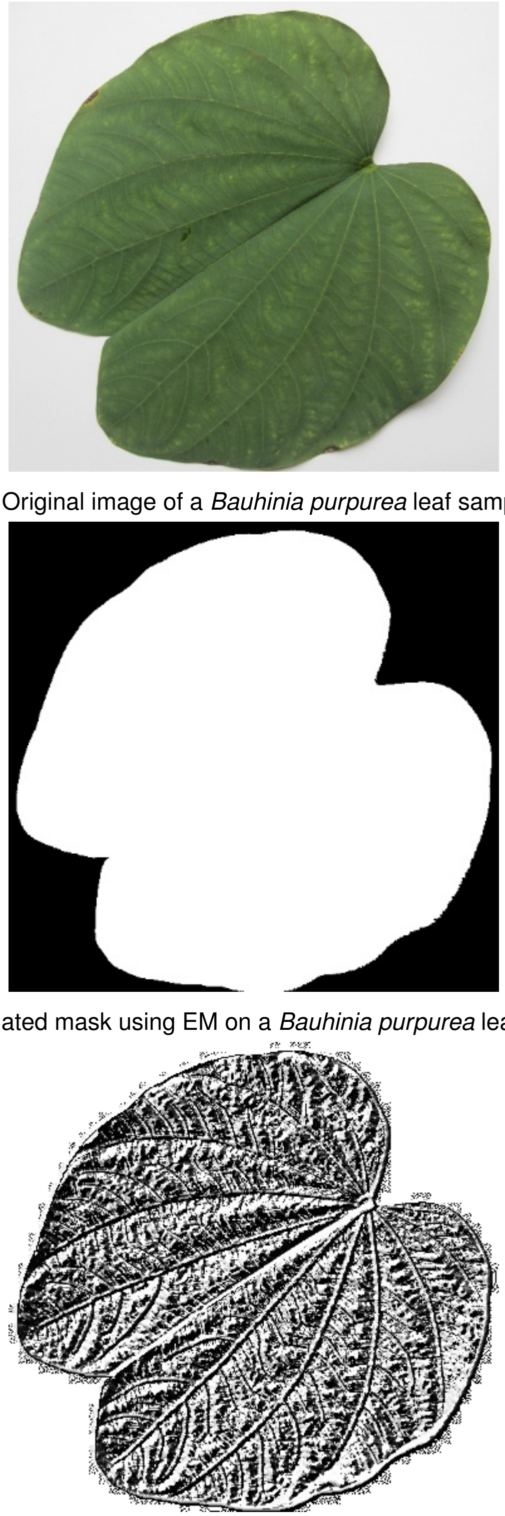
We chose CNNs (LeCun et al. 1995) because of the impressive results obtained in classification challenges such as ImageNet (Krizhevsky et al. 2012). The main strength of this approach comes from its ability to learn discriminant visual features directly from the raw pixels of the images without falling into the trap of the *curse of dimensionality* (Goodfellow et al. 2016). This is achieved by stacking multiple *convolutional layers*, which are the core building blocks of a CNN. A convolutional layer basically takes images as input and produces as output *feature maps* that correspond to different convolution kernels, while looking for different visual patterns.

Additionally, CNNs also achieved increasingly impressive results during the 2015, 2016, and 2017 editions of the PlantCLEF challenge (Goëau et al. 2015; Goëau et al. 2016), which demonstrates that they can capture important features from plant images directly. They represent the state of the art in plant identification. We used the GoogleNet model (Szegedy et al. 2015), which is based on the idea of the so-called inception layers, providing faster training with less parameters as it does not depend heavily on fully-connected layers. The model used is extended with batch normalization (Ioffe et al. 2015) for faster convergence and with PRELU activation function (He et al. 2015).

The GoogleNet used is shown in Table 10.2. The model comprises several inception modules, uses batch normalization after all pooling layers, and is implemented by using Caffe (Jia et al. 2014). A batch size of 24 images is used for each iteration, with a learning rate of 0.0075 with images of 224×224 resolution. Simple crop and resize data augmentation is used with the default settings of Caffe.

Table 9.2. GoogleNet architecture (Szegedy et al. 2015) modified with Batch Normalization.

Type	Patch size / Stride	Output Size	Depth	Params	Ops
convolution	7x7/2	112x112x64	1	2.7K	34M
max pool	3x3/2	56x56x64	0		
batch norm		56x56x64	0		
LRN		56x56x64	0		
convolution	3x3/1	56x56x192	2	112K	360M
max pool	3x3/2	28x28x192	0		
batch norm		28x28x192	0		
LRN		28x28x192	0		
inception (3a)		28x28x256	2	159K	128M
inception (3b)		28x28x480	2	380K	304M
max pool	3x3/2	14x14x480	0		
batch norm		14x14x480	0		
inception (4a)		14x14x512	2	364K	73M
inception (4b)		14x14x512	2	437K	88M
inception (4c)		14x14x512	2	463K	100M
inception (4d)		14x14x528	2	580K	119M
inception (4e)		14x14x832	2	840K	170M
max pool	3x3/2	7x7x832	0		
batch norm		7x7x832	0		
inception (5a)		7x7x832	2	1072K	54M
inception (5b)		7x7x1024	2	1388K	71M
avg pool	7x7/1	1x1x1024	0		
batch norm		1x1x1024	0		
linear		1x1x10000	1	1000K	1M
softmax		1x1x10000	0		



(a) Original image of a *Bauhinia purpurea* leaf sample.

(b) Calculated mask using EM on a *Bauhinia purpurea* leaf sample.

(c) LBPU calculated with 16 pixels of circumference and 3 pixels of radius for a *Bauhinia purpurea* leaf sample image after applying a mask.

Figure 9.3. Traditional pipeline applied to a *Bauhinia purpurea* leaf sample image.

To do transfer learning we initialized the model parameters from an ImageNet previous training. This was used in all our experiments with the deep learning approach in order to maximize the accuracy, based on the fact that initial layers of the model learn very general patterns that can be used in other domains (Yosinski et al. 2014).

5 Experiments

5.1 Hand-crafted Feature Extraction Experiment

The first experiment consists of running a traditional computer vision pipeline using a LBPU feature extractor and kNN as explained in Section 4. In terms of the four scenarios depicted in Table 9.1, this experiment covers scenarios 1 and 2, that is, the first row of the table. This experiment applies the whole traditional pipeline to the data; so, there is a segmentation phase in both, the biased and unbiased runs. As indicated in the previous section, because this experiment includes scenario 2, we aim at comparing not only the impact of SSPB when a hand-crafted feature extraction approach is used, but also, the scalability of accuracy by replicating the experiments in (Mata-Montero et al. 2015) with a larger dataset.

5.2 Deep Learning Experiment

The second experiment is similar to the first one but it uses deep learning CNN to determine if the SSPB introduces a significant bias. It corresponds to the second row in Figure 9.1. There is no segmentation applied to the data, only a resize to 224x224 pixels before running the network on the data.

6 Results

6.1 Hand-crafted Feature Extraction Experiment

Table 9.3 shows the results related to the hand-crafted feature extraction approach experiment. The accuracy obtained is always considerably better when the biased dataset is used, with a 26.4% difference when top-5 is used. This clearly shows that SSPB introduces a significant bias.

Table 9.3. Unbiased and Biased Top-5 Accuracy with LBPU.

k	Unbiased Accuracy	Biased Accuracy
1	0.0669	0.2095
2	0.0963	0.2805
3	0.1200	0.3357
4	0.1383	0.3800
5	0.1513	0.4162

It is also important to notice how this approach does not scale well as the number of species becomes higher. By comparing the results in (Mata-Montero et al. 2015), where only 66 species were used, we see an abrupt decline in accuracy. The best accuracy obtained here is 41.6% which is considerably lower than 90%, the corresponding accuracy reported in (Mata-Montero et al. 2015).

6.2 Deep Learning Experiment

Figure 9.4 summarizes the results of this experiment. Analogously to the first experiment, results are considerably better in scenario 4 as compared to scenario 3. Thus, SSPB also introduces a significant bias when this approach is used. Top-5 accuracy is almost a 100% with the biased dataset, while it drops to 51% with the unbiased dataset. Similarly, with top-1 accuracy the difference is around 50%. Once again, the results show that SSPB has a significant impact on the accuracy of leaf-based automated species identification using CNNs.

It should also be noticed that the deep learning approach obtains considerably better results than the traditional approach, both for unbiased and biased datasets. For the unbiased case, it improves from 0.1513 to 0.51 for top-5 accuracy, and from to 0.0669 to 0.38 for top-1 accuracy. For the biased case, it improves from 0.4162 to 0.99 for top-5 accuracy, and from 0.2095 to 0.95 for top-1 accuracy. This matches the behavior during the last three PlantCLEF competitions, where the best results were obtained using deep learning, regardless of the increasing growth of the challenge dataset (Goëau et al. 2015; Goëau et al. 2016).

7 Conclusions

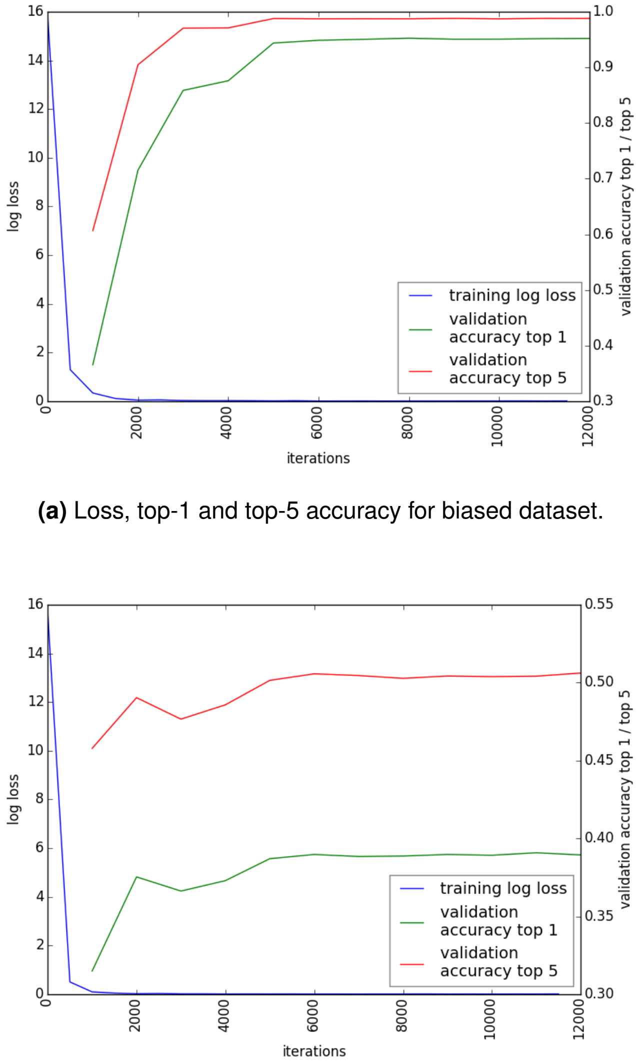
We demonstrated that automated image-based plant identification can be very sensitive to an often overlooked bias with respect to pictures of the same specimens. We studied two specific cases, one uses deep learning and the other uses a hand-crafted feature extraction approach. In both cases, SSPB introduces a very significant bias. Given the fact that users of a production system will most likely take pictures of specimens that were not used for training, it is realistic to assume that the training and testing phases should resemble that scenario, i.e., the testing phase should not assess accuracy by using pictures of specimens that were also used in the training phase, even if the pictures are different.

With the exception of PlantCLEF publications, most reports on experiments about automated image-based identification systems do not explicitly state that accuracy has been measured while avoiding the SSPB. However, if not taken into account, this bias can dramatically cause an over-fit of the data. Therefore, future automated image-based plant identification experiments and reports should explicitly address what measures were taken to vis-a-vis this bias.

It is also important to notice that the traditional approach used with hand-crafted feature extractors does not scale up as well as deep learning approaches. This was concluded by comparing the accuracy obtained with each approach (scenarios 2 and 4 of the experiments) when compared with the results in (Mata-Montero et al. 2015). For scenario 2 top-5 accuracy decreased from 90% to 41.6

8 Future Work

We demonstrated that SSPB introduces a considerable bias in four scenarios that use plant leaf image datasets. Preliminary results suggest it also happens with datasets of images that include the whole plant or other components. This should be further investigated. The tested bias (SSPB) may not be the only one present in plant datasets. In other scenarios such as plant identification using herbarium sheets (Carranza-Rojas et al. 2017b), there might be hidden biases with regards to other variables such as the author of the sheets. Thus, additional experiments are needed with such datasets in order to understand if additional hidden biases are also present.



(a) Loss, top-1 and top-5 accuracy for biased dataset.

(b) Loss, top-1 and top-5 accuracy for unbiased dataset.

Figure 9.4. Experiment using Deep Learning on both biased and unbiased datasets.

Acknowledgement

To Alexander Rodriguez and Armando Estrada from the National Museum of Costa Rica (MNCR) and to the MNCR for their support in the process of collecting and identifying leaf samples in the field. To Nelson Zamora and INBio for their contribution also in the process of collecting and identifying leaf samples in the field. Special thanks to the Costa Rica Institute of Technology for providing the funding that made this research possible.

Chapter 10

Automated identification of herbarium specimens at different taxonomic levels

Reference Jose Carranza-Rojas, Alexis Joly, Hervé Goëau, Erick Mata-Montero, and Pierre Bonnet (2018). "Automated Identification of Herbarium Specimens at Different Taxonomic Levels". In: *Multimedia Tools and Applications for Environmental & Biodiversity Informatics*. Ed. by Alexis Joly, Stefanos Vrochidis, Kostas Karatzas, Ari Karppinen, and Pierre Bonnet. Cham: Springer International Publishing, pp. 151–167. ISBN: 978-3-319-76445-0. DOI: 10.1007/978-3-319-76445-0_9. URL: https://doi.org/10.1007/978-3-319-76445-0_9

Keywords Biodiversity Informatics, Computer Vision, Image Processing, Automated Plant Identification, Hierarchical Deep Learning, Taxonomy, Hierarchies, Hierarchical Classification

1 Abstract

The estimated number of flowering plant species on Earth is around 400,000. In order to classify all known species via automated image-based approaches, current datasets of plant images will have to become considerably larger. To achieve this, some authors have explored the possibility of using herbarium sheet images. As the plant datasets grow and start reaching the tens of thousands of classes, unbalanced datasets become a hard problem. This causes models to be inaccurate for certain species due to intra- and inter-specific similarities. Additionally, automatic plant identification is intrinsically hierarchical. In order to tackle this problem of unbalanced datasets, we need ways to classify and calculate the loss of the model by taking into account the taxonomy, for example, by grouping species at higher taxon levels. In this research we compare several architectures for automatic plant identification, taking into account the plant taxonomy to classify not only at the species level, but also at higher levels, such as genus and family.

2 Introduction

In general, Deep Learning classification has focused mostly on flat classification, i.e., hierarchies and knowledge associated with higher levels are normally not taken into account. However, in the biological domain, the approach traditionally followed by taxonomists is intrinsically hierarchical. Single-access and multiple-access identification keys are an example of such an approach (Mata-Montero et al. 2016). They are used to identify organisms mostly at the species level but sometimes at the genus and family levels too. To our knowledge, most of the research on image-based automated plant identifications classify plant images into species and do not exploit knowledge about other taxonomic levels.

Very few studies also have attempted to use herbarium images for plant identification. With new deep learning methods, large datasets of herbarium images such as those published by iDigBio [1], which comprises millions of images of thousands of species from around the globe, become very useful. These datasets are suitable for deep learning approaches and include as metadata all levels of the associated taxonomy. In (Carranza-Rojas et al. 2017b) a GoogleNet model with modifications is used to classify species from the Mediterranean region and Costa Rica. It shows promising results in terms of accuracy when training and testing with herbarium sheet images, as well as when doing transfer learning from the Mediterranean region to Costa Rica. However, classifications are conducted only at the species level and do not use additional knowledge vis a vis other taxonomic levels.

Herbaria normally hold many samples that have not been identified at the species level (Bebber et al. 2010) but they make an effort to at least have them identified at the genus or family level. It is therefore important to help streamline the identification process with tools that support identifications at multiple levels (probably with different levels of accuracy).

One of the biggest issues in plant identification is the lack of balanced datasets. At the species level, most available datasets are unbalanced due to taxonomically uneven nature of sample collection processes (Mata-Montero et al. 2016). So, an expected intuition in this domain is to exploit higher levels of the taxonomy in order to have more images of a single class and use that knowledge to help the classification at lower levels of the taxonomy, such as the species at the bottom. In other words, the unbalanced dataset issue could be tackled by using a class hierarchy and doing classifications from top to bottom.

In this work we compare several deep learning architectures to do herbarium specimen identification at not only species level, but also other taxonomic levels such as genus and family. We explore architectures that do several taxonomic level classifications at the same time by sharing parameters, as well as separated flat classifiers, independent from each other.

The rest of this manuscript is organized as follows: Section 3 presents relevant related work. Section 4 and Section 5 cover methodological aspects and experiment design, respectively. Section 6 describes the results obtained. Section 1 presents the conclusions and, finally, Section 8 summarizes future work.

3 Related Work

PlantCLEF is the largest and best known plant identification challenge (Joly et al. 2016c). It has helped to create bigger datasets each year as well as allowed participants to gradually improve the techniques (mostly deep learning based models) to achieve better accuracy. So far, PlantCLEF has focused on species level identifications only.

The same situation happens with apps for automated image-based plant identification such as LeafSnap (Kumar et al. 2012) and Pl@ntNet (Joly et al. 2016a). These apps are also focused on classification only at

[1] https://www.idigbio.org/

the species level; however, it would be useful in cases where the accuracy is low, to have predictions at other taxonomic levels such as genus and family.

Very few studies have tackled the problem of hierarchical classification. Silla et al. 2011 present a very comprehensive survey about different techniques used for hierarchical classification and also layout a unifying framework to classify existing approaches. Wu et al. 2005 discuss how there are no even proper standards to evaluate hierarchical classification systems, and use Naive Bayes approach on text data. Both studies are focused on traditional machine learning, not deep learning.

Shahbaba et al. 2007 create a new method using a Bayesian form of the softmax function, adding a prior that introduces correlations between the parameters of nearby classes of the hierarchy. This approach was also developed for traditional machine learning and not deep learning but could be easily adjustable to deep learning. This approach is useful also when there is a prior knowledge of the class hierarchy.

Yan et al. 2015 create a new architecture named HD-CNN, which uses 2 levels of classification. The first level is more general and then the second level is composed by several smaller classifiers per each class in the first classifier. This means the amount of classifiers grows after the first classification. Also, an error during the first classification will lead to error propagation to the second layer of classifiers.

There have been a lot of studies where the hierarchy is learned via unsupervised learning. In this paper, we focus on an already defined hierarchy which is a plant taxonomy. It is the result of decades if not centuries of work in the field of taxonomy, so we don't calculate automatically the class hierarchy.

In particular, to our knowledge, no plant identification system or study has been proposed that actually exploits the class hierarchy using the plant taxonomy. In (Carranza-Rojas et al. 2017a) authors do analyze the accuracy per species but also per genera and families, to see which species are better identified and also to evaluate if the amount of images per species has a direct impact over the accuracy obtained per class. They conclude after around 100 images, the classes are very well identified with some exceptions, but also some species are very well identified regardless of having a very small number of images. They also provide accuracy per genus and family.

4 Methodology

Previous work in (Carranza-Rojas et al. 2017b) has tackled the problem of using a big dataset with herbarium images for automatic plant identification. We describe the herbarium dataset taken from this study used for this research. We have also added information about genera and families, beyond species, in order to test the hierarchical architectures.

4.1 Datasets

Herbarium data used in the experiments comes from the iDigBio portal, which aggregates and gives access to millions of images for research purposes. As illustrated in Figure 10.1, typical herbarium sheets result in a significantly affected visual representation of the plant, with a typical monotonous aspect of brown and dark green content and a modified shape of the leaves, fruits or flowers due to the drying process and aging. Moreover, the sheets are surrounded by handwritten/typewritten labels, bar codes, institutional stamps and even reference colour bar patterns for the most recent ones.

Additionally, ImageNet weights were used to pre-train the deep learning model. We only used the weights of a pre-trained model on ImageNet, not the dataset itself. The following are the details of the datasets:

- H1K: this dataset covers 1,191 species, 918 of which are included in the 1,000 species of the PlantCLEF dataset from 2015 (Goëau et al. 2015). Obtained through iDigBio, the dataset contains 202,445

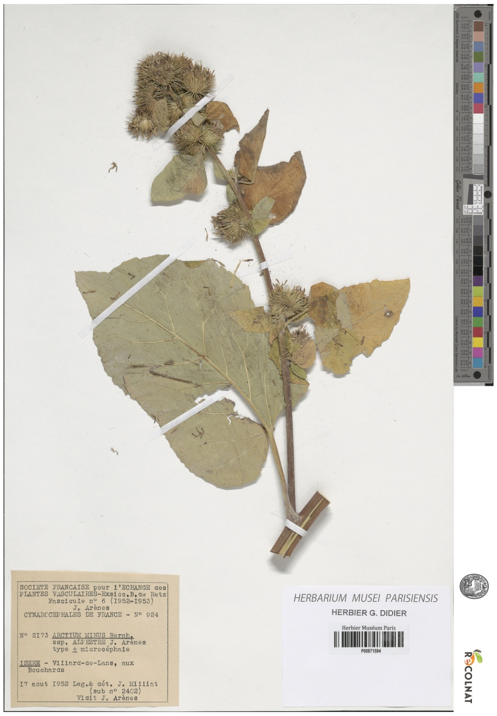

SOCIETE FRANCAISE pour l'ECHANGE des
PLANTES VASCULAIRES-Exsicc.B.de Retz
Fascicule n° 6 (1952-1953)
J. Arènes
CYNAROCEPHALES DE FRANCE - N° 924
N° 2173 ARCTIUM MINUS Bernh.
ssp. ALPESTRE J. Arènes
type ± microcéphale
ISERE - Villard-de-Lans, aux
Boucharas
17 aout 1952 Leg.& dét. J. Milliat
(sub n° 2402)
Vidit J. Arènes
HERBARIUM MUSEI PARISIENSIS
HERBIER G. DIDIER
Herbier Muséum Paris
P00871594
RECOLNAT

Figure 10.1. *arctium minus (hill) bernh.* herbarium sheet sample taken from Herbier Muséum Paris.

Table 10.1. Datasets used in this research

Name	Acronym	Source	Type	# of Images	# of Species/Classes
Herbarium1K	H1K	iDigBio	Herbarium Sheets	253,733	1,191
ImageNet	I	ImageNet Challenge	Generic Images	1M	1000

images for training and 51,288 for testing. All images have been resized to a width of 1,024 pixels and their height proportionally, given the huge resolutions used in herbarium images. This is an unbalanced dataset, as explained in next sections of this manuscript. In terms of genera, it contains a total of 498 genera, and regarding families it has a total of 124 families.

- ImageNet is the most widely used dataset by the machine learning research community. It contains 1,000 generalist classes and more than a million images (Russakovsky et al. 2015). It is the de facto standard for pre-training deep learning models. We use only the weights of a trained model with this dataset for transfer learning proposes.

4.2 Unbalanced dataset

Figure 10.2 shows how unbalanced the H1K dataset is. According to the work in (Carranza-Rojas et al. 2017a), the H1K dataset allows high identification rates with their deep learning model after 100 images per species. As shown in the figure, around 60% of the species have more tha 100 images, and 40% less than that. Some species have lots of images, for example 324 species have more than 300 images, but in contrast, 311 species have less than 11 images in total for both training and testing.

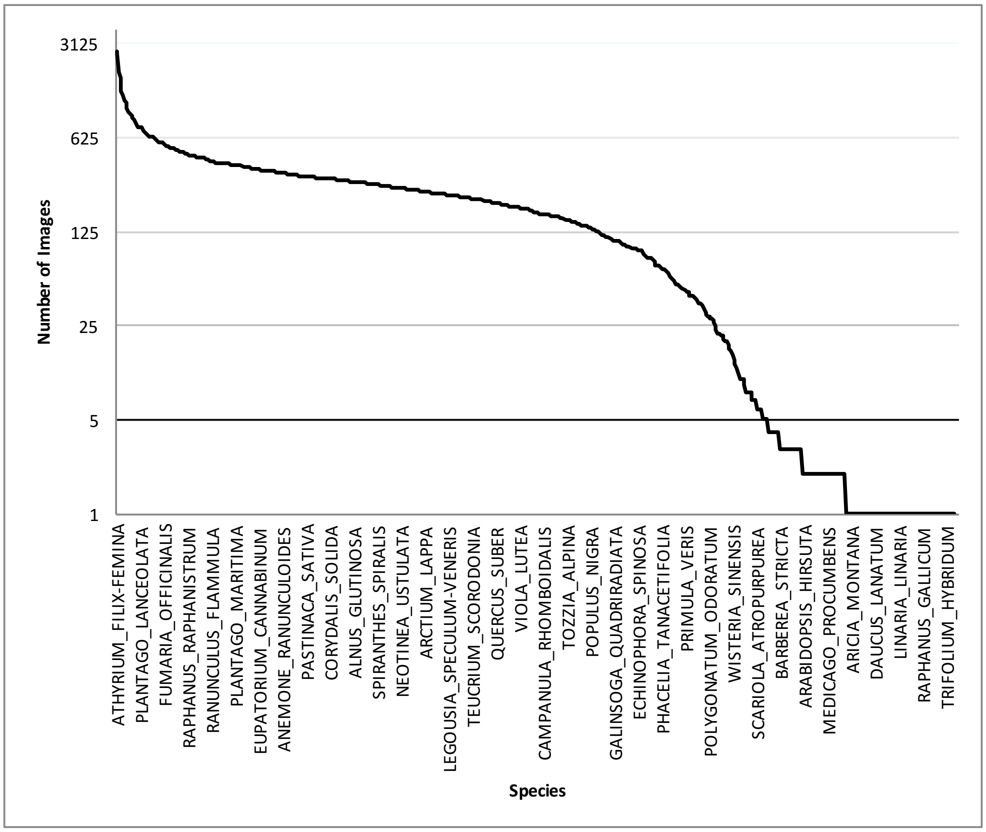


Figure 10.2. Image per class distribution showing an unbalanced H1K dataset

4.3 Architectures

The following architectures consist on a GoogleNet architecture with batch norm (Ioffe et al. 2015), as used in (Carranza-Rojas et al. 2017b) for plant identification on herbarium specimens. The main difference is at the last fully connected layer. Table 10.2 shows the modified GoogleNet network used in this research, taken from (Carranza-Rojas et al. 2017b), The network was implemented in Lasagne (Dieleman et al. 2015), using Theano (Theano Development Team 2016).

Table 10.2. GoogleNet architecture modified with Batch Normalization, taken from Carranza-Rojas et al. 2017b

Type	Patch size / Stride	Output Size	Depth	Params	Ops
convolution	7x7/2	112x112x64	1	2.7K	34M
max pool	3x3/2	56x56x64	0		
batch norm		56x56x64	0		
LRN		56x56x64	0		
convolution	3x3/1	56x56x192	2	112K	360M
max pool	3x3/2	28x28x192	0		
batch norm		28x28x192	0		
LRN		28x28x192	0		
inception (3a)		28x28x256	2	159K	128M
inception (3b)		28x28x480	2	380K	304M
max pool	3x3/2	14x14x480	0		
batch norm		14x14x480	0		
inception (4a)		14x14x512	2	364K	73M
inception (4b)		14x14x512	2	437K	88M
inception (4c)		14x14x512	2	463K	100M
inception (4d)		14x14x528	2	580K	119M
inception (4e)		14x14x832	2	840K	170M
max pool	3x3/2	7x7x832	0		
batch norm		7x7x832	0		
inception (5a)		7x7x832	2	1072K	54M
inception (5b)		7x7x1024	2	1388K	71M
avg pool	7x7/1	1x1x1024	0		
batch norm		1x1x1024	0		
linear		1x1x10000	1	1000K	1M
softmax		1x1x10000	0		

Baseline: Flat Classification Model (FCM)

In order to evaluate the performance of adding hierarchies to the architecture classification, first a base line is set based on a FCM. Since we are classifying not only species but also genera and families, the flat approach requires 3 different instances of the same model, with different number of outputs on the last dense layer and softmax, according to the dataset label size for each taxonomic level. Figure 10.3 shows the 3 main building blocks that will be used on the next sections with information about the models. For species we have a total of 1191 outputs, for genera 498 and for families 124. These output sizes are the same across all architectures.

Figure 10.4 shows how the flat model looks like. The model is basically a GoogleNet (Szegedy et al. 2015) model, modified with PRELU and batch normalization for faster convergence. A total of 3 different flat models were deployed: one for species, one for genera, and one for families. The 3 models are completely independent and do not share any parameters. They also have their own training and parameter update

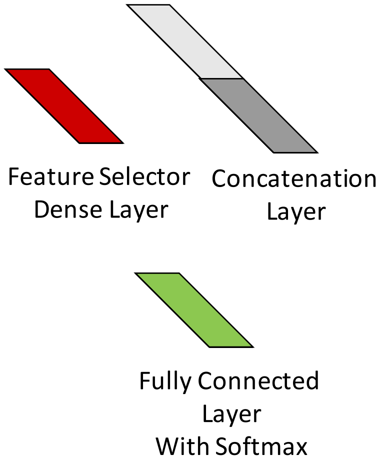


Figure 10.3. Representation of some building blocks of the different architectures.

process.

Multi-Task Classification Model (MCM)

Another approach to calculate accuracy at different taxonomic levels is by using a model where the different classifiers share the same deep network. MCM implements one classifier per taxonomic level, in this case 3 classifiers, one for species, one for genera and one for families. However, each classifier is connected to the last pooling layer of the GoogleNet model, allowing to do 3 classifications at the same time and sharing the same parameters of the model instead of having 3 separate models with their own parameters. The intuition behind is that the network will learn features from the 3 taxonomic levels at the same time. Figure 10.5 shows how a single main GoogleNet model is shared between 3 different classifiers, each one assigned to classifying a different taxonomic level. This model is inspired in the work of Goodfellow et al. 2014, where the authors identify multi-digit numbers from houses, using a classifier per digit.

TaxonNet: Hierarchical Classification

We present the following architecture that attempts to capture features at several levels of the plant taxonomy. We call this architecture TaxonNet, as it takes into account several levels of the plant taxonomy as the hierarchy, and uses knowledge of the previous taxonomic level classification for the next one, as shown in Figure 10.6.

We modified the GoogleNet model in the following fashion: the last fully connected layer which was used normally for a flat species classification is now used for the higher taxonomic level, in this case family. The loss of this fully connected layer will be calculated based on family labels of each image. Just before the softmax, the feature vector of the family fully connected layer output is concatenated with the last pooling layer feature vector. The idea behind this is to add a new fully connected layer for the genus, which will base its computations on both the family fully connected feature vector, and the raw feature vector coming from the CNN. Finally, we apply the same concept with the species: we add a new fully connected layer for species, which takes as input the concatenation of the genus fully connected layer output plus the last pooling layer feature vector from the CNN. In all cases, there is a middle feature selector layer in red, as shown in Figure

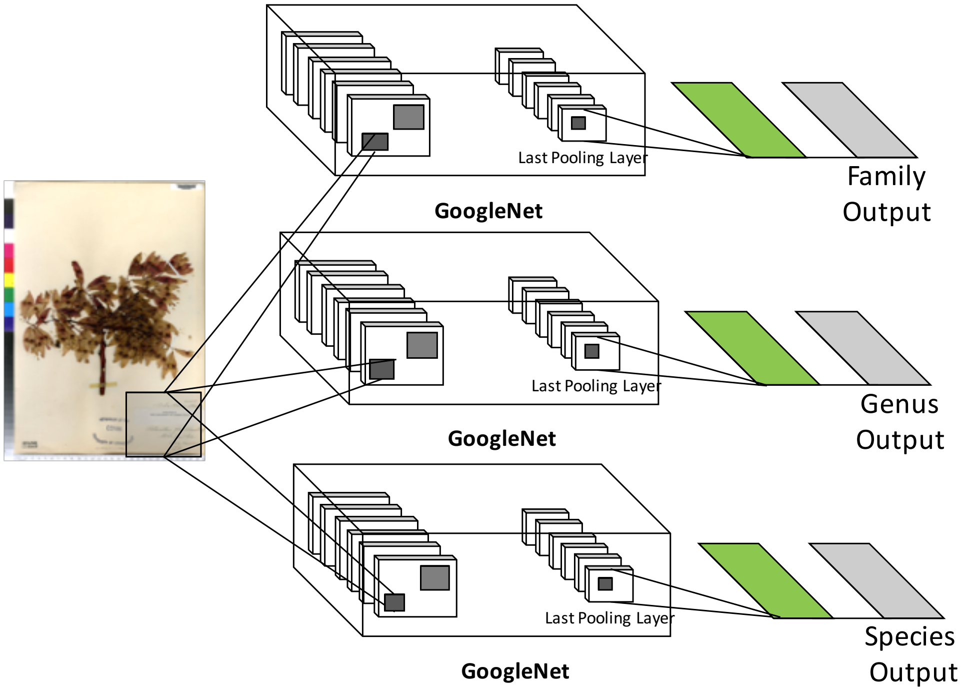


Figure 10.4. Separated Flat Classification Model (FCM) for species, genera and family

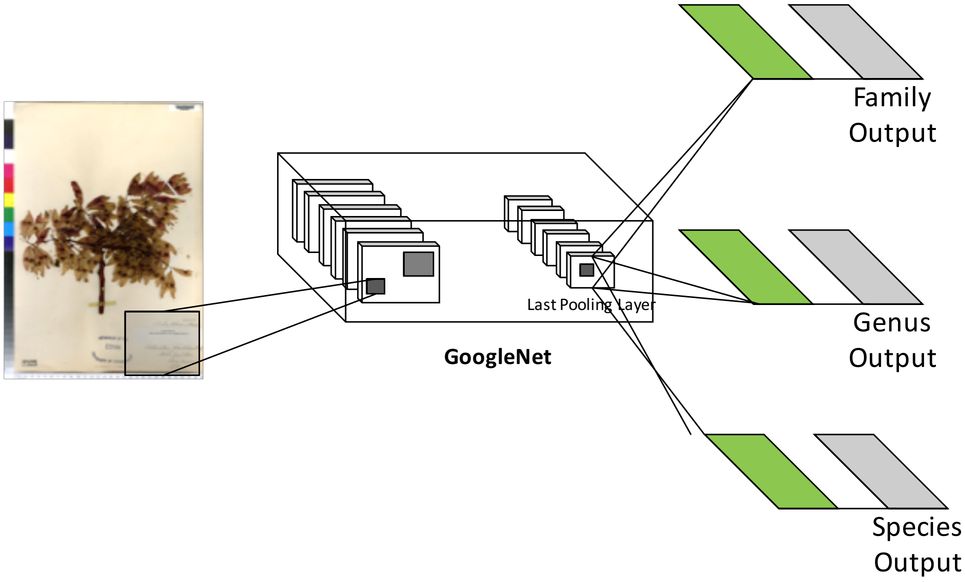


Figure 10.5. A Multi-Task Classification Model (MCM) for species, genera and family. Parameters are shared between the 3 taxonomic levels, similar to the work in (Goodfellow et al. 2014) for multi-digit identification on house numbers.

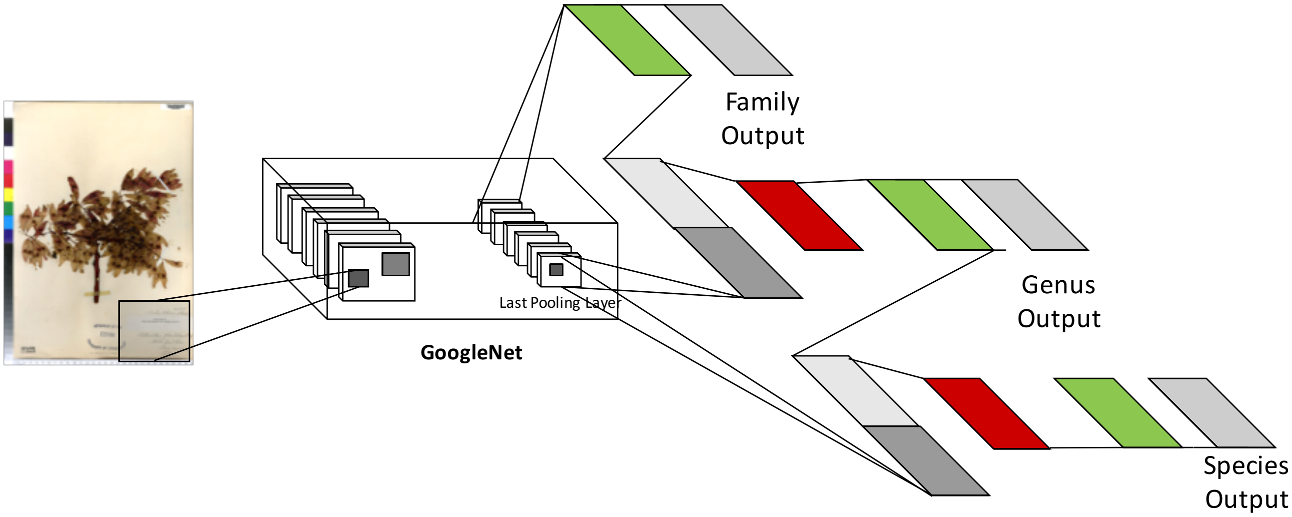


Figure 10.6. TaxonNet used to identify species, genera and family. The architecture allows to take into account previous classification of another taxonomic level for the next one

10.3, which allows the model to learn which features to take into account either from the original GoogleNet of from the previous taxonomic classification.

It is important to notice that the design allows the model to make mistakes at higher levels of the taxonomy, such as family or genus, but it can have good accuracy at the species level since it handles the raw feature vectors coming from the CNN. In other words, an error at higher levels of the taxonomy won't necessarily cause an error propagation to lower levels. It also allows to do classification at all taxonomic levels, thus, each one of them has its own loss which is back-propagated to the whole network. Our intuition is that the whole network learns features at all taxonomic levels, instead of having several complete CNN for each level. This of course allows for a smaller network to share parameters between levels.

5 Experiments

By using the previous explained models we ran several experiments to measure the effect of taking into account different taxonomic levels for the classification.

In all cases the used learning rate was 0.0075 and weight decay of 0.0002. The total number of training iterations was 6300 with a mini-batch size of 32 images, with 5 epochs. The number of validation iterations was 1500, same as for testing iterations.

5.1 Baseline Experiments: Flat Classification Model (FCM)

The first experiments are based on running the separated models for species, genus and family without sharing any type of parameters. This is considered the baseline, as there are no hierarchical characteristics at all, but just 3 models completely independent from each other.

5.2 Architecture Comparison Experiment

This experiment consists on a comparison of the different architectures at the different taxonomic levels. The experiment compares the MCM approach, where parameters are shared between the different classifiers, with the intention to see how the accuracy and loss behaves as the model is trained, compared to separated model per taxonomic level.The TaxonNet architecture is also compared with the separated models and the MCM aproach.

6 Results

6.1 FCM Baseline Results

First experiments consisted on running 3 separated FCM models to explore the loss and accuracy behavior at each taxonomic level. We consider this the baseline results as they are flat classifiers that do not share any hierarchical characteristics in terms of classification.

The results for all the FCM are shown in Figure 10.7. In particular, FCM for species gets Top-1 63.02% and Top-5 is about 82.93% as shown in Figure 10.7a. In case of the FCM for genus the accuracy goes up to Top-1 of 70.51 and Top-5 of 87.85%, as shown by Figure 10.7b. For the family, Figure 10.7c shows the best results for both Top-1 and Top-5, with 75.55% and 93.43% respectively.

It is important to notice that both genus and family show an improvement compared to the species. This makes sense as genus and family have more images per class and also both models have less classes, 498 for genus and 124 for families.

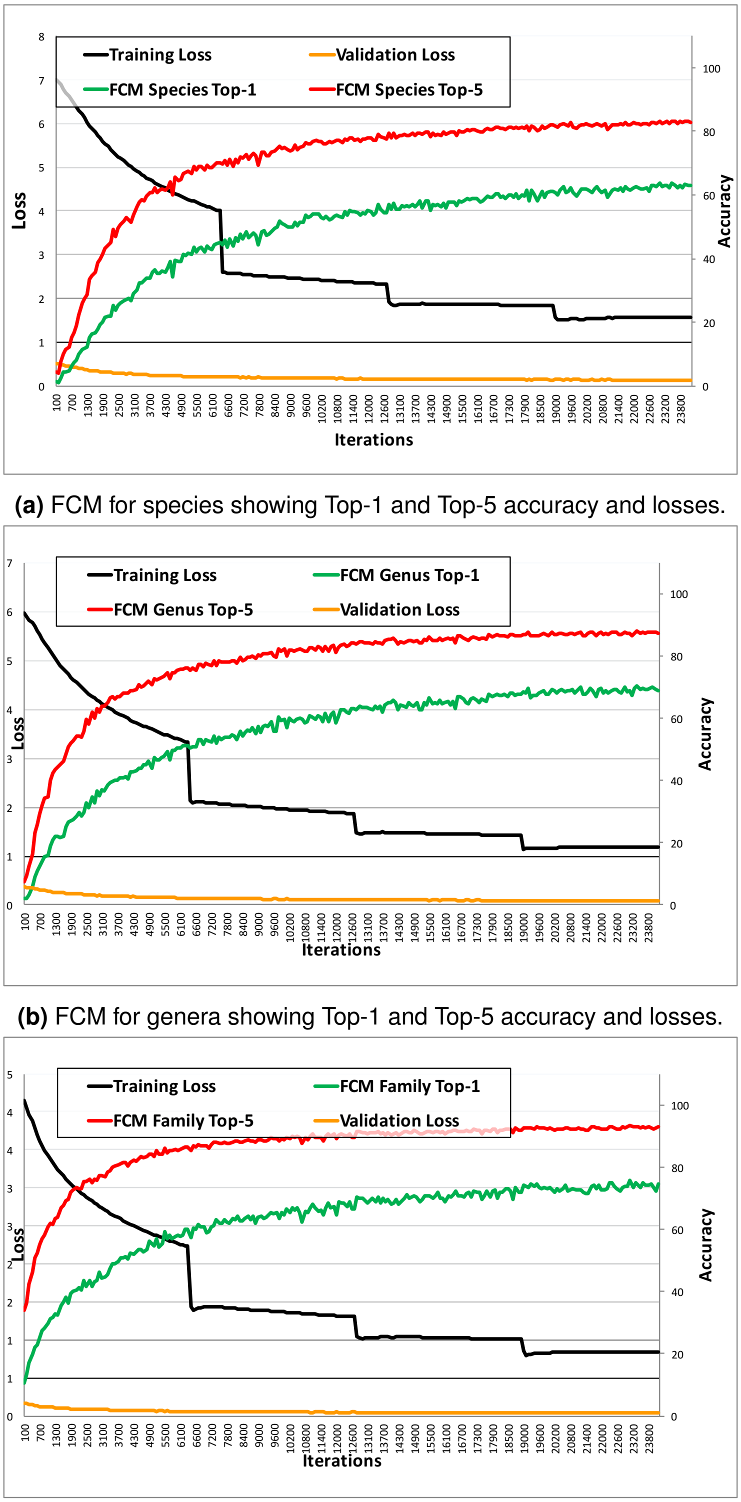
6.2 Architecture Comparison Results

MCM Top-1 and Top-5 behavior

In case of the MCM architecture, for Top-1 accuracy the results show 64.32% for species, 75.95% for genus, and for family 88.17%, as shown by Figure 10.8a. The parameter sharing allows the model to predict the family with a very high accuracy. In case of Top-5 accuracy, MCM results in 71.66% for species, 83.23% for genus, and 92.99% for family, again being the family classification the best among the 3, as shown by Figure 10.8b.

TaxonNet Top-1 and Top-5 behavior

In Figure 10.9a, TaxonNet architecture shows for Top-1 accuracy 62.39%, 76.23%, 86.92% for species, genus and family, respectively. Again, similarly to MCM, the parameter sharing allows the model to predict the genus and family with a very high accuracy. For Top-5 accuracy, as shown by Figure 10.9b, TaxonNet results in 70.20%, 82.36% and 92.80% for species, genus and family, respectively, again being the family classification the best among the 3.

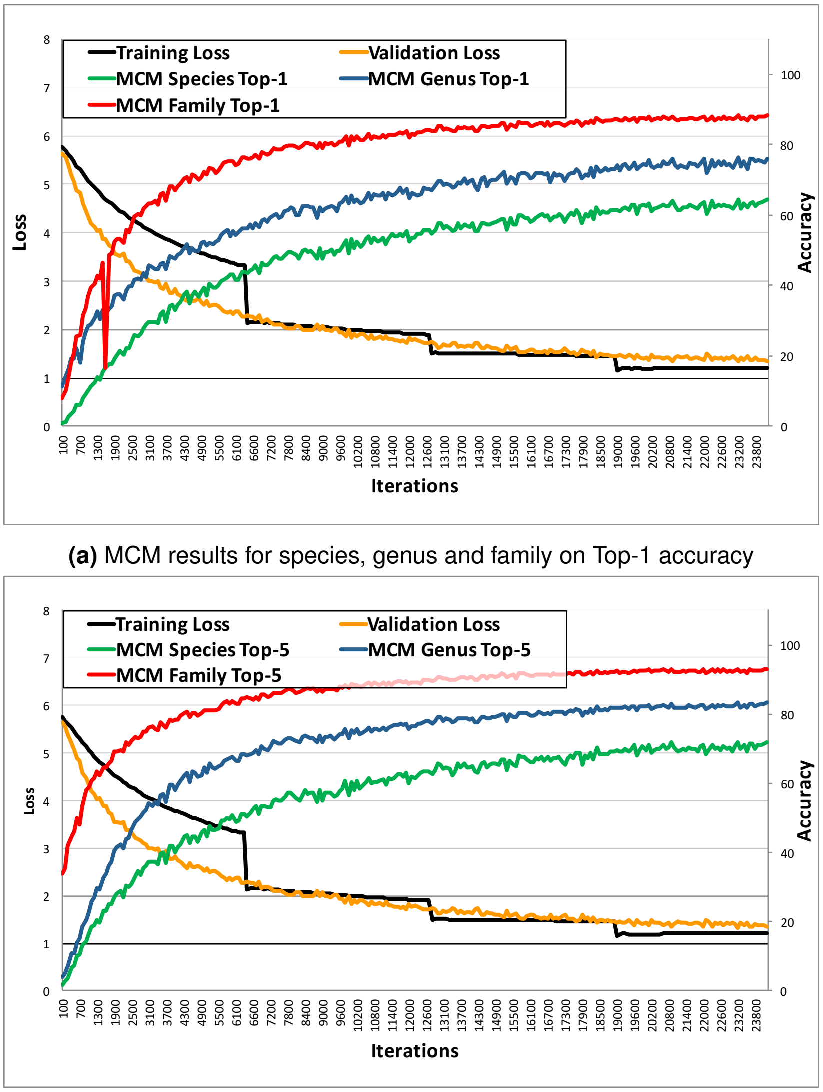


(a) FCM for species showing Top-1 and Top-5 accuracy and losses.

(b) FCM for genera showing Top-1 and Top-5 accuracy and losses.

(c) FCM for family showing Top-1 and Top-5 accuracy and losses.

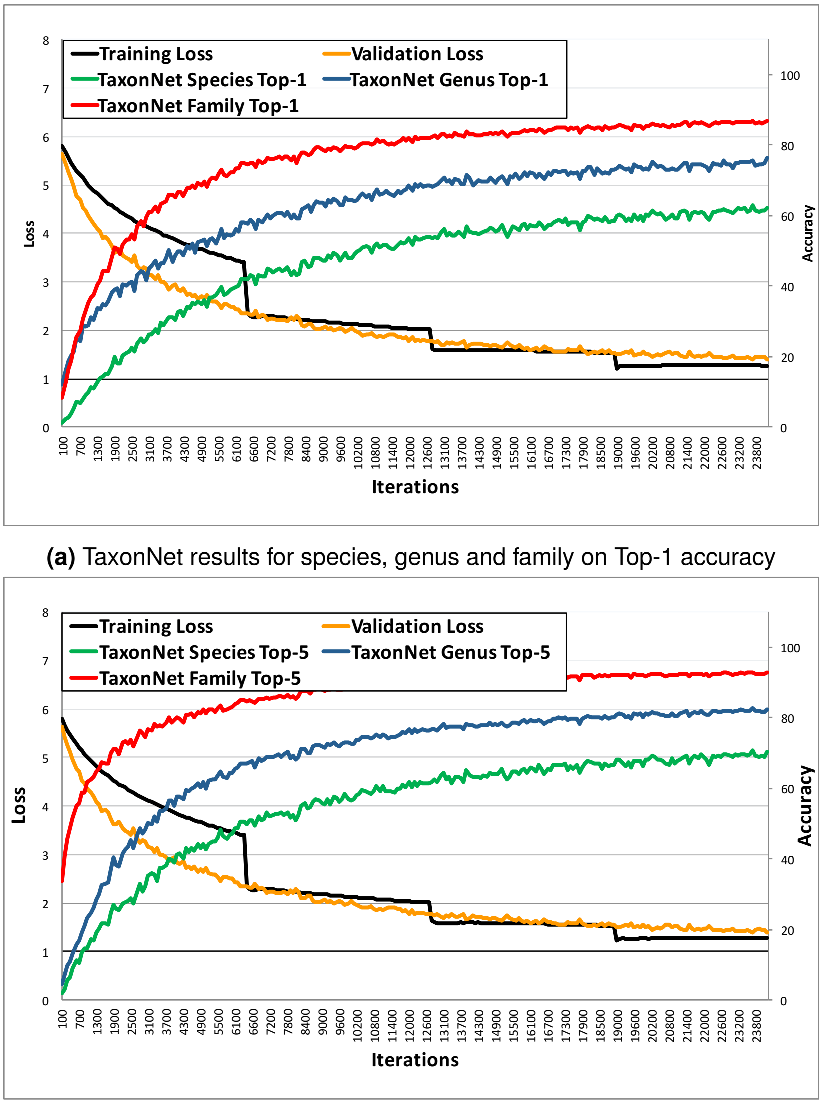
Figure 10.7. The 3 instances of the FCM architecture, one for each taxonomic level



(a) MCM results for species, genus and family on Top-1 accuracy

(b) MCM results for species, genus and family on Top-5 accuracy

Figure 10.8. Results for MCM architecture, both Top-1 and Top-5 for species, genus and family.



(a) TaxonNet results for species, genus and family on Top-1 accuracy

(b) TaxonNet results for species, genus and family on Top-5 accuracy

Figure 10.9. Results for TaxonNet architecture, both Top-1 and Top-5 for species, genus and family.

Architecture Comparisons

Figure 10.10 shows the Top-1 accuracy comparison between FCM, MCM and TaxonNet, regarding species, genus and family.

For species, Top-1 accuracy is 63.02%, 64.32%, 62.39% for FCM, MCM and TaxonNet, respectively, showing the best results on the MCM architecture by a margin of 1% approximately.

Regarding genus, Figure 10.10b shows a Top-1 accuracy is 70.51%, 75.95%, 76.23% for FCM, MCM and TaxonNet, respectively. In this case, the degradation of the flat classifier for the genus is improved significantly by both hierarchical architectures, with the TaxonNet being the best one.

Finally, for family, Figure 10.10c shows shows a Top-1 accuracy is 75.55%, 88.17%, and 86.92% for FCM, MCM and TaxonNet. Here the improvement is very strong compared to the flat classifier on Top-1.

7 Conclusions

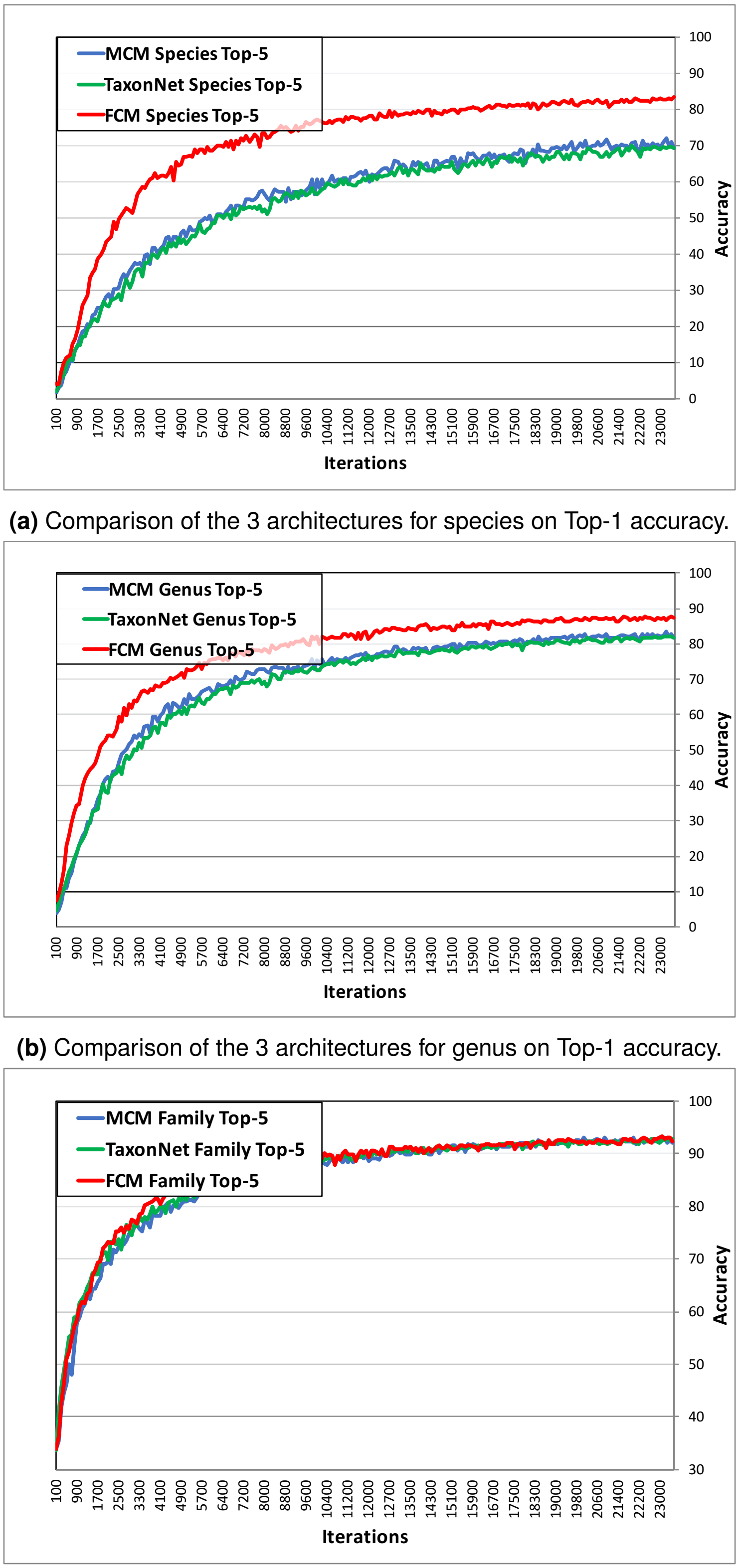
The best accuracy results for species and genus are provided by the independent Flat Classification Model (FCM), but at the cost of 3 times more GPU consumption as well as 3 times more parameters. In case of the family, both the Multi-Task Classification Model (MCM) and TaxonNet architectures provide similar results to the flat model.

8 Future Work

This work uses knowledge of higher levels of taxonomy for species classification, and allows to classify at higher levels of the taxonomy such as genus and family. However, it uses traditional fully connected layers with traditional cross entropy loss and softmax calculations. Next steps include exploiting the class hierarchy to calculate a different loss functions using perhaps Bayesian approaches of hierarchical softmax functions. Also, hierarchical regularization terms could be defined to regularize the loss calculation using the class hierarchy. Also, interesting future experiments include understanding how using the taxonomy impacts the classification of new, unseen classes, at higher taxon levels. For instance, a species may not have been included during training but the genus related to that species may be, thus, allowing the system to provide an identification at that level. Additional architectures are also needed to be explored such as Long-Short Term Memory (LSTM) based architectures for the taxonomy.

Acknowledgement

This research was partially supported by a machine allocation on Kabré supercomputer at the Costa Rica National High Technology Center. We also thank the Costa Rica Institute of Technology for the financial support for this research.



(a) Comparison of the 3 architectures for species on Top-1 accuracy.

(b) Comparison of the 3 architectures for genus on Top-1 accuracy.

(c) Comparison of the 3 architectures for family on Top-1 accuracy.

Figure 10.10. Comparison of the 3 architectures at each taxonomic level.

Chapter 11

Taxonomy-Softmax: A Hierarchical Loss Function for Deep Automatic Plant Identification

Reference No yet published.

Keywords Deep Learning, Loss Functions, Hierarchical Classification, Taxonomy, Biodiversity Informatics, Automated Plant Identification

1 Abstract

Loss functions that exploit a class hierarchy have not been widely studied in the deep learning literature, with some notable exceptions such as Hierarchical Softmax (H-Softmax) in the Natural Language Processing (NLP) domain. Instead, "flat" classification and "flat" loss calculation have been the norm. However, when prior taxonomy knowledge is available, higher levels of classes may provide important information to guide the model optimization. Human taxonomists do use the plant taxonomy for plant identification. Fortunately, most datasets include an accessible underlying taxonomy. In contrast, researchers and practitioners in the automatic plant identification domain do not have tackled the possibility of using the taxonomy as a class hierarchy that can guide the model parameter optimization. In this research we propose a new loss function named Taxonomy Softmax (T-Softmax), that takes into account hierarchies of classes to guide the model optimization. We used the PlantCLEF 2015 and the Herbarium255 (H255) datasets. The obtained loss has a tendency to be higher since it punishes the model for low scores at the ancestor class level as well as the class level to be classified. We experiment with plant images and use the plant taxonomy as the prior known class hierarchy. We classify at the species level, but the loss function provides additional information based on higher hierarchy levels such as genera and family. We demonstrate the feasibility of using the hierarchy to guide the optimization of the deep learning model parameters to achieve, in some cases, better accuracy results. Our results also suggest T-Softmax serves as a regularization method. We focus on plant taxonomy as our prior known class hierarchy, leveraging the knowledge provided by centuries of research in the botanical domain.

2 Introduction

Most deep learning research has left out class hierarchies during the model loss minimization phase. Thus, flat classification and flat loss functions are prevalent. In particular, to our knowledge, most research on image-based automated plant identifications does not exploit other taxonomic levels besides the species level. However, in the biological domain, the approach traditionally followed by taxonomists is intrinsically hierarchical. Single-access and multiple-access identification keys are an example of such an approach (Mata-Montero et al. 2016). They are mostly used for species identification, but can also be used for higher class level classification such as genera or family. In fact, a single-entry identification key could be defined such that, along the way, as the user moves from the root question towards the leaves of the decision tree, identifications at higher taxonomic levels are determined.

Plant image datasets are often very taxonomically unbalanced at the species level. This is the case either because there are enough images but they have not been digitized, or – most likely – because some plant species are hard to find in-situ in a given area (Carranza-Rojas et al. 2018; Mata-Montero et al. 2016). However, most datasets do have an underlying taxonomy that was used to label each image with a given scientific name. Having taxonomically unbalanced image datasets introduces biases in experiments and limits the power of deep learning approaches, where, in principle, large amounts of images are critical.

In other Artificial Intelligence tasks such as Natural Language Processing (NLP) tasks, some forms of hierarchical loss functions have been developed, such as H-Softmax, but in this case it is based on a balanced binary tree, a data structure that does not fit in the plant taxonomy domain.

In this research we propose a new loss function that takes into account hierarchies of classes by using probabilities of the classes and of their ancestor classes. We experiment in particular with plant images and use the plant taxonomy as the prior known class hierarchy. We classify at the species level but the loss functions provide additional information based on higher hierarchy levels such as genera and family. We demonstrate the feasibility of using the hierarchy to guide the optimization of the deep learning model parameters to achieve slightly better accuracy results. Our results also suggest that, by comparing accuracy during training and testing, T-Softmax works as a regularization method, avoiding overfitting, even in presence of dropout. We focus on the plant taxonomy as our prior known class hierarchy, leveraging the knowledge provided by centuries of research in the botanical domain. Our main objective is to improve accuracy and to provide an additional method to do regularization based on the class hierarchy.

This paper is organized as follows: Section 3 presents related work about hierarchical loss functions and classification. Section 4 covers the methodology, including datasets, hardware, models, and a mathematical formulation. Section 5 describes the experiment design. Section 6 summarizes the results obtained. Section 7 presents the conclusions and, finally, Section 8 summarizes future work.

3 Related Work

Literature about using hierarchies of classes for automatic classification is scarce (Silla et al. 2011). Furthermore, hierarchical loss techniques are almost nonexistent in terms of deep learning approaches (Mnih et al. 2009; Shahbaba et al. 2007; Yan et al. 2015).

In the automatic plant identification domain, challenges such as PlantCLEF (Goeau et al. 2017) and also applications such as Pl@ntNet (Joly et al. 2016a) focus on flat species identification only. If multi-level classification was taken into account, then it might be feasible to provide better results at genus or family levels, when species estimation is not accurate enough.

The PlantCLEF challenge has become the most influential world-wide for automatic plant identification based on images (**Goeau2015**; Goeau et al. 2017; Joly et al. 2016c). To date, participants have focused on species-level identification only, leaving other taxonomic levels aside. This challenge has been key not only to improve the state of the art vis a vis automatic image-based plant identification algorithms, but also to develop large plant image datasets. We use the PlantCLEF 2015 dataset in our experiments to test our new hierarchical loss functions.

A softmax function using a bayesian form is proposed in (Shahbaba et al. 2007). It introduces a prior probability and correlations between nearby classes in the hierarchy. The authors test their approach in classical machine learning but it could easily be adapted to current deep learning technologies by implementing a new optimizable layer.

A very complete survey regarding hierarchical classification is presented in (Silla et al. 2011). They attempt to unify the existing hierarchical classification approaches, which do not include hierarchical loss functions. Additionally, the survey does not cover current deep learning technologies.

In general, the most familiar hierarchical loss function is Hierarchical Softmax (H-Softmax) (Mnih et al. 2009), a widely used loss function in Natural Language Processing (NLP) tasks. Such function makes use of a balanced binary tree to find words at the leaves. Middle nodes are functions that guide the word search in a binary fashion. This changes the computational cost of finding the correct class from linear to logarithmic. The tree is not necessarily associated with prior knowledge, as it can be generated based on clustering or other techniques. Also, since the tree is balanced and binary, it means each tree node has two descendants or less. This is useful in natural language settings, where similar words can be put within made-up nodes which do not have necessarily a concrete meaning beyond just grouping similar words. However, in domains such as plant identification, a plant taxonomy is represented by a tree that is neither balanced nor binary. This makes the pure H-Softmax definition unsuitable for such domain.

In 2015, Yan et al. 2015 develop a deep learning architecture called Hierarchical Deep Convolutional Neural Network (HD-CNN), which, given p parent classes, creates p sub-classifiers (dense layers) of size m, where m is the amount of descendants per parent. This allows training to be done based on a two level hierarchy, but it is difficult to scale up to more than two levels. Additionally, the number of dense layers grows exponentially as p increases. It also leads to error propagation: if the model does not learn to detect well enough the parent class, it will classify wrong descendants classes.

Two years later, Carranza-Rojas et al. 2017a ran deep learning models to identify plants on herbarium sheet datasets from Costa Rica and France. They provided results on identification at three taxonomic levels: species, genus, and family, but they run a separate model for each class level. Accuracy results for genera and families were better than species. Since separate models are ran per hierarchy level, the class hierarchy is not exploited to improve species identification by using higher level class knowledge.

In 2018, Carranza-Rojas et al. 2018 take into account plant taxonomies to do automatic plant identification by proposing new architectural approaches in deep learning to learn several classifiers at the same time. They propose architectures with different dense layer configurations called Multi-Task Classification Model (MCM) and TaxonNet, which allow simultaneous multi-level classification. This allows to optimize the model for multi-level classification during a single training session instead of training several models for each class level of the hierarchy. Additionally to the training time reduction by keeping just one set of model parameters, the architectures show a slight increase in Top-1 accuracy of 1.3% for species, 5.72% for genus and 12.62% for families, compared with the Flat Classification Model (FCM), which are separated models per class level. This work covers architectural approaches to the hierarchical classification, but does not cover the creation of new loss functions to drive the model optimization.

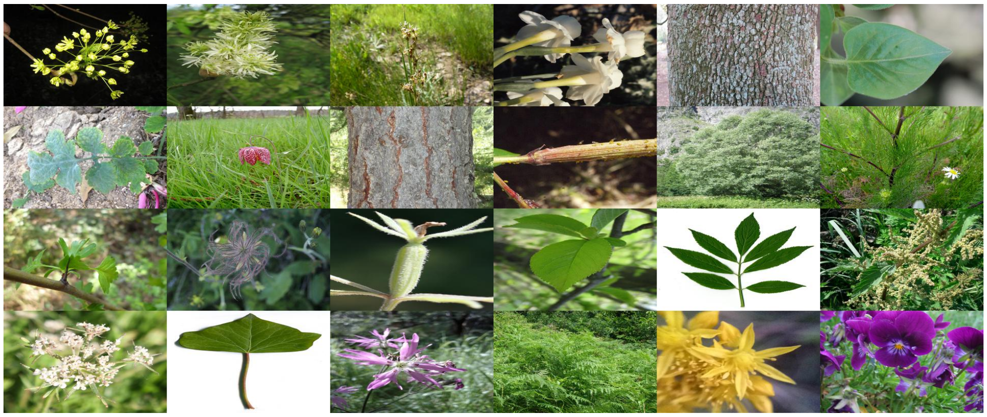


Figure 11.1. Random sample of different plant images available in the PlantCLEF 2015 dataset challenge.

4 Methodology

4.1 Datasets

We tested our hierarchical loss functions with two datasets, namely, the PlantCLEF 2015 dataset and the Herbarium255 (H255) dataset created in (Carranza-Rojas et al. 2017b). The former consists of 91,759 images for training and 21,446 for testing (Goëau et al. 2015). In total, it comprises 1,000 species, 516 genera, and 124 families. We used the labels of this dataset to build the class hierarchy needed to run the hierarchical loss functions. Figure 11.1 shows a random sample taken from the dataset. Images are mostly taken in-situ without any particular protocol. Images may come even from non-experts.

The Herbarium255 (H255) dataset consists of herbarium sheet images from plants collected in Costa Rica. It includes 203 species, 158 genera, and 66 families. It comprises a total of 11,071 images. The dataset was built based on the iDigBio (*iDigBio* 2017) database. We randomly took a 20% sample of the images for testing.

4.2 Implementation and Hardware

The initial parameters of the model were taken from a previous training with ImageNet (Russakovsky et al. 2015), as part of the provided functionality from PyTorch to initialize models with pre-trained parameters (Paszke et al. 2017). The ImageNet dataset itself was not used for any other process beyond parameter initialization. PyTorch was selected as the framework to be used given its autograd capabilities, useful to minimize the new loss layers. All PyTorch implementation code can be found online [1]. We run our models in a NVIDIA Tesla k40 with 16 gbs of DDR5.

4.3 Deep Learning Model

We use a ResNet18 architecture to run our experiments, proposed in (He et al. 2016). ResNet has been the winner architecture during the PlantCLEF 2017 competition (Goeau et al. 2017), as well as during the ImageNet 2015 challenge (Russakovsky et al. 2015). ResNet152 was also a potential candidate but given its huge size it was complicated to find the proper hardware to run it. This type of architecture allows to

[1] https://github.com/maeotaku

learn residual functions that reference the inputs of each layer, instead of directly the inputs. These residual functions consist on identity shortcut connections between the input and the output of a layer.

4.4 Mathematical Formulation

Wu et al. 2005 define a *hierarchy* as an ordered set $(T, \prec)$, where T is a finite set that contains all class concepts in the application domain (in our case, all species, genera, families, etc) and $\prec$ denotes the "IS-A" relationship. Based on this definition, we extend the "IS-A" relation as follows:

- Let $(T, \prec)$ be a strictly partially ordered set, where T is a finite set that enumerates all class concepts in the application domain. Symbol $\prec$ denotes the "IS-AN-ANCESTOR-OF" relation.
- By definition of $(T, \prec)$, $\prec$ is irreflexive, asymmetric, and transitive. Thus, for all classes p, q, and r in set T, we have that

$$p \nprec p \tag{11.1}$$

$$p \prec q \implies q \nprec p \tag{11.2}$$

$$p \prec q \wedge q \prec r \implies p \prec r \tag{11.3}$$

- In general, a strictly partially ordered set corresponds to a Directed Acyclic Graph (DAG). However, in this research, we consider only DAGs that have the topology of a forest, i.e., a collection of directed rooted trees. We say that the root of each tree is at $level$ 1, and other levels are defined naturally according to conventional graph theory. We also assume that each directed tree has the same number of levels.
- Set T can therefore be expressed as the union of n mutually exclusive sets T_1, T_2, $\ldots$, T_n, where T_k is the set of all classes at level k for all $1 \leq k \leq n$.

Descendant Sets

For all classes p in T, we define the set $C(p)$ as all descendant classes of class p, that is,

$$C(p) = \{q \mid p \prec q\} \tag{11.4}$$

In Equation 11.5 we define $C(p, k)$ as the set of all descendants q at level k of class p.

$$C(p,k) = \{q \in T_k \mid p \prec q\} \tag{11.5}$$

Thus, in particular, $C(p, n)$ represents all descendants of class p at the deepest level n. For our domain, the deepest hierarchy level includes all species, and p may be a particular genus, family, order, or any other taxon at higher levels.

Taxonomy Softmax (T-Softmax) Definition

Traditional cross entropy loss, also known as log loss L, for a training/testing item is defined by Equation 11.6. Let m be the amount of classes at hand. Let $\vec{y}$ be the one-hot vector of size m with a 1 at the correct answer index and 0 elsewhere. Let $\hat{\vec{y}}$ be the vector of size m of predictions for all classes. The term $\vec{y}_i$ represents the expected binary answer for the i-th class, while $\hat{\vec{y}}_i$ represents the predicted value for the i-th class.

$$L(\vec{y}, \hat{\vec{y}}) = -\sum_{i=1}^{m} \vec{y}_i \ln \hat{\vec{y}}_i \tag{11.6}$$

The scalar $\hat{\vec{y}}_i$ is calculated using softmax as shown in Equation 11.7 (Bishop 1995). Scalar $\vec{x}_i$ represents the i-th value of the input vector $\vec{x}$ of size m.

$$\begin{aligned} \hat{\vec{y}}_i &= P(\vec{x}_i) \\ &= \frac{e^{\vec{x}_i}}{\sum_{j=1}^{m} e^{\vec{x}_j}} \end{aligned} \tag{11.7}$$

The complete softmax distribution estimate $\hat{\vec{y}}$ of the input vector $\vec{x}$ is shown in Equation 11.8.

$$\begin{aligned} \hat{\vec{y}} &= P(\vec{x}) \\ &= \sum_{i=1}^{m} P(\vec{x}_i) \\ &= \sum_{i=1}^{m} \frac{e^{\vec{x}_i}}{\sum_{j=1}^{m} e^{\vec{x}_j}} \end{aligned} \tag{11.8}$$

In Equation 11.9 we calculate the probability of class p as the sum of the probabilities associated with each descendant class q at level k.

$$P(p, k, \vec{x}) = \sum_{q \in C(p,k)} P(\vec{x}_q) \tag{11.9}$$

Finally, Equation 11.10 shows the T-Softmax formalization. We multiply the softmax probability of class q at level n (last level of the hierarchy) by the probability of its ancestor p at level k. This is expected to reduce the probability of the class q depending on the probability of the ancestor class p, which is also based on the siblings of q. We also parameterize the T-Softmax by a hierarchy level k of our choice, allowing to re-define the siblings of q at different levels of the taxonomy.

$$P(q, k, \vec{x}) = P(\vec{x}_q) P(p, n, \vec{x}) \mid p \prec q \wedge p \in T_k \tag{11.10}$$

In our particular case, the value of k will guide the equation to take into account the genus or family level. Also, in our experiments, all classes q will correspond to species.

Log Loss with T-Softmax

We adapt the cross entropy loss to T-Softmax as shown in Equation 11.11. We replace the original estimation $\hat{\vec{y}}$ with the estimation using T-Softmax. T-Softmax is calculated for each class q with respect to its ancestor at level k.

$$L(\vec{y},k,\vec{x}) = -\sum_{q=1}^{m} \vec{y}_q \ln P(q,k,\vec{x}) \qquad \textbf{(11.11)}$$

The T-Softmax equation will force the model to be optimized based not only on having estimated the correct class q (as traditional softmax), but also it will depend on the ancestor class p having a high probability. Furthermore, the ancestor class p will have a high probability only if the sum of all siblings of q and q itself have a high probability. In other words, T-Softmax will force the model to be optimized based on a high score of the correct class at the lowest hierarchy level, but also on high scores of its siblings according to the desired ancestor class at a given higher level.

5 Experiments

We measure the accuracy changes when using the proposed hierarchical loss function using T-Softmax compared to the traditional cross entropy loss with softmax. The cross entropy loss function with softmax is referred as the baseline. We also measure the loss behavior and how it decreases as the model is optimized using the proposed loss function.

In case of T-Softmax, we also calculate not only T-Softmax with genus but also with family, in order to understand if higher hierarchy levels lead to better or worst accuracy gains and also how loss behavior is affected.

Additionally, by comparing the behavior of T-Softmax during training and testing, we can also measure if there are any regularization effects.

The architecture used in all experiments was ResNet18. In all cases the used learning rate was 0.0085 and weight decay of 0.09. The mini-batch size used was 32 images, with 100 epochs. The resolution of all images was resized to a standard of $224x224$ pixels. We used dropout at 0.5 as well.

In all cases, we logged the training loss, test loss, as well as both training and test Top-1 and Top-5 accuracy.

5.1 The PlantCLEF Experiment

This experiment ran the training sessions with the PlantCLEF 2015 dataset. We run a ResNet-18 baseline using cross entropy with softmax and another ResNet-18 architecture using our T-Softmax loss function. We measured training and testing losses, as well as training and testing top-1 and top-5 accuracy. The T-Softmax was calculated with genus and family levels to measure changes when using distinct levels of the class hierarchy.

5.2 The Herbarium Experiment

This experiment evaluates the training and testing with the H255 dataset. Top-1 and top-5 accuracy achieved by T-Softmax versus cross entropy with softmax were measured. The dataset was split at 20% for

Table 11.1. Results of species classification with the PlantCLEF dataset, baseline (softmax) versus T-Softmax. A regularization effect is noticed in particular for family, where during training the accurcy goes down compared when the baseline, but goes up during testing.

Hierarchy Level (k)	Phase	Baseline Top-1	T-Softmax Top-1 / Boost	Baseline Top-5	T-Softmax Top-5 / Boost
Genus	Training	78.27	**79.67** / **1.40**	92.99	**93.77** / **0.78**
Genus	Testing	45.54	**47.84** / **2.30**	68.75	**70.23** / **1.47**
Family	Training	78.27	77.21 / -1.05	92.99	92.46 / -0.52
Family	Testing	45.54	**47.55** / **2.01**	68.75	**70.42** / **1.66**

the testing set, with the rest for training.

6 Results

This section summarizes the results achieved in the PLantCLEF and Herbarium experiment. In particular, Table 11.1 summarizes the results for T-Softmax with the PlantCLEF dataset and Table 11.2 for the Herbarium255 (H255) dataset.

6.1 Results of PlantCLEF experiment

Training results at genus level

Table 11.1 summarizes the results for the PC experiments. In general, results show a general improvement over the softmax baseline. Figure 11.2 depicts the results of training the baseline with softmax, versus using T-Softmax at the genus level. It shows that during training, top-1 and top-5 accuracy do not gain a big increase when T-Softmax. The loss ranges at bigger numbers with T-Softmax compared to the baseline. This is expected as T-Softmax is more strict with respect to not only estimating the right species, but also estimating right the siblings of such species at the ancestor class level, in this case, genus.

Testing results at genus level

Figure 11.3 shows the results of testing top-1, top-5 accuracy, and loss for both T-Softmax with genus and the baseline with the PlantCLEF 2015 dataset. During testing, T-Softmax shows a slightly better accuracy. Top-1 accuracy goes up from 45.85% to 46.47% during epoch 100 and Top-5 accuracy also increases from 68.75% to 69.15%.

Training results at family level

Figure 11.4 shows the top-1, top-5, and loss during training with the PlantCLEF 2015 dataset. It compares the baseline and T-Softmax. It can be noticed that the baseline provides better accuracy results for both top-1 and top-5.

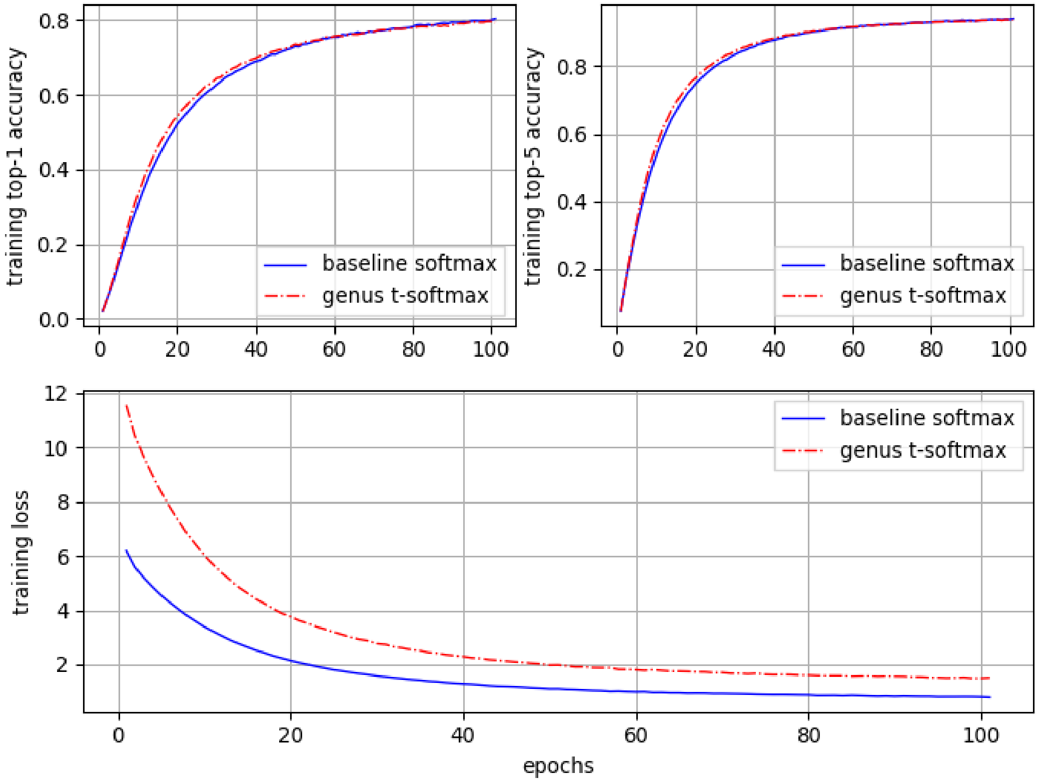


Figure 11.2. Training top-1 and top-5 accuracy and losses. Blue line represents the traditional cross entropy with softmax, red line is T-Softmax at the genus class level. Training was done with PlantCLEF 2015 dataset.

Testing results at family level

Testing results with family and PlantCLEF 2015 are shown in Figure 11.5. Top-1, top-5, and loss are measured for both the baseline and T-Softmax. During the last epoch, top-1 and top-5 accuracy increase with T-Softmax. Testing Top-1 increases from 45.85% to 46.84% and top-5 goes from 68.75% to 69.71%. In general, the plot shows a monotonic accuracy increase during all last epochs. The comparison between Figure 11.4 and Figure 11.5 suggests that there is a regularization effect applied by the T-Softmax, as testing top-1 and top-5 accuracy are better than training, compared to the baseline. This regularization effect can also be noticed in Table 11.1. Loss shows also a more strict loss regimen compared to the baseline, but does not sacrifice accuracy.

6.2 Results on the Herbarium255 (H255) dataset

Training results at genus level

Table 11.2 summarizes the results for H255. At genus level, changes in training top-1 and top-5 accuracy are small. Figure 11.6 shows the training top-1 and top-5 accuracy and the training loss behavior at the genus

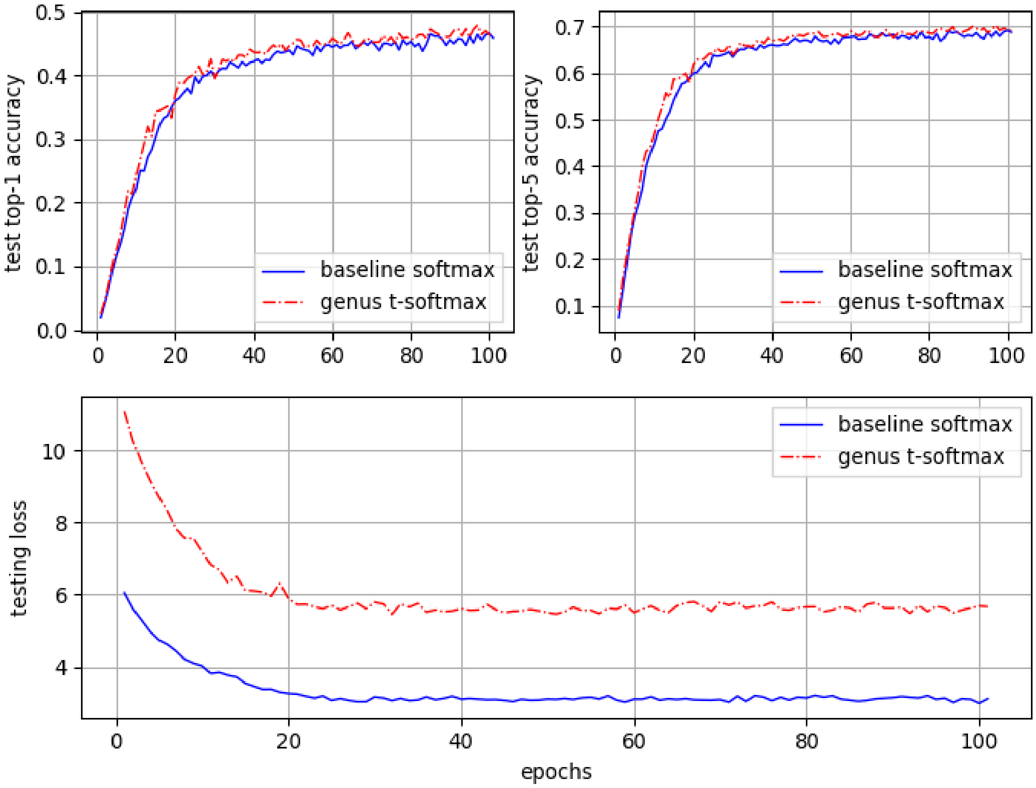


Figure 11.3. Testing top-1 and top-5 accuracy and losses. Blue line represents the traditional cross entropy with softmax, red line is T-Softmax at the genus class level. Testing was done with PlantCLEF 2015 dataset.

level. Top-1 shows a slight increase in accuracy with T-Softmax at the last ephocs. The loss, as expected, is bigger with T-Softmax.

Testing results at genus level

For testing, Figure 11.7 shows the testing top-1 and top-5 accuracy as well as the testing loss. To the end of the 100 epochs there is a slightly bigger testing accuracy with the baseline. Top-1 goes from 55.53% to 53.95% while top-5 decreases from 76.18% to 73.39%.

Training results at family level

Figure 11.8 shows the training top-1 and top-5 accuracy, and the training loss behavior at the family level. In general, the difference of top-1 accuracy and top-5 accuracy during training does not have a clear improvement when using T-Softmax. The loss is bigger in case of T-Softmax, but this does not sacrifice any type of training accuracy.

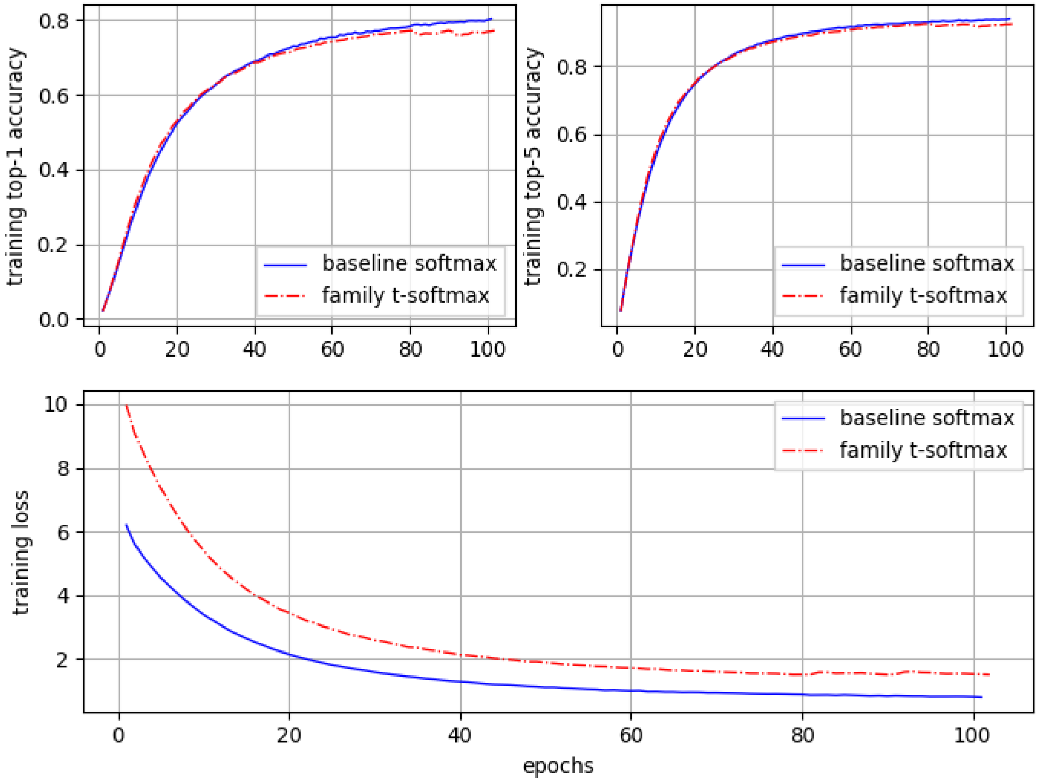


Figure 11.4. Training accuracy and losses for top-1 and top-5. The blue line corresponds to the traditional cross entropy with softmax and the red line is T-Softmax at the family class level. Training was done with the PlantCLEF 2015 dataset.

Testing results at family level

For testing, Figure 11.9 shows the testing top-1 and top-5 accuracy as well as the testing loss. At the end of the 100 epochs there is a bigger testing accuracy improvement with the T-Softmax at family level, compared to the baseline. Top-1 accuracy goes up from 55.53% to 59.25% in epoch 100. Top-5 accuracy goes from 76.18% to 78.79%. It is clear that at the family level with the H255 dataset, the T-Softmax improves the testing accuracy, regardless of having bigger test losses during training. The bigger test loss can be interpreted as a more strict loss regimen, where the optimization is forced towards not only having a good species estimation, but also to have high scores for the species siblings, guided by the ancestor class level.

We can observe how the T-Softmax improves testing top-1 and top-5 accuracy with the H255 dataset and also with the PlantCLEF 2015 dataset. In both cases, losses are bigger for T-Softmax, but they do not imply lower accuracy readings. As stated before, even when the loss is bigger, testing top-1 and top-5 accuracy improve. In case of PlantCLEF 2015, there seems to be a regularization effect when compating training and testing results.

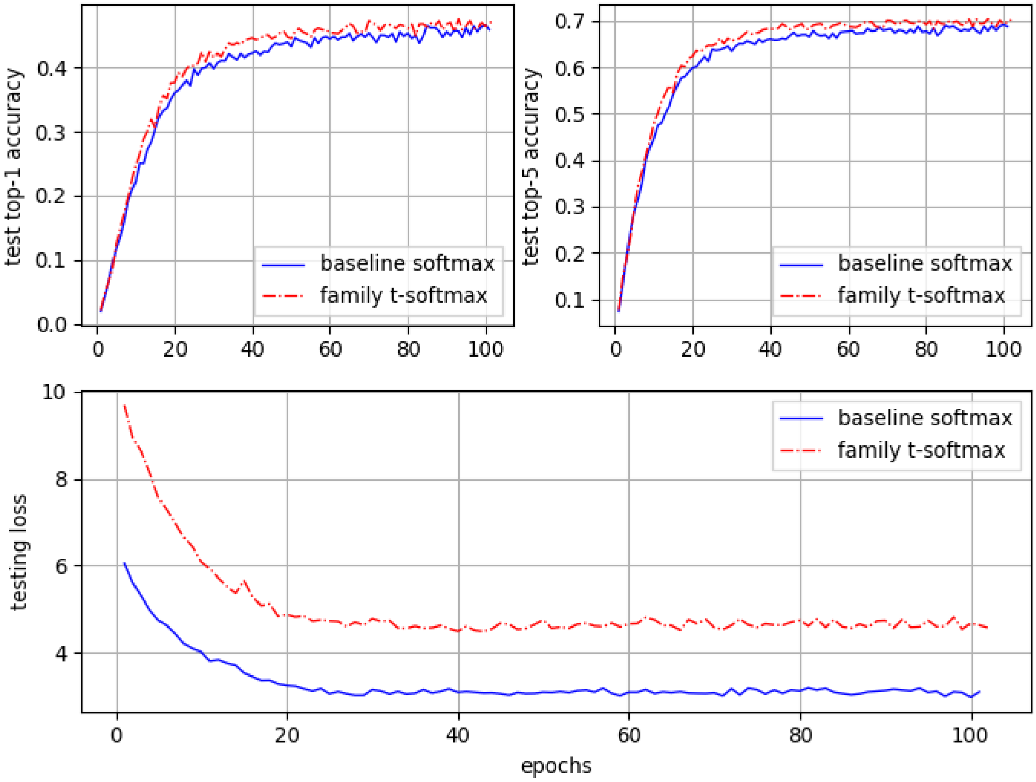


Figure 11.5. Testing top-1 and top-5 accuracy and losses. Blue line represents the traditional cross entropy with softmax, red line is T-Softmax at the family class level. Testing was done with PlantCLEF 2015 dataset.

7 Conclusions

A new loss function has been derived from a hierarchy of classes. The Taxonomy Softmax (T-Softmax) provides higher loss values, forcing the correct class to have the higher score together with its siblings, both with respect to the ancestor class node. This reflects a more strict regimen of calculation of the loss with the support of ancestor classes.

The training top-1 and top-5 accuracy have a tendency to not vary from the softmax baseline, and even in some cases to be worst. However, during the testing phase T-Softmax shows an increase in both top-1 and top-5 accuracy, making the new loss function suitable for accuracy gains on real scenarios, when a hierarchy of classes is present.

The fact that the accuracy gain at higher hierarchical levels (family) was better during testing than lower levels (genus) indicates that the label distribution across the class hierarchy may affect the behavior of T-Softmax.

Finally, the fact that during some experiments the softmax baseline provided higher accuracy during training, but lower accuracy than T-Softmax during testing, suggests that T-Softmax serves also as a regularization method, even in presence of other regularization methods such as dropout.

Table 11.2. Results of species classification with the Herbarium255 (H255) dataset, baseline (softmax) versus T-Softmax. Genus shows adverse results however at family level the accuracy boost goes up to 4.93% during testing.

Hierarchy Level (k)	Phase	Baseline Top-1 %	T-Softmax Top-1 % / Boost %	Baseline Top-5 %	T-Softmax Top-5 % / Boost %
Genus	Training	84.50	**86.24** / **1.74**	97.33	97.23513 / -0.09
Genus	Testing	55.72	54.97 / -0.74	76.19	74.325584 / -1.86
Family	Training	84.50	**85.08** / **0.58**	97.33	**97.42** / **0.09**
Family	Testing	55.72	**60.65** / **4.93**	76.19	**79.72** / **3.53**

8 Future Work

Several avenues of future research have been opened by this research and the T-Softmax concept. Even new layers (beyond loss functions) driven by the class taxonomy can be tested. A hierarchical layer could be proposed with parameters that relate the species classes to higher class levels such as genera or family, somehow similar to Attention mechanisms (Luong et al. 2015).

Additionally, more experiments are needed to understand the effects of using the class hierarchy in other domains and with other, more generic datasets, such as ImageNet.

Also, merging several T-Softmax from different hierarchy levels can be further explored.

The effects of T-Softmax on the top-k scores is worth studying. Intuition suggests that T-Softmax could improve the estimation of the top-k classes, not only of the correct class, thanks to the optimization with the ancestor class. This could translate directly in a better top-k list from a user experience perspective, where the correct class siblings could have better positions in the top-k ranking.

Finally, accuracy seems to increase directly by using T-Softmax at different hierarchy levels. For instance, accuracy gain might be higher as one uses higher levels of the hierarchy. This has been the case with the genus and family level, but remains open for higher levels.

Acknowledgement

This research was partially supported by a machine allocation on Kabré supercomputer at the Costa Rica National High Technology Center (CeNAT). We also thank the Vice-Rectorate of Research and Extension of the Costa Rica Institute of Technology for the support of this research project under project number 1370002.

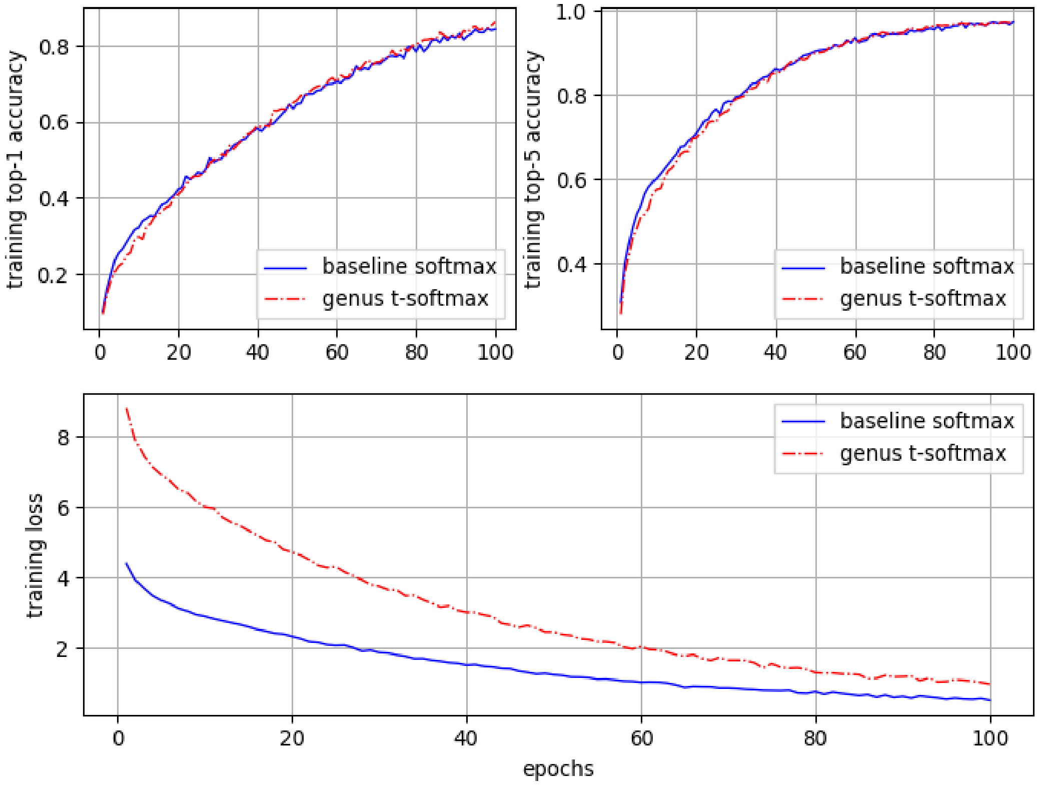


Figure 11.6. Training top-1 and top-5 accuracy and losses. Blue line represents the traditional cross entropy with softmax, red line is T-Softmax at the genus class level. Training was done with H255 dataset.

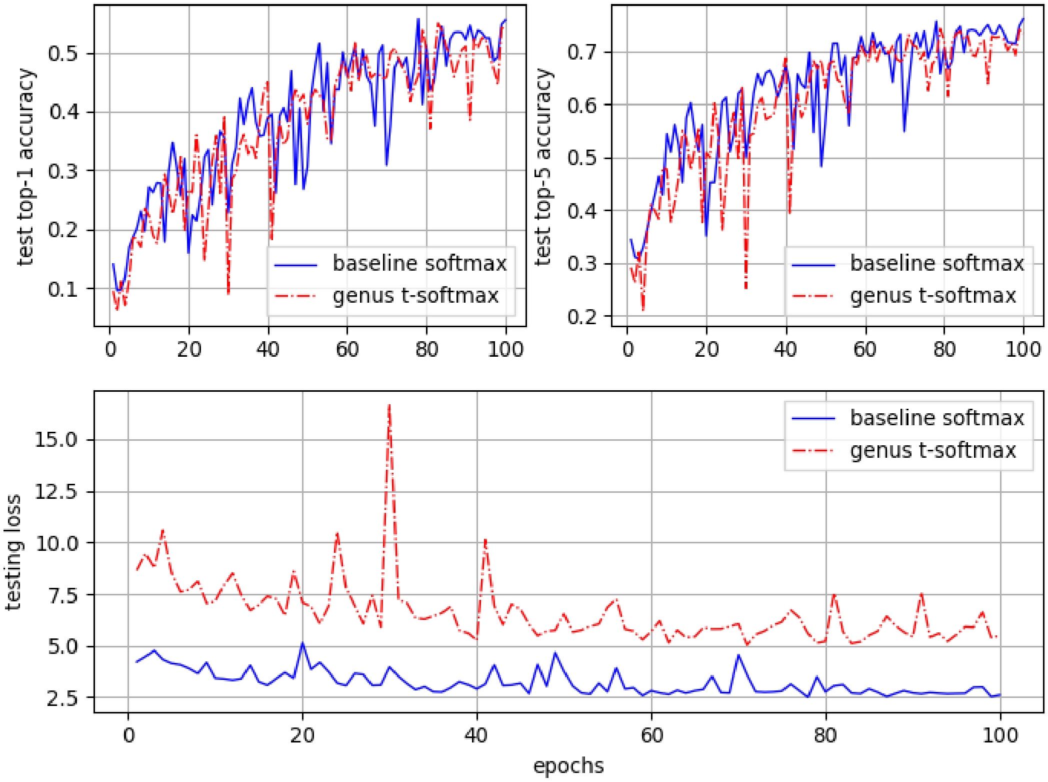


Figure 11.7. Testing top-1 and top-5 accuracy and losses. Blue line represents the traditional cross entropy with softmax, red line is T-Softmax at the genus class level. Testing was done with H255 dataset.

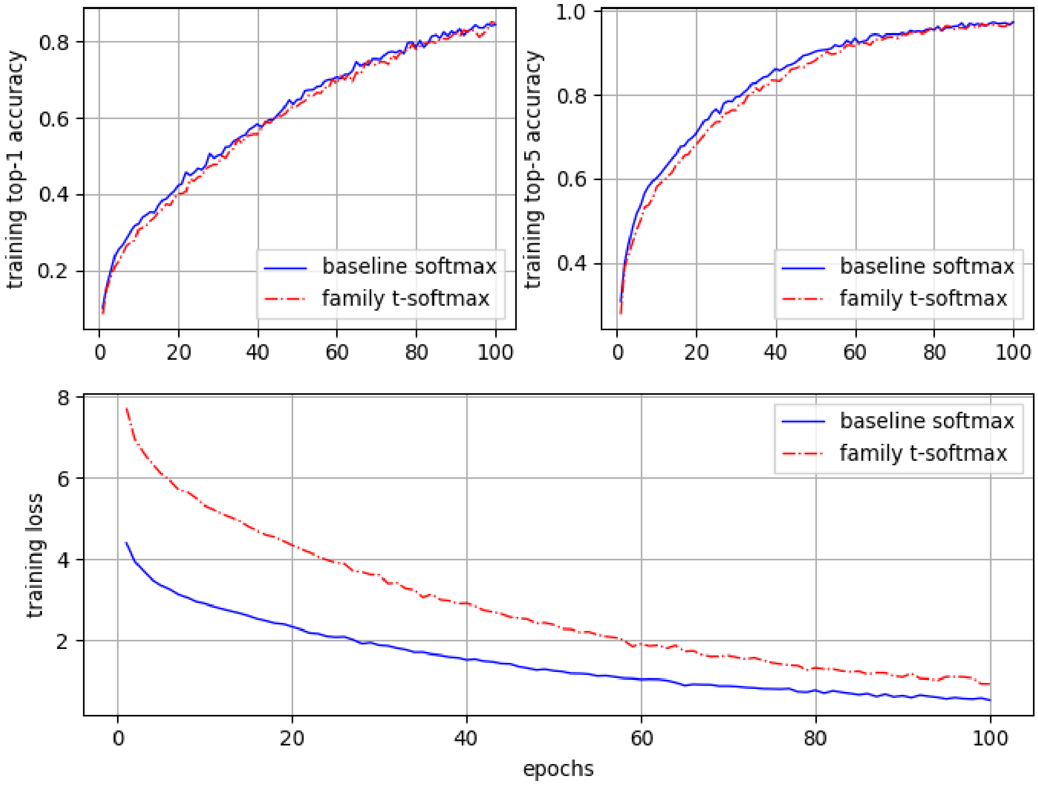


Figure 11.8. Training top-1 and top-5 accuracy and losses. Blue line represents the traditional cross entropy with softmax, red line is T-Softmax at the family class level. Training was done with H255 dataset.

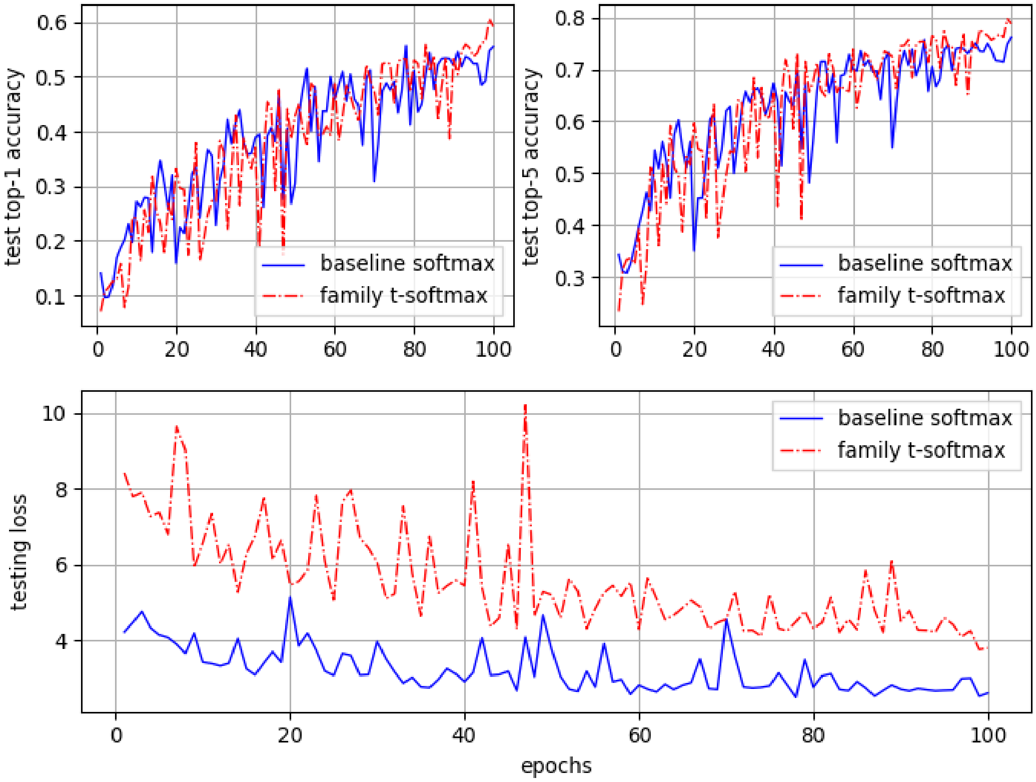


Figure 11.9. Testing top-1 and top-5 accuracy and losses. Blue line represents the traditional cross entropy with softmax, red line is T-Softmax at the family class level. Testing was done with H255 dataset.

Chapter 12

Conclusions and Future Work

> Science knows no country, because knowledge belongs to humanity, and is the torch which illuminates the world.
>
> Louis Pasteur

1 Conclusions

This chapter covers the conclusions of this research as a whole.

Scaling up automatic plant identification with Deep Learning

As explained in Chapter 6, over the last five years, there has been an obvious shift from hand-crafted traditional computer vision approaches to Deep Learning methods for automatic plant identification. Initially, hand-crafted feature extractors provided good identification results and accuracy for relatively small datasets (with less than a hundred species and few thousand pictures). In Chapter 4, by adding texture features in the form of Local Binary Pattern Variance (LBPV), there were improvements ranging from 14.1% to 25.5% of accuracy, depending on the value of k, using a kNN classifier. Similarly, with a noisy subset (pseudo-scan images), the improvement ranges from 35.5% to 42.5%. These improvements showed how the curvature of the leaf was not enough to get better results. Then, better identification results were obtained in Chapter 7 by using Convolutional Neural Networks (CNNs), even when the datasets are far bigger in terms of number of classes. Undoubtedly, Deep Learning has become the state-of-the-art for automatic plant identification.

It is also important to notice that the traditional approach used with hand-crafted feature extractors does not scale up as well as Deep Learning approaches. This was confirmed with the work presented in Chapter 9, where the accuracy obtained between hand-crafted and Deep Learning methods is compared.

Higher concentration of vein and additional patterns improves accuracy

In Chapter 5 we conclude that the leaf side is a significant factor for automatic plant species identification (Carranza-Rojas et al. 2016b). On the average, and for most cases, underside pictures lead to more accurate identifications, most likely because of the more prominent vein patterns present. For 61.9% of the studied species, classification is better if the sample to be identified is a back side leaf image and 23.8% of the species got better results when the front side image was used. In agreement with intuition, the worst accuracy is obtained when the model is trained with back side images and tested with front side images and vice-versa. This is an interest finding as citizen user may always use the frontal side of the leaves to take pictures, while the back seems to be provide better results in most species.

In contrast, because the curvature of the front side and the back side of a leaf are mirror images of each other, this feature is not sensibly affected by which side of the leaves are used.

Existence of easily classifiable species

In Chapter 4 we show how some species can be identified more easily compared to others, based on leaf images and traditional hand-crafted feature extractors. This, however, is also true for newer Deep Learning approaches as shown in Chapter 8, where we found species that have high accuracy of identification regardless of having a small amount of images in the herbarium sheets domain (Carranza-Rojas et al. 2017a). These species are of special interest since they may have very particular visual features that allow an easier identification and may allow further research to understand what the Deep Learning models are actually learning. Consequently, we could compare the Deep Learning model with human taxonomists.

Dataset inherent biases affect automatic identification

With the exception of PlantCLEF publications, most reports on experiments about automated image-based identification systems do not explicitly state that accuracy has been measured while avoiding the Same-Specimen-Picture Bias (SSPB). However, if not taken into account, this bias can dramatically cause an overfit of the data. Therefore, future automated image-based plant identification experiments and reports should explicitly address what measures were taken to vis a vis this bias.

We demonstrated that automated image-based plant identification can be very sensitive to the Same-Specimen-Picture Bias (SSPB), as shown in Chapter 9. We studied two specific cases, one uses Deep Learning and the other uses a hand-crafted feature extraction approach. In both cases, SSPB introduces a very significant bias. Given the fact that users of a production system will most likely take pictures of specimens that were not used for training, it is realistic to assume that the training and testing phases should resemble that scenario, i.e., the testing phase should not assess accuracy by using pictures of specimens that were also used in the training phase, even if the pictures are different.

In Chapter 7, we do take into account similar biases, this time in terms of herbarium sheet author and date. It is imperative that new studies in automatic plant identification state how they avoid potential biases beyond the ones found and studied in this research.

Herbarium images are suitable for automatic plant identification

In Chapter 6, we argued that building a global dataset is a must, in order to identify any plant on earth. Despite the success of global biodiversity informatics initiatives such as GBIF, EOL, iDigBio and BHL, a global dataset of digital images of plant elements and complete plants does not exist. Our results in Chapter 7 show the potential of herbarium sheet images for automatic plant identification. This is the first study that analyzes a big dataset with thousands of specimen images from herbaria and uses Deep learning to identify them.

Additionally, based on our results, we believe that the development of Deep Learning technology based on herbarium data, together with the recent recognition of e-publication in the International Code of Nomenclature

(Nicolson et al. 2017) will also contribute to significantly increase the volume of descriptions of new species in the following years.

Herbarium institutions gain additional value thanks to our research

As a very important herbaria financial sustainability side effect, herbaria around the world have now more arguments to demonstrate the value and impact of maintaining and investing in their collections as their usefulness in automatic plant identifcation has been proven in Chapter 7 and Chapter 8.

Several other studies been done based on this research, such as further exploration of herbarium images and trait information ("Herbarium data: Global biodiversity and societal botanical needs for novel research"), and new literature reviews of the state-of-the-art such as (Wäldchen et al. 2018), among others. This validates the importance of the new provided datasets of herbarium images, as well as their use for automatic plant identification. This could lead to the creation of a semi, or even fully, automatic system to help taxonomists and experts do their annotation, classification, and revision work at herbaria.

Transfer Learning between different global regions provides state-of-the-art results

Overall, transfer learning was used from ImageNet to both in-situ images as well as herbarium images, as shown in Chapter 8 and Chapter 11. This is a common technique used nowadays to improve accuracy. However, Chapter 7 shows that it is possible to use a herbarium image dataset from one region of the world to do transfer learning to another region, even when the species do not match, getting similar accuracy results as with ImageNet-based transfer learning. This indicates that a Deep Learning approach could be used in regions that do not have lots of herbarium images, by doing transfer learning from regions that do have lots of data. This is quite suitable for regions rich in biodiversity but poor in herbarium images.

Additionally, in Chapter 7 we showed that it is not beneficial to do transfer learning from herbarium data to leaf scan pictures and it is even counterproductive to do transfer learning from herbarium data to field images. This confirms some previous studies in the literature that concluded that the observable morphological attributes can change significantly with the drying process. Additionally, the particular unnatural layout of plants and their parts on herbarium sheets may also have a negative effect.

Multi-level architectures save training time for multiple taxon identification

As shown in Chapter 10 multi-level architectures cost three times less Graphics Processing Unit (GPU) consumption, and need three times less parameters (in this case we used three levels of the taxonomy). The multi-level architectures have the plus of using only a single set of parameters. These architectures allow to identify several class levels at the same time.

Multi-level architectures provide better accuracy in some cases

In Chapter 10 the Multi-Task Classification Model (MCM) model shows better top-1 accuracy results by a slight margin of 1.3% compared to running an independent species model. This is a small improvement at the species level. However in case of genus identification, the TaxonNet improves the top-1 accuracy by 5.72% from 76.23% to 70.51%, showing a high increase by using the multi-level architecture in cascade. Similarly for family, the usage of multi-level architecture improves a lot more, with top-1 accuracy going from 75.55% with the stand alone model to 88.17% in case of MCM, an increase of 12.62%. This demonstrates the feasibility of creating architectures that are trained with several levels of the class hierarchy at the same time, while not sacrificing accuracy, and gaining training time while keeping the same parameter size.

Taxonomy Softmax (T-Softmax) provides a more strict loss regimen for model optimization

In Chapter 11 T-Softmax results in good correct class scores only if the ancestor class also has a good probability. Furthermore, the ancestor class will have a high probability only if the correct class siblings also have a high score. Thus, cross entropy with T-Softmax forces a more strict loss regimen, where the hierarchy guides the calculation. This is reflected in higher loss values, compared with a softmax baseline.

T-Softmax provides accuracy increases

Chapter 11 describes how the results of using T-Softmax during testing provides higher top-1 and top-5 accuracy than the softmax baseline. This happens even when the T-Softmax shows bigger loss values, meaning such bigger loss values do not compromise the accuracy of the model. The accuracy gain in some cases is bigger than in others, so more research needs to be done towards understanding why this happens.

T-Softmax is affected by the label distribution

In Chapter 11 the accuracy increases at the family level are better than the increases at genus level. This indicates that the label distribution across the class hierarchy may affect the behavior of T-Softmax. Of course, the data distribution also affects the new loss function behavior, in the sense of the number of images per ancestor class.

T-Softmax serves as a regularization method

One of the biggest contributions of this dissertation is the regularization effect found with the usage of T-Softmax in Chapter 11. In general, the accuracy obtained with T-Softmax was higher during testing compared to the softmax baseline, in contrast with the equal or lower accuracy obtained during training. This shows that T-Softmax has a regularization effect, allowing better testing results which makes it suitable for real world scenarios. Finally, the regularization effect is noticed even when dropout also exists, allowing both regularization methods to co-exist.

2 Future Work

Model herbarium drying process

Concerning the question of how herbarium data could be useful for field images classification, we believe we should rather try to model the drying process itself typically by learning a transfer function between a representation space dedicated to herbarium images and another one dedicated to field images. This might be achieved by using generative models such as Generative Adversarial Network (GAN) or Autoencoders, such as the image-to-image translation work of (Zhu et al. 2017).

Opening the Deep Learning black box

It is important to understand what these Deep Learning models are exactly "learning" (Carranza-Rojas et al. 2017a). This in order to have insights of the similarities between what human taxonomists use and what the Deep Learning models learn. To achieve this, a possible avenue is to use Deconvolutional Networks (DN) to understand what pixel patterns are influencing the final plant identification. This can be shown as heat maps where red pixels have the most influence in the final decision and blue ones the least. Human taxonomists then can help in defining how these patterns match their own human expertise, to show 2 things: first, if the model is learning to capture the same visual characteristics used by human experts to identify such species, and secondly, if perhaps the model may have learned features that were not know yet by human experts. This could even provide insights of new ways to develop taxonomic keys based on new patterns discovered by the Deep Learning models.

Additionally, DN could allow to correct the model in case it is not learning only plant-related related features, but other elements that may cause biases, such as hand-written labels, numbers, color palettes, among others.

Create specialized software for herbarium institutions

As shown by this research Deep Learning algorithms and herbarium specimen images are suitable for automatic plant identification (Carranza-Rojas et al. 2017a,b). As such, tools for herbarium-related experts can be created using this technology, to help in the tedious work of classifying all herbarium sheets availble, but yet identified properly.

Create models for automatic captioning of plant images

Image captioning is a very exciting research topic nowadays. It could be feasible to automatically assign botanical descriptions to herbarium images (initially), and then to in-situ images, by exploiting the descriptions associated with the images that are provided by expert taxonomists. After our contributions towards automatic plant identification using herbarium sheet images, an obvious next step is to use their descriptions to train image captioning models so one can automatically not only identify species but also describe the contents of the plant image. In case of herbarium it is easier since most of the time, datasets may have text descriptions.

Build a global dataset of plant images

Additional ways to gather plant species information need to be exploited in order to increase the size and completeness of a global dataset for plant identification (Mata-Montero et al. 2016). Crowd-sourcing is a good way to get more data by exploiting citizen science. Similarly, a web crawling approach can also help to build such a global dataset by using images available online from several websites. Both need a post-phase of quality assurance after the image acquisition, but we must rely on citizen science to get more and more data as taxonomists and botanist are running low.

Expand the hierarchical loss function formulation

It is feasible to create new variations of T-Softmax on top of this research by changing the equations to different styles. Variations with learnable parameters can be explored, that relate the species classes to ancestor class levels such as genera or family, somehow similar to Attention mechanisms (Luong et al. 2015). Merging several T-Softmax from different hierarchy levels can be further explored to see how the loss behaves when it is calculated at two or more hierarchy levels at the same time.

Additionally, more experiments are needed to understand the effects of using the class hierarchy in other domains and with other, more generic datasets, such as ImageNet.

Study how the top-k class list is affected by T-Softmax

Since T-Softmax is calculated based on the correct class plus its ancestor class and its sibling classes, intuition says the top-k correct classes may change compared to a cross entropy with softmax baseline. In other words, the obtained list of best suitable species may be better and it is worth studying further.

Bibliography

Adams, Dean C., F. James Rohlf, and Dennis E. Slice (2004). “Geometric morphometrics: ten years of progress following the ‘revolution”. In: *Italian Journal of Zoology*, pp. 5–16 (cit. on p. 67).

Aggarwal, N. and R. K. Agrawal (2012). “First and Second Order Statistics Features for Classification of Magnetic Resonance Brain Images”. In: *Journal of Signal and Information Processing* 3.2, pp. 146–153. DOI: 10.4236/jsip.2012.32019 (cit. on pp. 4, 23, 102).

Andreopoulos, Alexander and John K. Tsotsos (2013). “50 Years of object recognition: Directions forward.” In: *Computer Vision and Image Understanding* 117.8, pp. 827–891. URL: http://dx.doi.org/10.1016/j.cviu.2013.04.005 (cit. on pp. 23, 68).

Arora, Akhil, Ankit Gupta, Nitesh Bagmar, Shashwat Mishra, and Arnab Bhattacharya (2012). “A Plant Identification System using Shape and Morphological Features on Segmented Leaflets: Team IITK, CLEF 2012.” In: *CLEF (Online Working Notes/Labs/Workshop)*. Ed. by Pamela Forner, Jussi Karlgren, and Christa Womser-Hacker. ISBN: 978-88-904810-3-1. URL: http://dblp.uni-trier.de/db/conf/clef/clef2012w.html#AroraGBMB12 (cit. on p. 4).

Arun, C. H., W. R. Sam Emmanuel, and D. Christopher Durairaj (2013). “Texture Feature Extraction for Identification of Medicinal Plants and Comparison of Different Classifiers”. In: *International Journal of Computer Applications* 62.12 (cit. on pp. 4, 5, 23, 102).

Bebber, Daniel P. et al. (2010). “Herbaria are a major frontier for species discovery”. In: *Proceedings of the National Academy of Sciences* 107.51, pp. 22169–22171. ISSN: 0027-8424. DOI: 10.1073/pnas.1011841108. eprint: http://www.pnas.org/content/107/51/22169.full.pdf. URL: http://www.pnas.org/content/107/51/22169 (cit. on pp. 75, 93, 114).

Beghin, Thibaut, JamesS Cope, Paolo Remagnino, and Sarah Barman (2010). “Shape and Texture Based Plant Leaf Classification”. In: *Advanced Concepts for Intelligent Vision Systems*. Ed. by Jacques Blanc-Talon, Don Bone, Wilfried Philips, Dan Popescu, and Paul Scheunders. Vol. 6475. Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 345–353. ISBN: 978-3-642-17690-6. DOI: 10.1007/978-3-642-17691-3_32. URL: http://dx.doi.org/10.1007/978-3-642-17691-3_32 (cit. on pp. 3–5, 23).

Bhardwaj, Anant, Manpreet Kaur, and Anupam Kumar (2013). “Recognition of plants by Leaf Image using Moment Invariant and Texture Analysis”. In: *International Journal of Innovation and Applied Studies* 3.1 (1), pp. 237–248. ISSN: 2028-9324. URL: http://www.ijias.issr-journals.org/abstract.php?article=IJIAS-13-087-01 (cit. on pp. 4, 5, 23, 68, 69, 71, 102).

Bishop, Christopher M. (1995). *Neural Networks for Pattern Recognition*. New York, NY, USA: Oxford University Press, Inc. ISBN: 0198538642 (cit. on p. 133).

Bonnet, Pierre, Alexis Joly, Hervé Goëau, Julien Champ, Christel Vignau, Jean-François Molino, Daniel Barthélémy, and Nozha Boujemaa (2015). “Plant identification: man vs. machine”. In: *Mul-*

timedia Tools and Applications, pp. 1–19. ISSN: 1573-7721. DOI: 10.1007/s11042-015-2607-4 (cit. on p. 7).

Bookstein, Fred L. (1997). *Morphometric Tools for Landmark Data: Geometry and Biology*. Cambridge (cit. on p. 67).

Bradski, G. (2000). "The OpenCV Library". In: *Dr. Dobb's Journal of Software Tools* (cit. on p. 25).

Canziani, Alfredo, Adam Paszke, and Eugenio Culurciello (2016). "An Analysis of Deep Neural Network Models for Practical Applications". In: *CoRR* abs/1605.07678. URL: http://arxiv.org/abs/1605.07678 (cit. on pp. 6, 7).

Carranza-Rojas, Jose (2014). "A Texture and Curvature Bimodal Leaf Recognition Model for Costa Rican Plant Species Identification". MA thesis. Cartago, Costa Rica: Costa Rica Institute of Technology. URL: http://hdl.handle.net/2238/3913#sthash.dxxgH0FI.dpuf (cit. on p. 24).

Carranza-Rojas, Jose, Alexis Joly, Pierre Bonnet, Hervé Goëau, and Erick Mata-Montero (2017a). "Automated Herbarium Specimen Identification using Deep Learning". In: *Biodiversity Information Science and Standards* 1, e20302. DOI: 10.3897/tdwgproceedings.1.20302. eprint: https://doi.org/10.3897/tdwgproceedings.1.20302 (cit. on pp. 19, 93, 115, 116, 130, 146, 149, 162).

Carranza-Rojas, Jose, Alexis Joly, Hervé Goëau, Erick Mata-Montero, and Pierre Bonnet (2018). "Automated Identification of Herbarium Specimens at Different Taxonomic Levels". In: *Multimedia Tools and Applications for Environmental & Biodiversity Informatics*. Ed. by Alexis Joly, Stefanos Vrochidis, Kostas Karatzas, Ari Karppinen, and Pierre Bonnet. Cham: Springer International Publishing, pp. 151–167. ISBN: 978-3-319-76445-0. DOI: 10.1007/978-3-319-76445-0_9. URL: https://doi.org/10.1007/978-3-319-76445-0_9 (cit. on pp. 2, 113, 129, 130, 163).

Carranza-Rojas, Jose and Erick Mata-Montero (2016a). "Combining Leaf Shape and Texture for Costa Rican Plant Species Identification". en. In: *CLEI Electronic Journal* 19, pp. 7 –7. ISSN: 0717-5000. URL: http://www.scielo.edu.uy/scielo.php?script=sci_arttext&pid=S0717-50002016000100007&nrm=iso (cit. on pp. 22, 57, 58, 94, 106, 162).

Carranza-Rojas, Jose, Herve Goeau, Pierre Bonnet, Erick Mata-Montero, and Alexis Joly (2017b). "Going deeper in the automated identification of Herbarium specimens". In: *BMC Evolutionary Biology* 17.1, p. 181. ISSN: 1471-2148. DOI: 10.1186/s12862-017-1014-z. URL: https://doi.org/10.1186/s12862-017-1014-z (cit. on pp. 19, 74, 93, 94, 110, 114, 115, 118, 131, 149, 162).

Carranza-rojas, Jose, Erick Mata-Montero, and Herve Goeau (2018). "Hidden Biases in Automated Image-Based Plant Identification". In: *2018 IEEE International Work Conference on Bioinspired Intelligence (IWOBI)*, pp. 1–9. DOI: 10.1109/IWOBI.2018.8464187 (cit. on pp. 101, 163).

Carranza-Rojas, Jose and Erick Mata-Montero (2016b). "On the significance of leaf sides in automatic leaf-based plant species identification". In: *2016 IEEE 36th Central American and Panama Convention (CONCAPAN XXXVI)*, pp. 1–6. DOI: 10.1109/CONCAPAN.2016.7942341 (cit. on pp. 56, 145, 162).

Carranza-Rojas, Jose, Alexis Joly, Herve Goeau, and Erick Mata-Montero (2018). "Taxonomy-Softmax: A Hierarchical Loss Function for Deep Automatic Plant Identification". unpublished (cit. on p. 163).

Carvalho, MarceloR. de et al. (2007). "Taxonomic Impediment or Impediment to Taxonomy? A Commentary on Systematics and the Cybertaxonomic-Automation Paradigm". English. In: *Evolutionary Biology* 34.3-4, pp. 140–143. ISSN: 0071-3260. DOI: 10.1007/s11692-007-9011-6 (cit. on pp. 23, 102).

Casanova, Dalcimar, Jarbas Joaci de Mesquita Sa Junior, and Odemir Martinez Bruno (2009). "Plant leaf identification using Gabor wavelets". In: *International Journal of Imaging Systems and Technology* 19.3, pp. 236–243 (cit. on p. 76).

Cerutti, Guillaume, Laure Tougne, Julien Mille, Antoine Vacavant, and Didier Coquin (2013). "Understanding leaves in natural images–a model-based approach for tree species identification". In: *Computer Vision and Image Understanding* 117.10, pp. 1482–1501 (cit. on pp. 70, 76).

Choi, Sungbin (2015). "Plant Identification with Deep Convolutional Neural Network: SNUMedinfo at LifeCLEF Plant Identification Task 2015". In: *CLEF* (cit. on pp. 5, 6, 8).

Clarke, James, Sarah Barman, Paolo Remagnino, Ken Bailey, Don Kirkup, Simon Mayo, and Paul Wilkin (2006). "Venation Pattern Analysis of Leaf Images". In: *Proceedings of the Second International Conference on Advances in Visual Computing - Volume Part II*. ISVC'06. Lake Tahoe, NV: Springer-Verlag, pp. 427–436. ISBN: 3-540-48626-7, 978-3-540-48626-8. DOI: 10.1007/11919629_44. URL: http://dx.doi.org/10.1007/11919629_44 (cit. on p. 23).

Coelho, Luis Pedro (2013). "Mahotas: Open source software for scriptable computer vision". In: *Journal of Open Research Software* 1. DOI: 10.5334/jors.ac. URL: http://dx.doi.org/10.5334/jors.ac (cit. on pp. 33, 35).

Corney, David, Jonathan Y Clark, H Lilian Tang, and Paul Wilkin (2012). "Automatic extraction of leaf characters from herbarium specimens". In: *Taxon* 61.1, pp. 231–244 (cit. on p. 76).

Cun, Y. Le, B. Boser, J. S. Denker, R. E. Howard, W. Habbard, L. D. Jackel, and D. Henderson (1990). "Advances in Neural Information Processing Systems 2". In: ed. by David S. Touretzky. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. Chap. Handwritten Digit Recognition with a Back-propagation Network, pp. 396–404. ISBN: 1-55860-100-7. URL: http://dl.acm.org/citation.cfm?id=109230.109279 (cit. on p. 5).

Dempster, A. P., N. M. Laird, and D. B. Rubin (1977). "Maximum likelihood from incomplete data via the EM algorithm". In: *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B* 39.1, pp. 1–38 (cit. on p. 23).

Dieleman, Sander, Jan Schlüter, Colin Raffel, Eben Olson, Søren Kaae Sønderby, Daniel Nouri, et al. (2015). *Lasagne: First release.* DOI: 10.5281/zenodo.27878. URL: http://dx.doi.org/10.5281/zenodo.27878 (cit. on p. 118).

Duckworth, W Donald, Hugh H Genoways, and Carolyn L Rose (1993). "Preserving natural science collections: chronicle of our environmental heritage". In: (cit. on p. 75).

Ebach, Malte C and Marcelo R de Carvalho (2010). "Anti-intellectualism in the DNA barcoding enterprise". In: *Zoologia (Curitiba)* 27, 165–178 (cit. on p. 68).

Ellwood, Elizabeth R et al. (2015). "Accelerating the digitization of biodiversity research specimens through online public participation". In: *BioScience*, biv005 (cit. on p. 75).

Gaston, Kevin J and Mark A O'Neill (2004). "Automated species identification: why not?" In: *Philosophical Transactions of the Royal Society of London B: Biological Sciences* 359.1444, pp. 655–667 (cit. on p. 76).

Goëau, Hervé, Pierre Bonnet, and Alexis Joly (2015). "LifeCLEF Plant Identification Task 2015". In: *CLEF: Conference and Labs of the Evaluation forum*. Ed. by CEUR-WS. Vol. 1391. CLEF2015 Working notes. Toulouse, France. URL: https://hal.inria.fr/hal-01182795 (cit. on pp. 1, 3, 7, 13, 17, 77, 81, 89, 102, 107, 110, 115, 131).

Goeau, Herve, Pierre Bonnet, and Alexis Joly (2017). "Plant identification based on noisy web data: the amazing performance of deep learning (LifeCLEF 2017)". In: *CLEF 2017 - Conference and Labs of the Evaluation Forum*. Dublin, Ireland, pp. 1–13. URL: https://hal.archives-ouvertes.fr/hal-01629183 (cit. on pp. 8, 129, 131).

Goëau, Hervé, Pierre Bonnet, and Alexis Joly (2016). “Plant Identification in an Open-world (LifeCLEF 2016)”. In: *Working Notes of CLEF 2016 - Conference and Labs of the Evaluation forum, Évora, Portugal, 5-8 September, 2016.* Pp. 428–439. URL: http://ceur-ws.org/Vol-1609/16090428.pdf (cit. on pp. 8, 77, 102, 104, 107, 110).

Goëau, Hervé, Alexis Joly, Pierre Bonnet, Vera Bakic, Daniel Barthélémy, Nozha Boujemaa, and Jean-François Molino (2013). “The imageCLEF plant identification task 2013”. In: *Proceedings of the 2nd ACM international workshop on Multimedia analysis for ecological data*. ACM, pp. 23–28 (cit. on p. 76).

Goo, Wonjoon, Juyong Kim, Gunhee Kim, and Sung Ju Hwang (2016). “Taxonomy-Regularized Semantic Deep Convolutional Neural Networks”. In: *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II*, pp. 86–101. DOI: 10.1007/978-3-319-46475-6_6. URL: https://doi.org/10.1007/978-3-319-46475-6_6 (cit. on p. 9).

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville (2016). *Deep Learning*. http://www.deeplearningbook.org. MIT Press (cit. on pp. 3, 8, 9, 12, 75, 77, 107).

Goodfellow, Ian J., Yaroslav Bulatov, Julian Ibarz, Sacha Arnoud, and Vinay Shet (2014). “Multi-digit Number Recognition from Street View Imagery using Deep Convolutional Neural Networks”. In: URL: https://arxiv.org/pdf/1312.6082.pdf (cit. on pp. 9, 119, 120).

Grinblat, Guillermo L., Lucas C. Uzal, Mónica G. Larese, and Pablo M. Granitto (2016). “Deep learning for plant identification using vein morphological patterns”. In: *Computers and Electronics in Agriculture* 127, pp. 418 –424. ISSN: 0168-1699. DOI: http://dx.doi.org/10.1016/j.compag.2016.07.003. URL: http://www.sciencedirect.com/science/article/pii/S0168169916304665 (cit. on pp. 7, 104).

Hang, Siang Thye, Atsushi Tatsuma, and Masaki Aono (2016). “Bluefield (KDE TUT) at LifeCLEF 2016 Plant Identification Task”. In: *Working Notes of CLEF 2016 - Conference and Labs of the Evaluation forum, Évora, Portugal, 5-8 September, 2016.* Pp. 459–468. URL: http://ceur-ws.org/Vol-1609/16090459.pdf (cit. on pp. 8, 104).

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2016). “Deep Residual Learning for Image Recognition”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778 (cit. on pp. 20, 131).

— (2015). “Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification”. In: *CoRR* abs/1502.01852. URL: http://arxiv.org/abs/1502.01852 (cit. on pp. 8, 77, 107).

— (2014). “Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition”. In: *CoRR* abs/1406.4729. URL: http://arxiv.org/abs/1406.4729 (cit. on p. 8).

Hebert, Paul D. N., Alina Cywinska, Shelley L. Ball, and Jeremy R. Dewaard (2003). “Biological identifications through DNA barcodes”. In: *Proc. Biol. Sci* 270, pp. 313–321 (cit. on pp. 68, 102).

Herdiyeni, Y. and M.M. Santoni (2012). “Combination of morphological, Local Binary Pattern Variance and color moments features for Indonesian medicinal plants identification”. In: *Advanced Computer Science and Information Systems (ICACSIS), 2012 International Conference*, pp. 255–259 (cit. on pp. 4, 5, 23, 68, 69, 71, 102).

Herdiyeni, Y. and I. Kusmana (2013). “Fusion of Local Binary Patterns features for tropical medicinal plants identification”. In: *Advanced Computer Science and Information Systems (ICACSIS), 2013 International Conference on*, pp. 353–357. DOI: 10.1109/ICACSIS.2013.6761601 (cit. on pp. 5, 23, 58, 70, 72, 104).

iDigBio (2017). URL: https://www.idigbio.org/ (visited on 06/01/2017) (cit. on pp. 19, 75, 81, 131).

Ioffe, Sergey and Christian Szegedy (2015). “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”. In: *CoRR* abs/1502.03167. URL: http://arxiv.org/abs/1502.03167 (cit. on pp. 77, 107, 118).

J., Chaki . and Parekh. R. (2012). “Designing an Automated System for Plant Leaf Recognition”. In: *International Journal of Advances in Engineering and Technology (IJAET)*, pp. 149–158 (cit. on p. 4).

James, Shelley A., Pamela S. Soltis, Lee Belbin, Arthur D. Chapman, Gil Nelson, Deborah L. Paul, and Matthew Collins. “Herbarium data: Global biodiversity and societal botanical needs for novel research”. In: *Applications in Plant Sciences* 6.2, e1024. DOI: 10.1002/aps3.1024. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/aps3.1024. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/aps3.1024 (cit. on p. 147).

Jia, Yangqing, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell (2014). “Caffe: Convolutional Architecture for Fast Feature Embedding”. In: *Proceedings of the 22Nd ACM International Conference on Multimedia*. MM ’14. Orlando, Florida, USA: ACM, pp. 675–678. ISBN: 978-1-4503-3063-3. DOI: 10.1145/2647868.2654889. URL: http://doi.acm.org/10.1145/2647868.2654889 (cit. on pp. 21, 77, 107).

Joly, Alexis et al. (2016a). “A look inside the Pl@ntNet experience”. In: *Multimedia Systems* 22.6, pp. 751–766 (cit. on pp. 76, 102, 114, 129).

Joly, Alexis, Hervé Goëau, Julien Champ, Samuel Dufour-Kowalski, Henning Müller, and Pierre Bonnet (2016b). “Crowdsourcing biodiversity monitoring: how sharing your photo stream can sustain our planet”. In: *Proceedings of the 2016 ACM on Multimedia Conference*. ACM, pp. 958–967 (cit. on pp. 77, 89).

Joly, Alexis et al. (2014a). “Interactive plant identification based on social image data”. In: *Ecological Informatics* 23. Special Issue on Multimedia in Ecology and Environment, pp. 22 –34. ISSN: 1574-9541. DOI: http://dx.doi.org/10.1016/j.ecoinf.2013.07.006. URL: http://www.sciencedirect.com/science/article/pii/S157495411300071X (cit. on pp. 1, 7, 76, 104).

Joly, Alexis et al. (2014b). “LifeCLEF 2014: Multimedia Life Species Identification Challenges”. In: *Information Access Evaluation. Multilinguality, Multimodality, and Interaction: 5th International Conference of the CLEF Initiative, CLEF 2014, Sheffield, UK, September 15-18, 2014. Proceedings*. Ed. by Evangelos Kanoulas, Mihai Lupu, Paul Clough, Mark Sanderson, Mark Hall, Allan Hanbury, and Elaine Toms. Cham: Springer International Publishing, pp. 229–249. ISBN: 978-3-319-11382-1. DOI: 10.1007/978-3-319-11382-1_20 (cit. on pp. 3, 7, 13).

Joly, Alexis et al. (2015a). “LifeCLEF 2015: Multimedia Life Species Identification Challenges”. In: *Experimental IR Meets Multilinguality, Multimodality, and Interaction - 6th International Conference of the CLEF Association, CLEF 2015, Toulouse, France, September 8-11, 2015, Proceedings*, pp. 462–483 (cit. on pp. 5, 68, 69, 72, 76, 94).

Joly, Alexis et al. (2015b). “LifeCLEF 2015: multimedia life species identification challenges”. In: *Experimental IR Meets Multilinguality, Multimodality, and Interaction*. Springer, pp. 462–483 (cit. on p. 5).

Joly, Alexis et al. (2016c). “LifeCLEF 2016: Multimedia Life Species Identification Challenges”. In: *Experimental IR Meets Multilinguality, Multimodality, and Interaction: 7th International Conference of the CLEF Association, CLEF 2016, Évora, Portugal, September 5-8, 2016, Proceedings*. Ed. by Norbert Fuhr, Paulo Quaresma, Teresa Gonçalves, Birger Larsen, Krisztian Balog, Craig Macdonald, Linda Cappellato, and Nicola Ferro. Cham: Springer International Publishing, pp. 286–310. ISBN: 978-3-319-44564-9. DOI: 10.1007/978-3-319-44564-9_26 (cit. on pp. 114, 129).

Kadir, Abdul, Lukito Edi Nugroho, Adhi Susanto, and Paulus Insap Santosa (2011). "Leaf Classification Using Shape, Color, and Texture Features". In: *International Journal of Computer Trends and Technology* (cit. on pp. 4, 5, 23, 39).

Katole, Atul Laxman, Krishna Prasad Yellapragada, Amish Kumar Bedi, Sehaj Singh Kalra, and Mynepalli Siva Chaitanya (2015). "Hierarchical Deep Learning Architecture For 10K Objects Classification." In: *CoRR* abs/1509.01951. URL: http://dblp.uni-trier.de/db/journals/corr/corr1509.html#KatoleYBKC15 (cit. on p. 10).

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton (2012). "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems 25*. Ed. by P. Bartlett, F.c.n. Pereira, C.j.c. Burges, L. Bottou, and K.q. Weinberger, pp. 1106–1114 (cit. on pp. 1, 5, 7, 72, 73, 77, 104, 107).

Kumar, Neeraj, PeterN. Belhumeur, Arijit Biswas, DavidW. Jacobs, W.John Kress, IdaC. Lopez, and JoãoV.B. Soares (2012). "Leafsnap: A Computer Vision System for Automatic Plant Species Identification". In: *Computer Vision – ECCV 2012*. Ed. by Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid. Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 502–516. ISBN: 978-3-642-33708-6. DOI: 10.1007/978-3-642-33709-3_36. URL: http://dx.doi.org/10.1007/978-3-642-33709-3_36 (cit. on pp. 1, 3–5, 13, 23, 25, 27–31, 33, 35, 39, 57–59, 62, 68–71, 76, 102, 104, 106, 114).

Kumar, Prof Meeta, Mrunali Kamble, Shubhada Pawar, Prajakta Patil, and Neha Bonde. *Survey on Techniques for Plant Leaf Classification* (cit. on p. 69).

Larese, Mónica G., Rafael Namías, Roque M. Craviotto, Miriam R. Arango, Carina Gallo, and Pablo M. Granitto (2014). "Automatic Classification of Legumes Using Leaf Vein Image Features". In: *Pattern Recogn.* 47.1, pp. 158–168. ISSN: 0031-3203. DOI: 10.1016/j.patcog.2013.06.012. URL: http://dx.doi.org/10.1016/j.patcog.2013.06.012 (cit. on pp. 4, 7, 23).

Lasseck, Mario (2017). "Image-based Plant Species Identification with Deep Convolutional Neural Networks". In: *Working Notes of CLEF 2017 - Conference and Labs of the Evaluation Forum, Dublin, Ireland, September 11-14, 2017.* URL: http://ceur-ws.org/Vol-1866/paper_174.pdf (cit. on p. 8).

Le, Thi-Lan, Nam-Duong Duong, Van-Toi Nguyen, Hai Vu, Van-Nam Hoang, and Thi Thanh-Nhan Nguyen (2015). "Complex Background Leaf-based Plant Identification Method Based on Interactive Segmentation and Kernel Descriptor". In: *Proceedings of the 2Nd International Workshop on Environmental Multimedia Retrieval*. EMR '15. Shanghai, China: ACM, pp. 3–8. ISBN: 978-1-4503-3558-4. DOI: 10.1145/2764873.2764877 (cit. on p. 70).

LeCun, Yann, Yoshua Bengio, et al. (1995). "Convolutional networks for images, speech, and time series". In: *The handbook of brain theory and neural networks* 3361.10, p. 1995 (cit. on pp. 77, 107).

Lee, Kue-Bum, Kwang-Woo Chung, and Kwang-Seok Hong (2013a). "An Implementation of Leaf Recognition System Based on Leaf Contour and Centroid for Plant Classification". English. In: *Ubiquitous Information Technologies and Applications*. Ed. by Youn-Hee Han, Doo-Soon Park, Weijia Jia, and Sang-Soo Yeo. Vol. 214. Lecture Notes in Electrical Engineering. Springer Netherlands, pp. 109–116. ISBN: 978-94-007-5856-8. DOI: 10.1007/978-94-007-5857-5_12. URL: http://dx.doi.org/10.1007/978-94-007-5857-5_12 (cit. on pp. 4, 23, 39).

Lee, Kue-Bum and Kwang-Seok Hong (2013b). “An Implementation of Leaf Recognition System using Leaf Vein and Shape”. In: *International Journal of Bio-Science and Bio-Technology* (2), pp. 57–66 (cit. on pp. 4, 23).

Lee, Sue Han, Chee Seng Chan, Paul Wilkin, and Paolo Remagnino (2015). “Deep-Plant: Plant Identification with convolutional neural networks”. In: *Image Processing (ICIP), 2015 IEEE International Conference on*. IEEE, pp. 452–456 (cit. on pp. 6, 72, 73, 76, 104).

Li, Yan, Zheru Chi, and D.D. Feng (2006). “Leaf Vein Extraction Using Independent Component Analysis”. In: *Systems, Man and Cybernetics, 2006. SMC ’06. IEEE International Conference on*. Vol. 5, pp. 3890–3894. DOI: 10.1109/ICSMC.2006.384738 (cit. on pp. 4, 23, 69, 71).

Lu, Hongfei, Wu Jiang, M Ghiassi, Sean Lee, and Mantri Nitin (2012). “Classification of Camellia (Theaceae) Species Using Leaf Architecture Variations and Pattern Recognition Techniques”. In: *PLoS ONE* 7.1. Ed. by Robert DeSalle, e29704. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0029704. URL: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3250490/ (cit. on p. 71).

Luong, Thang, Hieu Pham, and Christopher D. Manning (2015). “Effective Approaches to Attention-based Neural Machine Translation”. In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, pp. 1412–1421. DOI: 10.18653/v1/D15-1166. URL: http://www.aclweb.org/anthology/D15-1166 (cit. on pp. 140, 150).

M. Z. Rashad B.S.el-Desouky, Manal S .Khawasik (2011). “Plants Images Classification Based on Textural Features using Combined Classifier”. In: *International Journal of Computer Science and Information Technology* 3.4 (cit. on pp. 3–5, 23, 68, 69, 71, 102).

MacLeod, N. (2007). *Automated Taxon Identification in Systematics: Theory, Approaches and Applications*. Systematics Association Special Volumes. Taylor & Francis. ISBN: 9780849382055 (cit. on pp. 57, 66, 102).

Manay, S., D. Cremers, Byung-Woo Hong, A.J. Yezzi, and S. Soatto (2006). “Integral Invariants for Shape Matching”. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 28.10, pp. 1602–1618. ISSN: 0162-8828. URL: http://dx.doi.org/10.1109/TPAMI.2006.208 (cit. on p. 31).

Mata-Montero, Erick. and Jose. Carranza-Rojas (2015). “A texture and curvature bimodal leaf recognition model for identification of Costa Rican plant species”. In: *Computing Conference (CLEI), 2015 Latin American*, pp. 1–12. DOI: 10.1109/CLEI.2015.7360026 (cit. on pp. 1, 4, 5, 17, 69, 71, 72, 81, 102, 104, 106, 109, 110).

Mata-Montero, Erick and Jose Carranza-Rojas (2016). “Automated Plant Species Identification: Challenges and Opportunities”. In: *ICT for Promoting Human Development and Protecting the Environment: 6th IFIP World Information Technology Forum, WITFOR 2016, San José, Costa Rica, September 12-14, 2016, Proceedings*. Springer International Publishing, pp. 26–36. ISBN: 978-3-319-44447-5. DOI: 10.1007/978-3-319-44447-5_3 (cit. on pp. 1, 6, 13, 57, 65, 79, 102, 104, 105, 114, 129, 150, 162).

Mnih, Andriy and Geoffrey E Hinton (2009). “A Scalable Hierarchical Distributed Language Model”. In: *Advances in Neural Information Processing Systems 21*. Ed. by D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou. Curran Associates, Inc., pp. 1081–1088. URL: http://papers.nips.cc/paper/3583-a-scalable-hierarchical-distributed-language-model.pdf (cit. on pp. 12, 129, 130).

Mononen, Tero, Riitta Tegelberg, S Mira, Markku A Huttunen, T Marko, Hannu Saarenmaa, et al. (2014). “DigiWeb-a workflow environment for quality assurance of transcription in digitization of natural history collections”. In: *Biodiversity Informatics* 9.1 (cit. on p. 75).

Mouine, Sofiène, Itheri Yahiaoui, and Anne Verroust-Blondet (2013). “A shape-based approach for leaf classification using multiscaletriangular representation.” In: *ICMR*. Ed. by Ramesh Jain, Balakrishnan Prabhakaran, Marcel Worring, John R. Smith, and Tat-Seng Chua. ACM, pp. 127–134. ISBN: 978-1-4503-2033-7. URL: http://dblp.uni-trier.de/db/conf/mir/icmr2013.html#MouineYV13 (cit. on p. 39).

Nguyen, Q., T. Le, and N. Pham (2013). “Leaf based plant identification system for Android using SURF features in combination with Bag of Words model and supervised learning”. In: *International Conference on Advanced Technologies for Communications (ATC)* (cit. on pp. 4, 5, 24, 39, 57, 58, 71, 104).

Nicolson, Nicky, Katherine Challis, Allan Tucker, and Sandra Knapp (2017). “Impact of e-publication changes in the International Code of Nomenclature for algae, fungi and plants (Melbourne Code, 2012)-did we need to “run for our lives”?” In: *BMC evolutionary biology* 17.1, p. 116 (cit. on pp. 91, 147).

Ojala, T., M. Pietikainen, and T. Maenpaa (2002). “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns”. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 24.7, pp. 971–987. ISSN: 0162-8828. URL: http://dx.doi.org/10.1109/TPAMI.2002.1017623 (cit. on pp. 33, 59).

Oliphant, Travis E. (2006). *Guide to NumPy*. Provo, UT. URL: http://www.tramy.us/ (cit. on p. 25).

Page, Lawrence M, Bruce J MacFadden, Jose A Fortes, Pamela S Soltis, and Greg Riccardi (2015). “Digitization of biodiversity collections reveals biggest data on biodiversity”. In: *BioScience*, biv104 (cit. on pp. 75, 93).

Paszke, Adam et al. (2017). “Automatic differentiation in PyTorch”. In: (cit. on pp. 21, 131).

Pedregosa, F. et al. (2011). “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12, pp. 2825–2830 (cit. on p. 29).

Pietikäinen, Matti, Abdenour Hadid, Guoying Zhao, and Timo Ahonen (2011). “Local Binary Patterns for Still Images”. In: *Computer Vision Using Local Binary Patterns*. Vol. 40. Computational Imaging and Vision. Springer London, pp. 13–47. ISBN: 978-0-85729-747-1. DOI: 10.1007/978-0-85729-748-8_2. URL: http://dx.doi.org/10.1007/978-0-85729-748-8_2 (cit. on p. 4).

R.D, Lagerwall and Viriri S (2011). “Plant Classification using Lead Recognition”. In: *Proceedings of the 22nd Annual Symposium of the Pattern Recognition Association of South Africa*, 91–95 (cit. on pp. 4, 23, 39, 68, 69, 71).

Rubinoff, Daniel, Stephen Cameron, and Kipling Will (2006). “A Genomic Perspective on the Shortcomings of Mitochondrial DNA for “Barcoding” Identification”. In: *Journal of Heredity* 97.6, pp. 581–594 (cit. on p. 68).

Russakovsky, Olga et al. (2015). “ImageNet Large Scale Visual Recognition Challenge”. In: *International Journal of Computer Vision (IJCV)* 115.3, pp. 211–252. DOI: 10.1007/s11263-015-0816-y (cit. on pp. 5–7, 20, 81, 104, 116, 131).

Shahbaba, Babak and Radford M. Neal (2007). “Improving classification when a class hierarchy is available using a hierarchy-based prior”. In: *Bayesian Anal.* 2.1, pp. 221–237. DOI: 10.1214/07-BA209. URL: http://dx.doi.org/10.1214/07-BA209 (cit. on pp. 115, 129, 130).

Silla, Carlos N and Alex A Freitas (2011). “A survey of hierarchical classification across different application domains”. In: *Data Min Knowl Disc* 22, pp. 31–72. DOI: 10.1007/s10618-010-0175-9 (cit. on pp. 9–11, 115, 129, 130).

Simard, Patrice Y., Dave Steinkraus, and John C. Platt (2003). “Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis”. In: *Proceedings of the Seventh International Con-*

ference on Document Analysis and Recognition - Volume 2. ICDAR '03. Washington, DC, USA: IEEE Computer Society, pp. 958–. ISBN: 0-7695-1960-1 (cit. on pp. 64, 73).

Simonyan, Karen and Andrew Zisserman (2014). "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *CoRR* abs/1409.1556. URL: http://arxiv.org/abs/1409.1556 (cit. on pp. 6, 8, 104).

Soares, JoãoV.B. and DavidW. Jacobs (2013). "Efficient segmentation of leaves in semi-controlled conditions". English. In: *Machine Vision and Applications* 24.8, pp. 1623–1643. ISSN: 0932-8092. DOI: 10.1007/s00138-013-0530-0. URL: http://dx.doi.org/10.1007/s00138-013-0530-0 (cit. on pp. 4, 70).

Söderkvist, Oskar J O (2001). "Computer Vision Classification of Leaves from Swedish Trees". LiTH-ISY-EX-3132. MA thesis. SE-581 83 Linköping, Sweden: Linköping University (cit. on p. 4).

Suhrbier, Lutz, W-H Kusber, Okka Tschöpe, Anton Güntsch, and Walter G Berendsohn (2017). "AnnoSys—implementation of a generic annotation system for schema-based data using the example of biodiversity collection data". In: *Database* 2017.1, bax018 (cit. on p. 75).

Sun, Zhihui, Shenglian Lu, Xinyu Guo, and Yuan Tian (2011). "Leaf Vein and Contour Extraction from Point Cloud Data". In: pp. 11–16. DOI: 10.1109/ICVRV.2011.40 (cit. on p. 71).

Suzuki, Satoshi and KeiichiA be (1985). "Topological structural analysis of digitized binary images by border following". In: *Computer Vision, Graphics, and Image Processing* 30.1, pp. 32 –46. ISSN: 0734-189X (cit. on p. 31).

Szegedy, Christian, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich (2015). "Going deeper with convolutions". In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 07-12-June, pp. 1–9. ISSN: 10636919. DOI: 10.1109/CVPR.2015.7298594. arXiv: 1409.4842 (cit. on pp. 6, 20, 21, 77, 93, 107, 118).

Szegedy, Christian, Sergey Ioffe, and Vincent Vanhoucke (2016). "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning". In: *CoRR* abs/1602.07261. URL: http://arxiv.org/abs/1602.07261 (cit. on p. 6).

Theano Development Team (2016). "Theano: A Python framework for fast computation of mathematical expressions". In: *arXiv e-prints* abs/1605.02688. URL: http://arxiv.org/abs/1605.02688 (cit. on pp. 21, 118).

Thiers, B. (2017). *Index Herbariorum: A global directory of public herbaria and associated staff*. URL: http://sweetgum.nybg.org/science/ih/ (visited on 06/01/2017) (cit. on p. 75).

Thiers, Barbara M., Melissa C. Tulig, and Kimberly A. Watson (2016). "Digitization of The New York Botanical Garden Herbarium". In: *Brittonia* 68.3, pp. 324–333. ISSN: 1938-436X. DOI: 10.1007/s12228-016-9423-7. URL: http://dx.doi.org/10.1007/s12228-016-9423-7 (cit. on p. 76).

Tomaszewski, Dominik and Angelika Górzkowska (2016). "Is shape of a fresh and dried leaf the same?" In: *PloS one* 11.4, e0153071 (cit. on pp. 77, 89).

Tschöpe, Okka, James A Macklin, Robert A Morris, Lutz Suhrbier, and Walter G Berendsohn (2013). "Annotating biodiversity data via the Internet". In: *Taxon* 62.6, pp. 1248–1258 (cit. on p. 75).

Unger, Jakob, Dorit Merhof, and Susanne Renner (2016). "Computer vision applied to herbarium specimens of German trees: testing the future utility of the millions of herbarium specimen images for automated identification". In: *BMC Evolutionary Biology* 16.1, p. 248. ISSN: 1471-2148. DOI: 10.1186/s12862-016-0827-5. URL: http://dx.doi.org/10.1186/s12862-016-0827-5 (cit. on pp. 76, 87).

Vishakha Metre, Jayshree Ghorpade (2013). "An Overview of the Research on Texture Based Plant Leaf Classification". In: *CoRR* abs/1306.4345 (cit. on pp. 4, 69).

Wäldchen, Jana and Patrick Mäder (2017). “Plant Species Identification Using Computer Vision Techniques: A Systematic Literature Review”. In: *Archives of Computational Methods in Engineering*, pp. 1–37. ISSN: 1886-1784. DOI: 10.1007/s11831-016-9206-z. URL: http://dx.doi.org/10.1007/s11831-016-9206-z (cit. on pp. 76, 77, 104).

Wijesingha, D and Fmmt Marikar (2012). “Automatic Detection System for the Identification of Plants Using Herbarium Specimen Images”. In: *Tropical Agricultural Research* 23.1, pp. 42–50. ISSN: 1016-1422. DOI: 10.4038/tar.v23i1.4630. URL: http://www.sljol.info/index.php/TAR/article/view/4630papers3://publication/uuid/7718FB7B-AFFE-4E9B-82F6-E7B6BF30ADD4 (cit. on pp. 4, 23, 76, 93, 102).

Wilf, Peter, Shengping Zhang, Sharat Chikkerur, Stefan A. Little, Scott L. Wing, and Thomas Serre (2016). “Computer vision cracks the leaf code”. In: *Proceedings of the National Academy of Sciences* 113.12, pp. 3305–3310. DOI: 10.1073/pnas.1524473113. eprint: http://www.pnas.org/content/113/12/3305.full.pdf. URL: http://www.pnas.org/content/113/12/3305.abstract (cit. on p. 76).

Wu, Cinna, Mark Tygert, and Yann LeCun (2017). “Hierarchical loss for classification”. In: *CoRR* abs/1709.01062. arXiv: 1709.01062. URL: http://arxiv.org/abs/1709.01062 (cit. on p. 12).

Wu, Feihong, Jun Zhang, and Vasant Honavar (2005). “Learning Classifiers Using Hierarchically Structured Class Taxonomies”. In: *Proceedings of the 6th International Conference on Abstraction, Reformulation and Approximation*. SARA’05. Airth Castle, UK: Springer-Verlag, pp. 313–320. ISBN: 3-540-27872-9, 978-3-540-27872-6. DOI: 10.1007/11527862_24. URL: http://dx.doi.org/10.1007/11527862_24 (cit. on pp. 115, 132).

Wu, S.G., F.S. Bao, E.Y. Xu, Yu-Xuan Wang, Yi-Fan Chang, and Qiao-Liang Xiang (2007). “A Leaf Recognition Algorithm for Plant Classification Using Probabilistic Neural Network”. In: *Signal Processing and Information Technology, 2007 IEEE International Symposium on*, pp. 11–16. DOI: 10.1109/ISSPIT.2007.4458016 (cit. on pp. 1, 3–5, 23, 24, 39, 57, 68, 69, 71, 102, 104).

Wäldchen, Jana, Michael Rzanny, Marco Seeland, and Patrick Mäder (2018). “Automated plant species identification—Trends and future directions”. In: *PLOS Computational Biology* 14.4, pp. 1–19. DOI: 10.1371/journal.pcbi.1005993. URL: https://doi.org/10.1371/journal.pcbi.1005993 (cit. on p. 147).

Yan, Zhicheng, Hao Zhang, Robinson Piramuthu, Vignesh Jagadeesh, Dennis DeCoste, Wei Di, and Yizhou Yu (2015). “HD-CNN: Hierarchical Deep Convolutional Neural Network for Large Scale Visual Recognition”. In: *ICCV’15: Proc. IEEE 15th International Conf. on Computer Vision* (cit. on pp. 10, 115, 129, 130).

Yanikoglu, B., E. Aptoula, and C. Tirkaz (2014). “Automatic plant identification from photographs”. In: *Machine Vision and Applications* 25.6, pp. 1369–1383 (cit. on p. 76).

Yosinski, Jason, Jeff Clune, Yoshua Bengio, and Hod Lipson (2014). “How transferable are features in deep neural networks?” In: *Advances in neural information processing systems*, pp. 3320–3328 (cit. on pp. 6, 77, 79, 109).

Zamora, Nelson (2014). Private Communication, National Biodiversity Institute, Costa Rica. National Biodiversity Institute (cit. on pp. 23, 24).

Zhao, Cong, Sharon S.F. Chan, Wai-Kuen Cham, and L.M. Chu (2015). “Plant Identification Using Leaf shapes-A Pattern Counting Approach”. In: *Pattern Recogn.* 48.10, pp. 3203–3215. ISSN: 0031-3203. DOI: 10.1016/j.patcog.2015.04.004 (cit. on p. 70).

Zhu, J., T. Park, P. Isola, and A. A. Efros (2017). “Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks”. In: *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2242–2251. DOI: 10.1109/ICCV.2017.244 (cit. on p. 149).

Zhu, Xiatian, Chen Change Loy, and Shaogang Gong (2014). “Constructing Robust Affinity Graphs for Spectral Clustering”. In: *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pp. 1450–1457. URL: http://dx.doi.org/10.1109/CVPR.2014.188 (cit. on p. 26).

Chapter 13

Appendix

Publications

- Jose Carranza-Rojas and Erick Mata-Montero (2016a). “Combining Leaf Shape and Texture for Costa Rican Plant Species Identification”. en. In: *CLEI Electronic Journal* 19, pp. 7 –7. ISSN: 0717-5000. URL: http://www.scielo.edu.uy/scielo.php?script=sci_arttext&pid=S0717-50002016000100007&nrm=iso
- Erick Mata-Montero and Jose Carranza-Rojas (2016). “Automated Plant Species Identification: Challenges and Opportunities”. In: *ICT for Promoting Human Development and Protecting the Environment: 6th IFIP World Information Technology Forum, WITFOR 2016, San José, Costa Rica, September 12-14, 2016, Proceedings*. Springer International Publishing, pp. 26–36. ISBN: 978-3-319-44447-5. DOI: 10.1007/978-3-319-44447-5_3
- Jose Carranza-Rojas and Erick Mata-Montero (2016b). “On the significance of leaf sides in automatic leaf-based plant species identification”. In: *2016 IEEE 36th Central American and Panama Convention (CONCAPAN XXXVI)*, pp. 1–6. DOI: 10.1109/CONCAPAN.2016.7942341
- Jose Carranza-Rojas, Herve Goeau, Pierre Bonnet, Erick Mata-Montero, and Alexis Joly (2017b). “Going deeper in the automated identification of Herbarium specimens”. In: *BMC Evolutionary Biology* 17.1, p. 181. ISSN: 1471-2148. DOI: 10.1186/s12862-017-1014-z. URL: https://doi.org/10.1186/s12862-017-1014-z
- Jose Carranza-Rojas, Alexis Joly, Pierre Bonnet, Hervé Goëau, and Erick Mata-Montero (2017a). “Automated Herbarium Specimen Identification using Deep Learning”. In: *Biodiversity Information Science and Standards* 1, e20302. DOI: 10.

3897/tdwgproceedings.1.20302. eprint: https://doi.org/10.3897/tdwgproceedings.1.20302

- Jose Carranza-rojas, Erick Mata-Montero, and Herve Goeau (2018). “Hidden Biases in Automated Image-Based Plant Identification”. In: *2018 IEEE International Work Conference on Bioinspired Intelligence (IWOBI)*, pp. 1–9. DOI: 10.1109/IWOBI.2018.8464187
- Jose Carranza-Rojas, Alexis Joly, Hervé Goëau, Erick Mata-Montero, and Pierre Bonnet (2018). “Automated Identification of Herbarium Specimens at Different Taxonomic Levels”. In: *Multimedia Tools and Applications for Environmental & Biodiversity Informatics*. Ed. by Alexis Joly, Stefanos Vrochidis, Kostas Karatzas, Ari Karppinen, and Pierre Bonnet. Cham: Springer International Publishing, pp. 151–167. ISBN: 978-3-319-76445-0. DOI: 10.1007/978-3-319-76445-0_9. URL: https://doi.org/10.1007/978-3-319-76445-0_9
- Jose Carranza-Rojas, Alexis Joly, Herve Goeau, and Erick Mata-Montero (2018). “Taxonomy-Softmax: A Hierarchical Loss Function for Deep Automatic Plant Identification”. unpublished