Image and attribute based identification of *Protea* species

by

Peter Thompson

Thesis presented in partial fulfilment of the requirements for the degree of Master of Science (Applied Mathematics) in the Faculty of Science at Stellenbosch University

Supervisor: Prof W. Brink

March 2020

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: March 2020

Abstract

The flowering plant genus *Protea* is a dominant representative for the biodiversity of the Cape Floristic Region in South Africa, and from a conservation point of view important to monitor. The recent surge in popularity of crowd-sourced wildlife monitoring platforms presents opportunities for automatic image based identification, for improved monitoring of species. We consider the problem of identifying the *Protea* species in a given image with additional (but optional) attributes linked to the observation, such as location, elevation and date. We collect training and test data from a crowd-sourced platform, and find that the *Protea* identification problem is exacerbated by considerable inter-class similarity, data scarcity, class imbalance, as well as large variations in image quality, composition and background. Our proposed solution consists of three parts. The first part incorporates a variant of multi-region attention into a pretrained convolutional neural network, to focus on the flowerhead in the image. The second part performs coarser-grained classification on subgenera (superclasses) and then rescales the output of the first part. The third part conditions a probabilistic model on the additional attributes associated with the observation. We perform an ablation study on the proposed model and its constituents, and find that all three components together outperform our baselines and all other variants quite significantly.

Uittreksel

Die blommende plantgenus *Protea* is ’n dominante verteenwoordiger vir die biodiversiteit van die Kaapse Floristiese Streek in Suid-Afrika. Vir hierdie rede, en uit ’n bewaringsoogpunt, is dit dus belangrik om die genus te monitor. Die onlangse toename in gewildheid en gebruik van skare-gebaseerde moniteringplatforms vir die natuurlike omgewing, bied geleenthede vir outomatiese beeldgebaseerde spesie-identifikasie. Ons oorweeg die probleem om die *Protea* spesie in ’n gegewe beeld te identifiseer, met behulp van addisionele (maar opsionele) eienskappe wat aan die waarneming gekoppel is, soos plek en datum. Ons versamel afrigtings- en toetsdata vanaf ’n skare-gebaseerde platform en vind dat die *Protea* identifikasieprobleem vererger word deur aansienlike interklas-ooreenkomste, dataskaarste, wanbalans in die hoeveelheid data vir elke klas, asook groot variasies in beeldkwaliteit, samestelling en agtergrond. Ons voorgestelde oplossing bestaan uit drie dele. Die eerste deel inkorporeer ’n variant van multi-gebied aandag in ’n vooraf-afgerigte neurale netwerk, om op die blomkop in die beeld te fokus. Die tweede deel voer ’n growwer klassifikasie op subgenusse (superklasse) uit, en skaleer dan die resultate van die eerste deel. Die derde deel kondisioneer ’n waarskynlikheidsmodel met die addisionele eienskappe wat met die waarneming verband hou. Ons voer ’n kombinasie-studie uit oor die voorgestelde model en sy komponente, en vind dat die drie komponente saam, in die voorgestelde wyse, beter presteer as ons basis-modelle en alle ander kombinasies.

Acknowledgements

I would like to express my sincere gratitude to the following people who assisted and supported me throughout this project:

- My supervisor Willie Brink for his unwavering support and guidance.
- The Wilhelm Frank Trust for the funding of my MSc studies.
- All the *iNaturalists* who allowed me to use their data, many of whom are friends, with their names and images to be found throughout the thesis.
- Tony Rebelo for providing access to the Protea Atlas Project dataset.
- My lab partner Shane Josias for freely sharing his knowledge and ideas.
- Finally, my parents and Werner Theron for continued love and support.

Contents

1 Introduction **1**

2 Related work **5**
2.1 Plant identification 5
2.2 Fine-grained image recognition 10

3 Datasets **14**
3.1 iNaturalist 15
3.2 The Protea Atlas Project 18

4 Image identification **22**
4.1 Neural networks 22
4.2 CNNs 25
4.3 CNN with attention 29

5 Subgenus network and attributes model **33**
5.1 Subgenus network 33
5.2 Attributes model 35

6 Experimental results **37**
6.1 Performace measures 37
6.2 Image preparation 37
6.3 Baseline image identification 38
6.4 CNN with attention 39
6.5 Subgenus network 43
6.6 Attributes model 43
6.7 Full *Protea* identification model 44

7 Conclusion **51**

References **53**

Appendix **58**

— 1 —

Introduction

The iconic plant genus *Protea* has its centre of diversity in the Cape Floristic Region (CFR) of South Africa; a region that accounts for 40% of the country's 20,400 species of indigenous flowering plants [50] while covering only 4% of the country's area. The diversity of *Protea* makes it a fitting surrogate for the biodiversity of the region [13] and consequently an important genus to monitor for the sake of conservation.

This sentiment follows a global trend of plant identification for the sake of conservation efforts. Studies of endangered and alien plant species are also performed in order to determine the effects of climate change on native plant distributions.

The monitoring of biodiversity is traditionally performed by expert scientists, but there is a growing tendency to utilise the power of crowd-sourced data [9, 45]. Such data is becoming important for understanding species populations in the midst of issues such as global warming, pollution, poaching and loss of biodiversity [4, 56, 8].

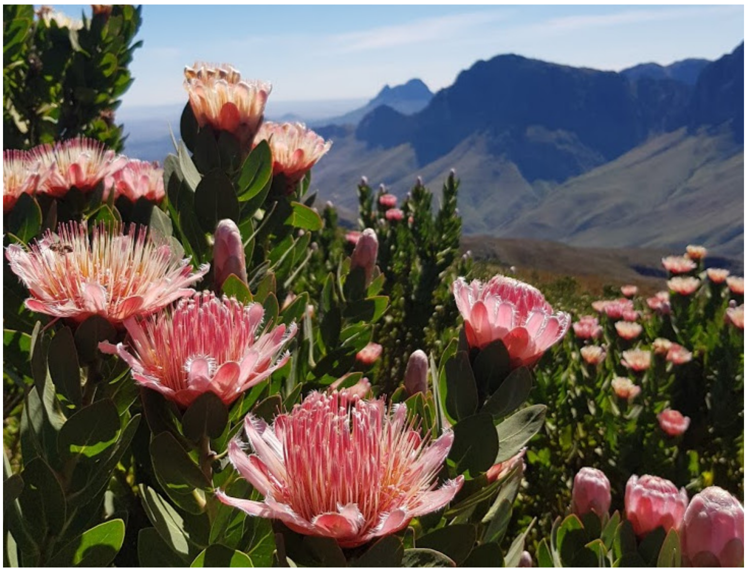
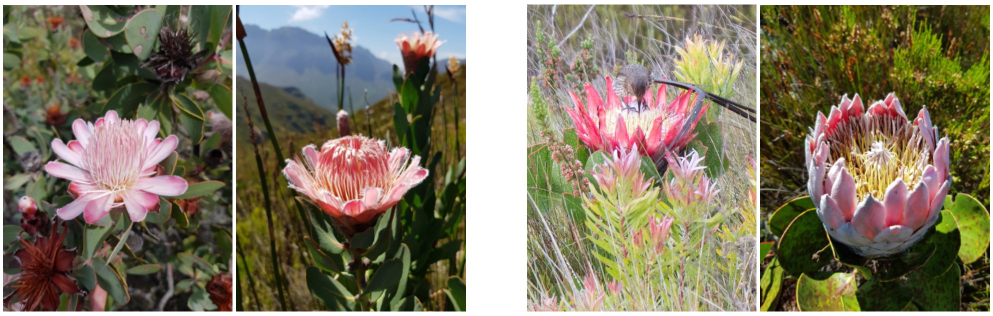


Figure 1.1: The endangered *Protea lacticolor* overlooks the town of Stellenbosch from the slopes of the Triplets in Jonkershoek. It is one of many native species threatened by anthropogenic activities.



(a) Interclass similarity

(b) Intraclass dissimilarity

Figure 1.2: (a) Different species can exhibit considerable visual similarity, such as *P. punctata* on the left and *P. lacticolor* on the right. (b) Different individuals of the same species may look markedly different, such as these two images of *P. cynaroides*.

The international crowd-sourced platform iNaturalist for example allows users to upload observations of wildlife, which typically include images, locations, dates, and identifications that can be verified by fellow users [51]. As of November 2019 the iNaturalist database contains over 28,000,000 observations for over 242,000 species, and it is impossible for experts to keep up with the sheer influx of data [54]. The additional decline in the number of experienced taxonomists [20] emphasises the urgency of efficient species identification techniques.

Automated tools based on computer vision may ease the task of identification, and could potentially provide expert-like knowledge to amateur naturalists. iNaturalist implements a top-k recommender system built on deep convolutional models for image identification [51], but challenges due to large class imbalances and fine granularity in biological domains remain [4, 57]. Class imbalance refers to the large discrepancy between the number of observations per species, while fine granularity refers to the visual similarities between separate species.

In this thesis we focus on the problem of automatically identifying *Protea* species from images, as a surrogate both for the biodiversity of the CFR and for the unbalanced and fine-grained databases of citizen science projects in general.

The problem is complicated by a number of factors. Firstly, it is a fine-grained classification problem where some species share striking visual similarities with others, as demonstrated in Figure 1.2. This fine granularity is complicated by the inherently small interclass variability, and a relatively large intraclass variability within the genus. This means that for two images from the same class, there may be large visual differences, while two images from separate classes may look remarkably similar.

Secondly, image data is extremely scarce for many of the rarer species. When we constructed our dataset (as detailed in Chapter 3), only 41 of the 70 *Protea* species known to exist in the CFR had at least 20 different images depicting an inflorescence (flowerhead). Unfortunately the prevalence of hybrid cultivars makes it infeasible to scrape the Internet for additional images, labelled or otherwise. The implication of this is that the distribution of data over species is unbalanced, as Figure 1.3

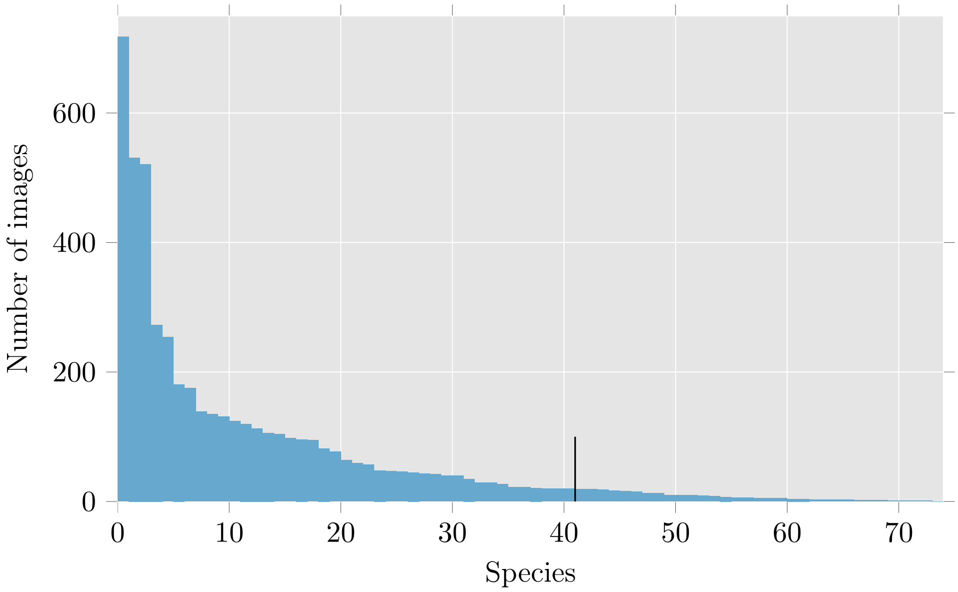


Figure 1.3: The number of available images per species indicates the degree of class imbalance and long tail in our dataset. The vertical line at species ID 41 indicates the cutoff for species we considered. All species to the right of this line have fewer than 20 images available.

indicates. Four of the 41 species mentioned above account for over 40% of the data, with less-frequent classes forming a long tail.

Finally, the image data is sourced from populations in the wild, by many different observers. There is no standard in how images were taken, resulting in large amounts of compositional and background variation.

In order to address these challenges we restrict the problem to the 41 species for which at least 20 images could be found, and propose an automated identification model that consists of three components, as summarised in Figure 1.4. The first component is a convolutional neural network (CNN) with a variant of multi-region attention [62], trained to perform classification over the 41 species. The second component leverages the fact that *Protea* species can be categorised into more easily distinguishable subgenera (the two species in Figure 1.2 (a) are both White Sugarbushes, for example) and accordingly consists of a CNN trained for subgenus classification. Its output is used essentially to rescale the class scores of the first network. The occurrence of *Protea* species tends to be relatively finely dependent on location and elevation, and different species also flower during different times of the year. Such attributes are often available as part of an observation, and the third component of our model exploits such additional data (if available) through a simple Bayesian approach.

The rest of the thesis is structured as follows. In Chapter 2 we discuss the literature of automated plant identification and fine-grained image classification. In Chapter 3 we introduce our fine-grained image dataset scraped from iNaturalist, and we discuss how we construct the necessary distributions for our attributes model. In Chapter 4 the technicalities of neural networks, convolutional neural networks and the proposed

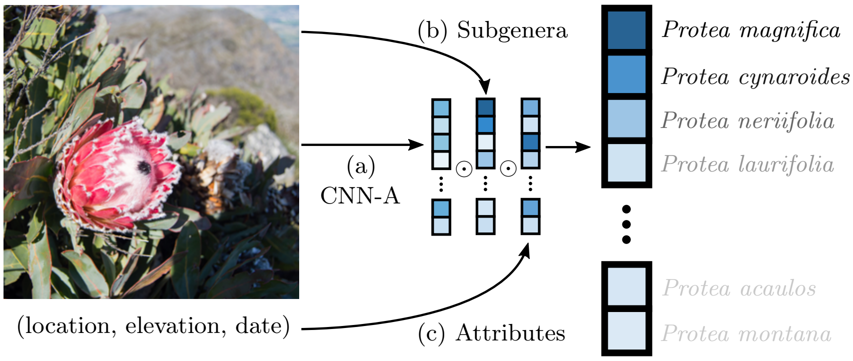


Figure 1.4: Our model for *Protea* species identification operates on an image with additional (but optional) attributes linked to the observation, and combines three parts: (a) a CNN with attention, (b) a separate network that classifies the image into coarser subgenera, and (c) a probabilistic model conditioned on the attributes.

attention model are presented. In Chapter 5 we examine both the subgenus network and the attributes model, which lead to the combined model as outlined in Figure 1.4. In Chapter 6 we compare the performance of various versions of our models. An ablation study on the proposed model suggests that all three components together outperform the baselines substantially in terms of test accuracy and recall. Finally the thesis is concluded in Chapter 7 and an appendix is included with information on *Protea* subgenera, as well as the credits for images and photographs used throughout the thesis.

Significant contributions made in this thesis can be summarised as follows. We introduce a challenging new dataset for fine-grained image classification. We collected and manually verified 4,849 images of 41 different *Protea* species with location, elevation and date information. We propose an identification model that consists of a CNN with attention, a second CNN to classify on the coarser subgenera-level and rescale the output of the first CNN, and a probabilistic model to condition the identification on the observed attributes. The performance of the proposed model is promising, and its various elements can be used separately or jointly to solve similar problems. A paper on the main findings of this work has been accepted for publication in the proceedings of the IEEE Winter Conference on Applications of Computer Vision 2020.

— 2 —

Related work

In this chapter we explore various forms of image based plant identification that uses machine learning and statistical techniques. We discuss classical methods that rely on feature engineering, as well as more recent advances in deep learning, which have opened a wealth of opportunities for fine-grained species identification [6]. The power of deep learning techniques extend into application, allowing members of the public to identify plant species automatically from photos.

We investigate fine-grained image recognition, as well as the standard datasets that are typically used to assess models. This includes methods that rely on attention mechanisms, as well as methods that incorporate non-visual attributes.

Since our task of *Protea* identification may be seen as a biological classification problem and as a fine-grained image classification problem, we consider related work that deals with both of these issues (combined or separately).

2.1 Plant identification

Approaches to address the problem of automated plant identification have evolved over many years, to reach the methods based on deep learning that are common today. We consider literature that assumes the availability of annotated training data, which naturally leads to supervised classification problems. We outline classical machine learning methods that tackle the problem using feature engineering, as well as the state-of-the-art deep learning methods. Further, we mention a number of benchmark datasets that are typically used to assess plant identification models. We consider datasets of plant images in controlled settings, as well as of images in natural settings (images of specimens in the wild). It is important to keep in mind that image classification of different plant species is inherently fine-grained, which can complicate the task of classification.

In Figure 2.1 we outline the general structure of a supervised machine learning model for plant identification.

2.1.1 Feature engineering

Images are typically composed of millions of pixels, in three colour channels. A machine learning model tasked to differentiate between plant species would need to be

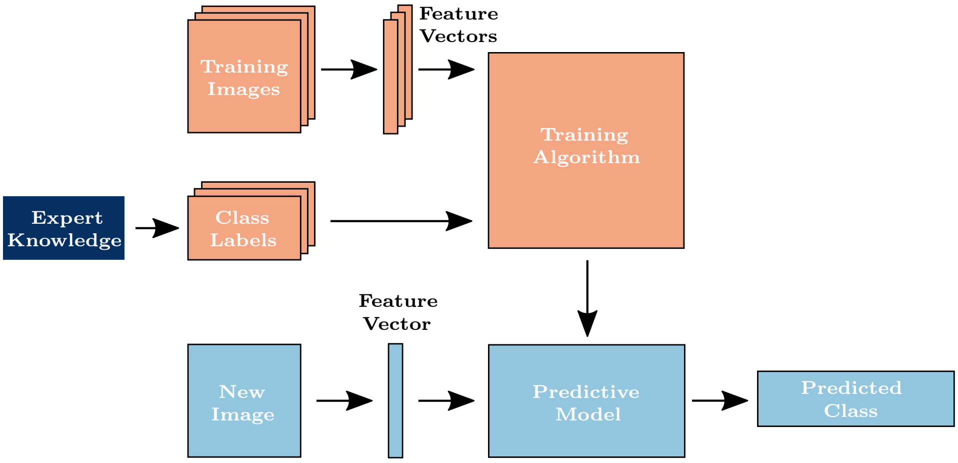


Figure 2.1: The steps of a supervised machine learning model for plant identification from images. The orange blocks refer to the training phase of the model, while the light blue blocks refer to the application phase. Image recreated from [54].

able to process and extract information from these high-dimensional inputs. Classical algorithms customarily create feature vectors from the input images [53], which help to reduce the dimensionality of the input while also distilling the information crucial for identification. Before the rise of deep neural networks, it was customary to perform feature engineering to aid the task of plant classification. This is a notoriously labour-intensive process and requires human input, with the added drawback of being domain specific [35].

Wäldchen et al. [53] review plant species identification on the basis of plant organs such as leaves, flowers, fruit, stems, as well as the whole plant. There are multiple examples of feature engineering for plant identification from leaves [12, 26, 58, 59], which all extract shape features from images. The images used in these studies are largely photographed on plain backgrounds, which makes comparison across features much easier. For leaves, flowers, or any plant organs photographed in a natural setting, the background uniformity is lost. This means that the lighting, composition, background clutter etc. are different across the images, which complicates the task of automated identification. We restrict this section to those studies focused solely on the identification of plants from images of the flowers.

Literature preceding the advent of deep neural networks for flower identification is not as dense as the literature for leaf based identification [53]. This may be explained by a premise that the automated identification of flowers is a comparatively difficult task. For example, the geometric rigidity of leaves is something that flowers might lack, especially in natural settings. Nevertheless, automated flower identification through the use of feature extraction does exist [2, 10, 11, 19, 22, 23, 36, 37, 38, 39, 41, 60]. We proceed to discuss some of the standard features and methods used in these studies.

In the approach by Huang et al. [23], a user has to manually draw the outline of the flower in an image, from which shape features are extracted.

In addition to shape, colour may also be used as a feature. The paper by Apriuanti et al. [2] deals with the identification of orchid species from images of their flowers. A flower is segmented from its background, which allows for the extraction of shape and colour features. For shape, distances across the segmented images are determined, in addition to the aspect ratio and roundness. To avoid problems caused by differences in scale, rotation and translation across images, features such as the area are not considered. Colour features are extracted from the hue and saturation components.

Hong et al. [19] propose a method of automated flower identification that relies on edge based contour detection, as well as colour. The images of the flowers are clustered into five colour groups in the HSV space. These clustered groups are used to build feature vectors to be used in the classification.

Finally, texture in images has been shown to be a useful feature for flower classification. In a paper by Zawbaa et al. [60], the input image is reduced to a set of binary images, from which texture patterns are extracted. A paper by Nilsback et al. [36] describes how texture can be extracted using convolutional filters.

Support vector machines (SVMs) have also proved successful for flower identification, as demonstrated in the work by Nilsback et al. [38]. In their work, a 103 class flower dataset is introduced, upon which feature extraction is performed by considering the shape, colour and texture of the flowers, as well as the spatial distribution of the petals. An SVM is subsequently used for classification. Although SVMs are useful for tasks where extensive and careful feature extraction is performed as a preprocessing step, they fail as a unified tool in which both feature extraction and classification can be optimised jointly.

2.1.2 Deep neural networks

With the rise of deep neural networks, specifically convolutional neural networks (CNNs), it is now possible for machine learning models to extract relevant features automatically, at a cost of requiring larger training datasets. Thus it is no longer necessary to perform manual feature engineering to obtain feature vectors. Rather,

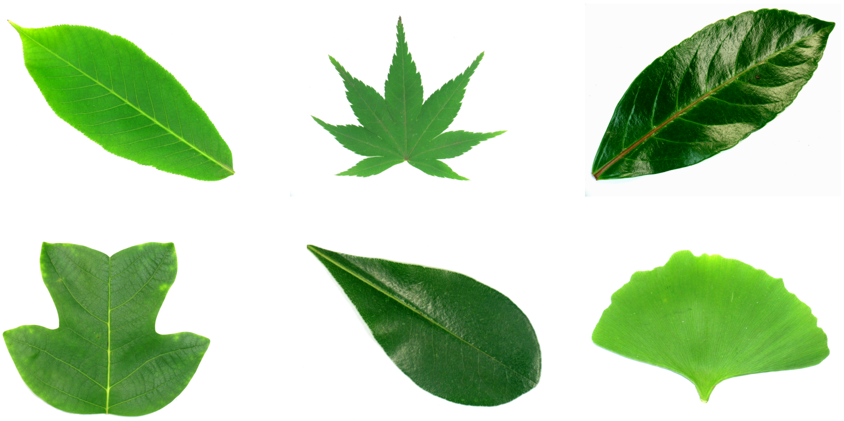


Figure 2.2: Examples of leaves from the Flavia dataset [58]. Notice how the images are taken in a controlled setting and are not representative of images taken in the wild.

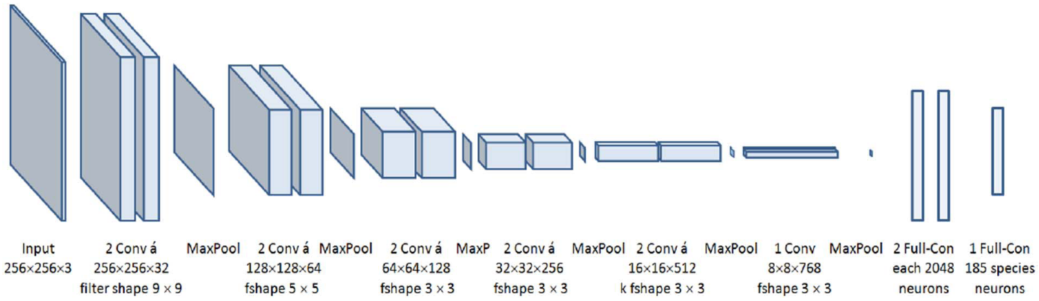


Figure 2.3: The LeafNet architecture. Image from [3].

the task of extracting statistically relevant feature representations from the images is left to the neural network. This implies potential for improved generalisation, which stands in stark contrast to the model-specific approaches from before [54].

In addition to the breakthroughs in terms of classification accuracy, the availability of graphical processing units (GPUs) has allowed for the training of deep CNNs with millions of parameters on large training sets [54].

Many studies focusing on plant identification using CNNs also use leaf based datasets, similar to those used in the feature engineering approaches. A number of studies construct their own CNNs, such as Zhang et al. [61] who use a six-layer CNN for leaf based identification of plants, trained on the Flavia dataset [58] (see Figure 2.2). Work by Barre et al. [3] improved on these results by building a 17-layer CNN.

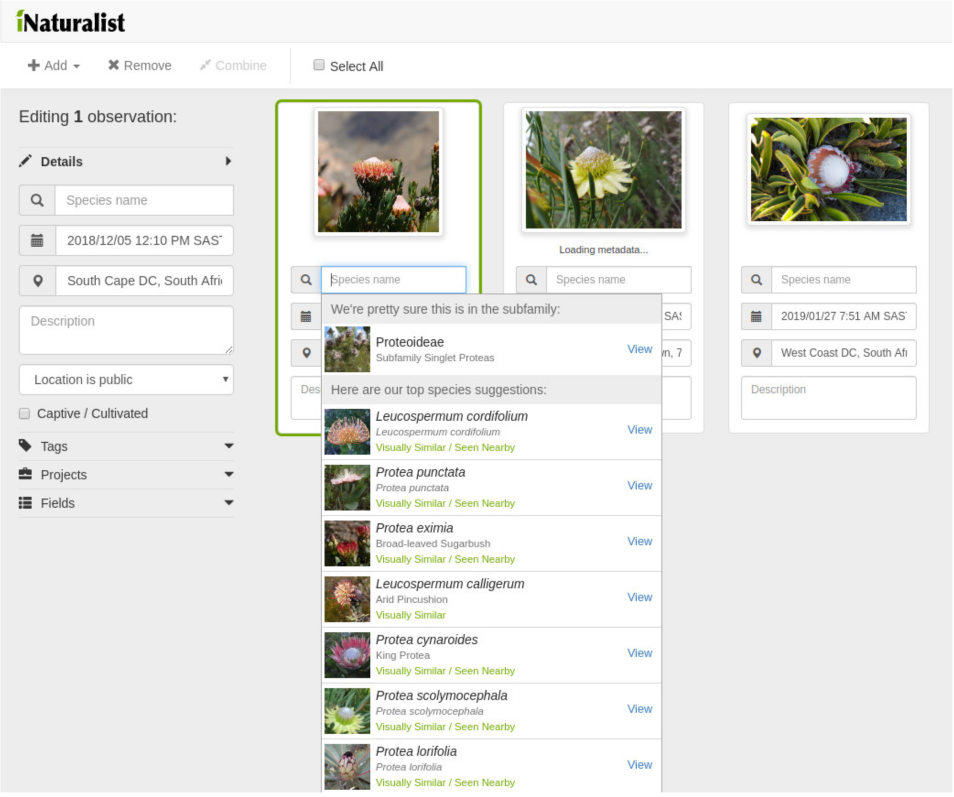
Another common approach is to leverage the power of pretrained CNNs and repurpose them for plant identification. This is referred to as transfer learning and is discussed in Chapter 4. One study considers the ResNet architecture [18] on the Flavia dataset and finds results superior to previous attempts. Other studies [46] use pretrained networks such as AlexNet [31] as a method of feature extraction, upon which an SVM is used for the classification. The latter makes use of the Oxford Flowers 102 dataset [37].

A further example is the CNN architecture called LeafNet [3], which was trained on the LeafSnap [30] and Flavia datasets to identify plants based on their leaves. This architecture can be seen in Figure 2.3.

2.1.3 Plant identification as a tool for the public

Machine learning for plant identification has spilled over into the public domain, with many applications rising to help the public identify plants without requiring much biological knowledge.

For example, LeafSnap [30] is a mobile application which allows people to identify trees based on images of their leaves. The idea is that a person uploads a picture of a leaf, whereafter the image is segmented from the background, and features are extracted. The developers created their own publicly available dataset of images from both lab and natural settings.



Editing **1** observation:

Species name

2018/12/05 12:10 PM SAS

South Cape DC, South Afri

Description

Location is public

□ Captive / Cultivated

Species name

We're pretty sure this is in the subfamily:

Proteoideae
Subfamily Singlet Proteas

Here are our top species suggestions:

Leucospermum cordifolium
Leucospermum cordifolium
Visually Similar / Seen Nearby

Protea punctata
Protea punctata
Visually Similar / Seen Nearby

Protea eximia
Broad-leaved Sugarbush
Visually Similar / Seen Nearby

Leucospermum calligerum
Arid Pincushion
Visually Similar

Protea cynaroides
King Protea
Visually Similar / Seen Nearby

Protea scolymocephala
Protea scolymocephala
Visually Similar / Seen Nearby

Protea lorifolia
Protea lorifolia
Visually Similar / Seen Nearby

Loading metadata...

Species name

SAS

m, 7

Species name

2019/01/27 7:51 AM SAS

West Coast DC, South Afı

Description

Figure 2.4: An example observation of *P. rupicola* (left), along with *P. scolymocephala* (centre) and *P. cryophila* (right) being uploaded to the iNaturalist web application. The iNaturalist recommender is used on *P. rupicola*, where it suggest *Leucospermum cordiifolium* as the most likely species candidate. Notice how the system also takes the location into consideration, by stating that the primary selection has been seen nearby. Although the system does not correctly identify the observation down to genus, or species, it is confident that the observation is in the *Protea* family.

Another application comes twofold from the crowd-sourced website iNaturalist [24]. They have a state-of-the-art CNN built into their website for species identification, and also provide multiple mobile applications on top of their identification system. When uploading a picture of a biological observation, the iNaturalist system makes a top-10 species recommendation (as demonstrated in Figure 2.4), with a higher-level genus recommendation if it is not sure on the species. The system also leverages locality data by considering similar observations in the nearby area, to refine its recommendation. This system is built into the iNaturalist and Seek [44] mobile applications. The iNaturalist application is an extension of the web application, and the Seek platform operates in a game-like manner to serve an educational purpose.

Similarly, Pl@ntNet [40] is a citizen science project where users upload pictures of plants, which are subsequently identified by their systems.

2.2 Fine-grained image recognition

Fine-grained image recognition (FGIR) is growing into an active field of research, with an increasing relevance to real-world applications [56]. This type of image recognition deals specifically with data where all the classes belong to the same super-category, such as species of bird, models of cars and in our case, species of *Protea*. As mentioned previously, there are clear challenges to this type of data, mainly due to the small interclass variability and relatively large intraclass variability, which may hinder standard CNNs.

There are a few standard methods to tackle FGIR problems. Firstly, a number of subnetworks may be employed to localise diagnostic features in the input images. These transformed vectors can then be used by a separate network for classification. An example of this would be the work of Zheng et al. [62], where the outputs of a pretrained CNN are used to construct an attention mechanism. The outputs of the pretrained network serve to extract only the relevant information from the image data, while the attention mechanism boosts this data to aid a separate subnetwork in its classification. Note that this approach does not rely on manual part based annotations of the images (as some earlier methods did [55]), but still only on the labels. Due to the time saved on image annotation, as well as the practical nature of networks that are able to operate on unannotated data, this method of attention subnetwork based FGIR has become a norm [16, 47].

End-to-end feature encoding is also a popular method for FGIR, with bilinear CNNs [33] as an example. The bilinear model consists of two parallel CNNs which act as feature extractors. The outputs of these networks are multiplied and pooled to produce a bilinear vector which acts as the image descriptor. A summary of such an architecture can be seen in Figure 2.5.

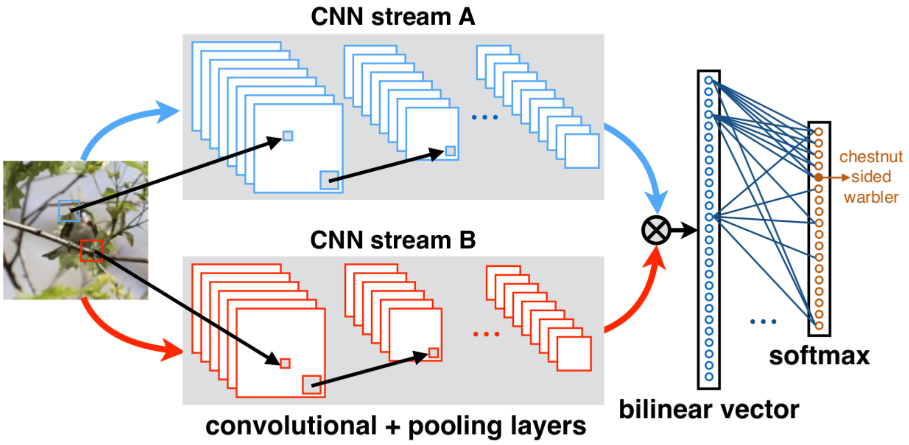


Figure 2.5: The bilinear CNN model proposed in [33], where we notice two parallel feature extraction networks whose outputs are combined for classification.

2.2.1 FGIR with attributes

A further method for FGIR relies on leveraging external information relating to the data, such as text based annotations, geospatial tags, or any other attributes relevant to the data.

A common source of external data comes from the web, as discussed in [56]. One approach is to scrape noisy web data to use as test data [64], thereby incorporating prior test data into the training set and thus increasing the training set size. The potential problems caused by the unreliability of annotated web data may be overcome by applying adversarial learning techniques, as introduced by Goodfellow et al. [17]. These methods seek to fool the network with noisy inputs, such that more robust representations of the data can be learnt.

There have also been advances in applying active learning to fine-grained image classification [29], which seeks to find only the relevant data (e.g. noisy images from the web) to train a network optimally. This approach works by training a network on a base set of labelled images, whereafter the model proceeds to iteratively select images to be annotated by an expert. Once the new images are labelled, they are included in the next training phase of the network. By not requiring all the data to be annotated, and allowing the network to pick only the most suitable images for labelling and training, the workload of the experts can be reduced significantly.

Additional modalities, such as the recorded location of an observation, have been used to solve biological classification problems. An example of this is BirdSnap [5] which, through an application of Bayes' rule, takes geographical distributions of bird species into consideration to aid an image based identification module. Similar geospatial methods are also employed by Tang et al. [49], where landscapes are classified with a CNN, with the aid of locations. For example, it may be hard to distinguish between snow and a field of flowers (as illustrated in Figure 2.6), yet if we know where and/or when the pictures were taken, it becomes much easier.

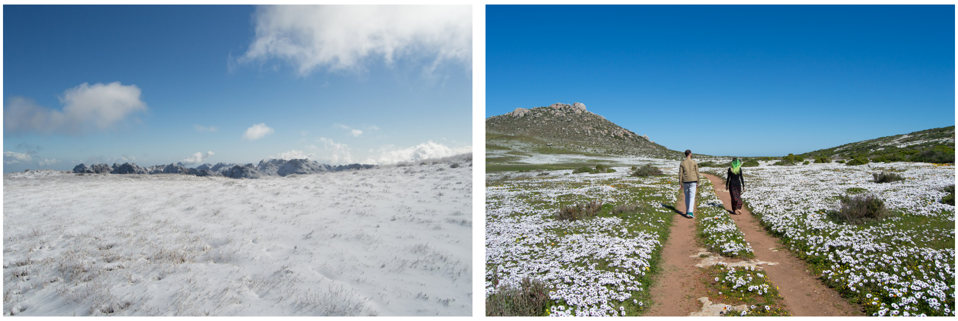


Figure 2.6: It may be easy to see how a conventional CNN may confuse the picture on the right for snow. Yet, if it is known that the photo on the left was taken high on the mountains above Stellenbosch during winter, while the photo on the right was taken on the West Coast of South Africa during spring, classification becomes much clearer.

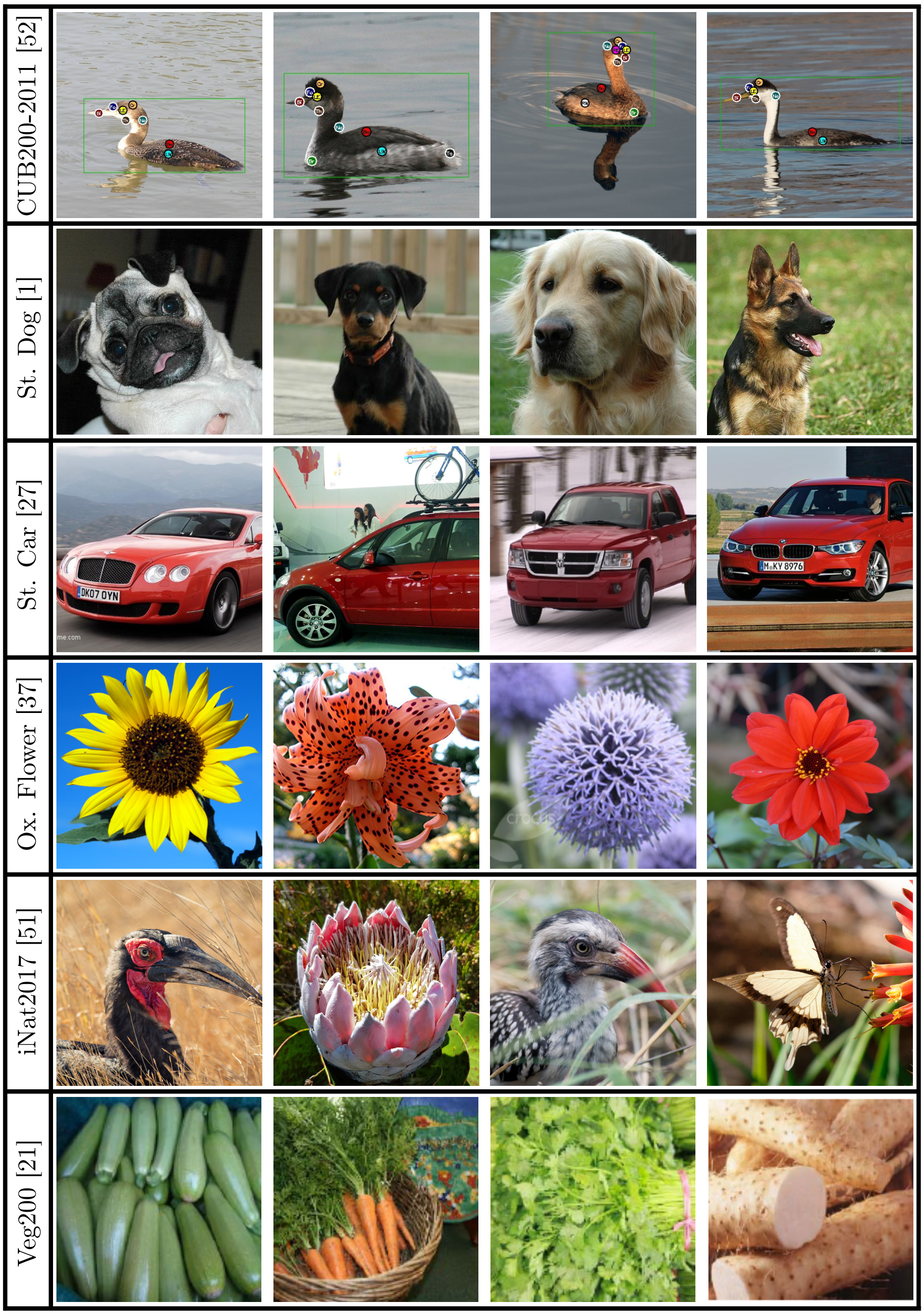


Figure 2.7: Examples images from popular FGIR datasets outlined in Table 2.1, arranged by row.

2.2.2 Fine-grained datasets

There are a number of fine-grained image datasets available to the public. We summarise a few of these datasets in Table 2.1 and provide example images in Figure 2.7. We also include information such as whether there are annotations, bounding boxes or attributes linked to the data.

Although all the datasets outlined below are potentially useful for biological applications, we do not use them for the training of any neural networks. We rather use the larger ImageNet dataset [14] which allows for the training of highly effective neural networks, as is discussed in Section 4.2.

Dataset	Superclass	Images	Classes	BB	PA	HR	AT	TX
Oxford Flower [37]	Flowers	8,189	102					✓
CUB200-2011 [52]	Birds	11,788	200	✓	✓		✓	✓
Stanford Dog [1]	Dogs	20,580	120	✓				
Stanford Car [27]	Cars	16,185	196	✓				
FGVC Aircraft [34]	Aircrafts	10,000	100	✓		✓		
Birdsnap [5]	Birds	49,829	500	✓	✓		✓	
Fru92 [21]	Fruit	69,614	92			✓		
Veg200 [21]	Vegetables	91,117	200			✓		
iNat2017 [51]	Life	859,000	5,089	✓		✓		

Table 2.1: A comparison of popular fine-grained datasets. BB refers to whether there are bounding box annotations, PA whether there are part annotations, HR whether the data has hierarchical labels (e.g. the taxonomic tree of plants), AT whether there are attribute labels (e.g. gender, colour, etc.) and TX whether there are text descriptions of the images included. This table is reproduced from [56].

— 3 —

Datasets

This chapter describes our process of collecting an annotated dataset of images of *Protea* species, as well as our construction of per-species distributions according to location, elevation and time of flowering. This will serve as the datasets that we use for our task of *Protea* species identification. The image dataset is interesting in that it presents a number of niche challenges which are typical of biological image datasets, as mentioned in Chapter 2. The data also serves as a surrogate for the plant diversity of the CFR, with the possibility of its study leading to interesting ecological applications.

We restrict our study to the Eastern and Western Cape of South Africa as a representative of the CFR, since the boundaries of the CFR are not strongly defined.

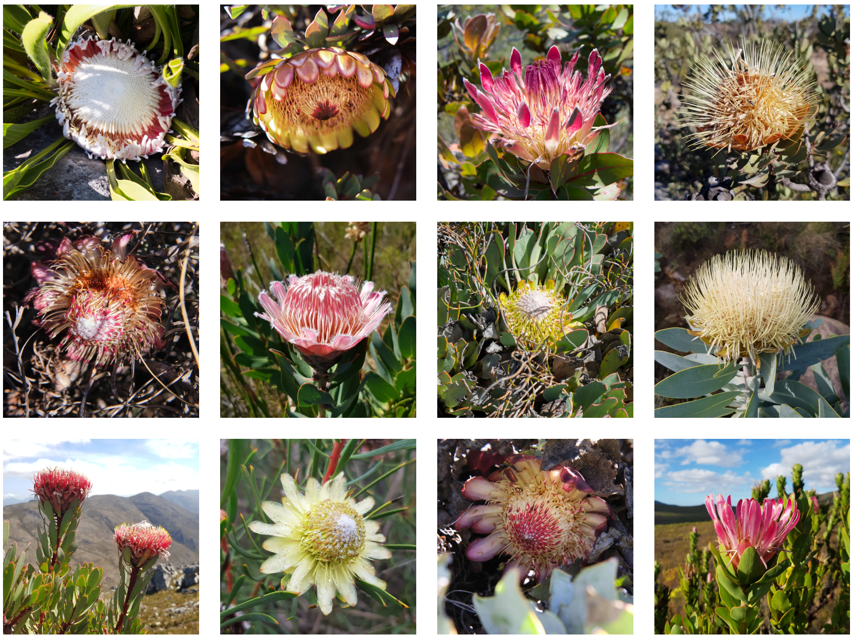


Figure 3.1: Example images of *Protea* from iNaturalist, which are included in the scraped data. The diversity of the genus is immediately evident, yet considering that all the images are of different species, the fine-grained nature of the data also becomes apparent.

3.1 iNaturalist

We collected images[1] from the crowd-sourced platform iNaturalist, where people across the world upload observations of fauna and flora they find in the wild, under a Creative Commons license. An observation typically consists of at least an image, a location, a date and a community-aided identification.

Of all the *Protea* records found on iNaturalist at the time of our dataset creation, we were interested only in those from non-cultivated observations in the CFR, whose identifications have been reviewed by multiple users. We also only kept images depicting flowering inflorescences, and restricted the dataset to species with at least 20 such images. This filter process resulted in a dataset containing 4,849 images in total, across 41 species, which is summarised in Table 3.1. We emphasise that the set is unbalanced in terms of samples per species, has fine granularity among many of the classes, and also contains significant variability in background, image quality, size of the inflorescence in the image, etc. It is, however, representative of the real world [51]. A key to the scientific names and abbreviated names (for the purpose of brevity) can be found in Table 3.1.

Every image corresponds to a latitude and longitude value of where it was taken, an elevation reading in metres above sea level, the date of the observation, and a community identification to species level. We note that elevation can be inferred from latitude and longitude by using the Elevation-API web application [15] with a 5 to 30

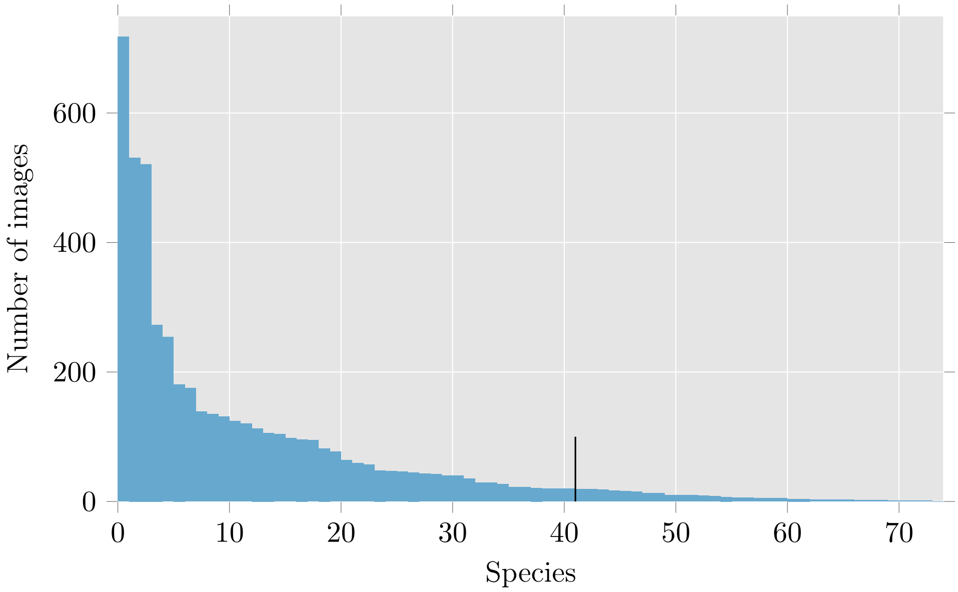


Figure 3.2: The distribution of number of images per species in our training and test set, indicating some degree of class imbalance and a long tail. We include the number of available images for the 74 species on iNaturalist in the Eastern and Western Cape. The vertical line at class 41 indicates the cutoff, whereafter fewer than 20 images per species were available.

[1] All images scraped may be found under the iNaturalist project "Sugarbushes of South Africa" at `https://www.inaturalist.org/projects/sugarbushes-of-south-africa`

Class	Key	Shortened key	Scientific name	# Images
1	PRREPE	REPE	*Protea repens*	718
2	PRCYNA	CYNA	*Protea cynaroides*	531
3	PRNERI	NERI	*Protea neriifolia*	521
4	PRNITI	NITI	*Protea nitida*	273
5	PREXIM	EXIM	*Protea eximia*	254
6	PRLAUR	LAUR	*Protea laurifolia*	181
7	PRLEPI	LEPI	*Protea lepidocarpodendron*	175
8	PRCORO	CORO	*Protea coronata*	139
9	PRSUSA	SUSA	*Protea susannae*	135
10	PROBTU	OBTU	*Protea obtusifolia*	131
11	PRACAU	ACAU	*Protea acaulos*	124
12	PRAUREA	AUREA	*Protea aurea aurea*	120
13	PRPUNC	PUNC	*Protea punctata*	113
14	PRMUND	MUND	*Protea mundii*	106
15	PRLONG	LONG	*Protea longifolia*	104
16	PRBURC	BURC	*Protea burchellii*	98
17	PRMAGN	MAGN	*Protea magnifica*	96
18	PRCPCT	CPCT	*Protea compacta*	95
19	PRLORI	LORI	*Protea lorifolia*	82
20	PRSPHL	SPHL	*Protea scolymocephala*	77
21	PRSPEC	SPEC	*Protea speciosa*	64
22	PRAMPL	AMPL	*Protea amplexicaulis*	59
23	PRLANC	LANC	*Protea lanceolata*	57
24	PREFFU	EFFU	*Protea effusa*	48
25	PRGLAB	GLAB	*Protea glabra*	47
26	PRSCBR	SCBR	*Protea scabra*	46
27	PRMONT	MONT	*Protea montana*	45
28	PRRUPI	RUPI	*Protea rupicola*	43
29	PRGRAN	GRAN	*Protea grandiceps*	42
30	PRNANA	NANA	*Protea nana*	40
31	PRSULP	SULP	*Protea sulphurea*	40
32	PRSRFL	SRFL	*Protea scolopendriifolia*	35
33	PRHUMI	HUMI	*Protea humiflora*	29
34	PRLORE	LORE	*Protea lorea*	29
35	PRCANA	CANA	*Protea canaliculata*	27
36	PRCRYO	CRYO	*Protea cryophila*	22
37	PRCORD	CORD	*Protea cordata*	22
38	PRLAEV	LAEV	*Protea laevis*	21
39	PRCAES	CAES	*Protea caespitosa*	20
40	PRVENU	VENU	*Protea venusta*	20
41	PRLACT	LACT	*Protea lacticolor*	20

Table 3.1: The key used to transition between the scientific names and the shortened names for display purposes. We list the total number of images, that will be split into training and test sets, as well.

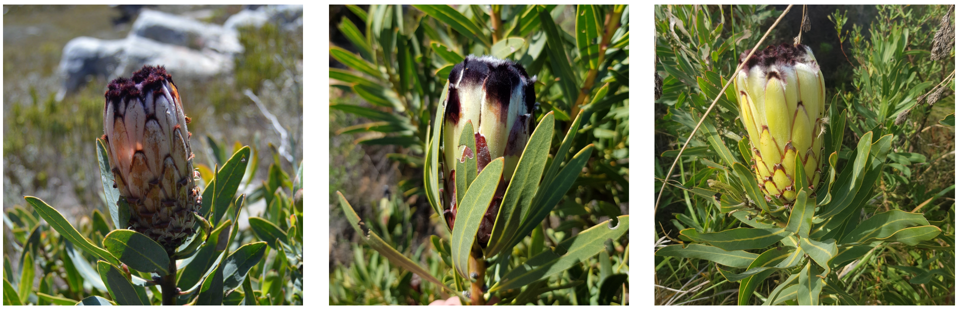


Figure 3.3: From left to right the three sister-species *P. laurifolia*, *P. lepidocarpodendron* and *P. neriifolia* showcase how our dataset has a high interclass similarity.

metre resolution. Nevertheless, we treat elevation as an additional attribute because of the sensitivity of certain *Protea* species to it. We also include the iNaturalist observation identification number in our dataset, for potential future use (to trace a specific observation, for example).

We split our dataset into a training set with 3,652 images and a test set with 1,197 images, by splitting the images of each of the 41 classes randomly with a fixed ratio. The test set is only used to assess the final performance of our various models, and is not used during training or any sort of cross-validation.

Challenges in the data

We re-emphasise that the images collected present a particular set of challenges, as outlined in Chapter 1.

A striking difference for our dataset when compared to general image datasets, is the lack of uniformity in the distribution of number of samples across classes. Accordingly, we refer to our dataset as having a long tail distribution. Consider the difference between *P. repens*, for which 714 unique images are available, and the multitude of species which have only 20 images each. The long tail is also evident in Figure 3.2, where we notice that the bulk of the image data is contained in the top few classes.

The second significant challenge is a result of the fine-grained nature of the data, as seen in Figure 3.1. As is the case with most fine-grained datasets [37, 52, 1, 34, 5, 21, 21, 51], there is high interclass similarity, as well as high intraclass variability. In Figure 3.3 we notice three different *Protea* species, namely *P. laurifolia*, *P. lepidocarpodendron* and *P. neriifolia*, which all look similar. On closer inspection we may notice subtle differences in the shape of the leaves, or colouring of the hairs on the involucral bracts[2], but none of these features are immediately apparent. It may be difficult even for an amateur botanist to distinguish between these species. On the other hand, in Figure 3.4 there are three images of *P. magnifica* which all look markedly different, in both colour and form.

[2] The bracts cover the inflorescence. These hairs on the bracts may also be referred to as the “beard” of the *Protea*.

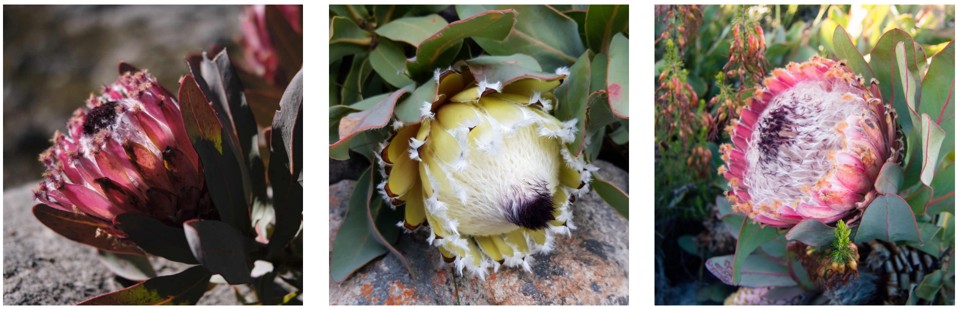


Figure 3.4: All three images are of *Protea magnifica*, yet they all are visibly different from one another. This emphasises the high interclass variability.

3.2 The Protea Atlas Project

The Protea Atlas Project was launched in November 1991 by Rebelo, in order to document the *Proteaceae* of Southern Africa. The project culminated in a vast collection of data: 252,513 species records at 61,591 locations [42]. Note that this dataset does not include image data.

We focus on records of our 41 *Protea* species, for an indication of where each species is found. We discretise the CFR into a gridmap, and for each species separately populate the grid cells with frequency counts from the Protea Atlas Project records. These frequencies are normalised and then interpreted as a probability distribution over observation location, given a species. We also construct similar distributions over elevation and flowering time, using the summarised data in Rebelo's field guide [43]. For every species we set up binary-valued distributions over discrete elevation intervals (in steps of 100m) and discrete flowering time of year (in months). These values are then smoothed to reduce potential quantisation effects, and normalised.

We discuss the details of the locality, elevation and flowering time data below. For each of these attributes we construct a probability table, which allows us to determine the likelihood of an attribute given a species of *Protea*.

3.2.1 Locality data

Each of the 61,591 localities of the Protea Atlas Project represents a location, limited to a diameter of 500m. These localities are irregular and often biased in their dispersal across South Africa. For example, Table Mountain National Park has a significantly larger set of observations than, say, remote areas of the Cederberg wilderness area. This is not due to the Cederberg being ecologically less important, but simply due to its comparative inaccessibility.

Discretising

A vector image of the Eastern and Western Cape is discretised into 11,425 regularly spaced points along both latitude and longitude, as seen in Figure 3.5. For each of our 41 *Protea* species, we take this discretised map and superimpose the exact location points for each observation from the Protea Atlas Project. Subsequently, we find the four nearest neighbours (from the discretised grid) for each observation

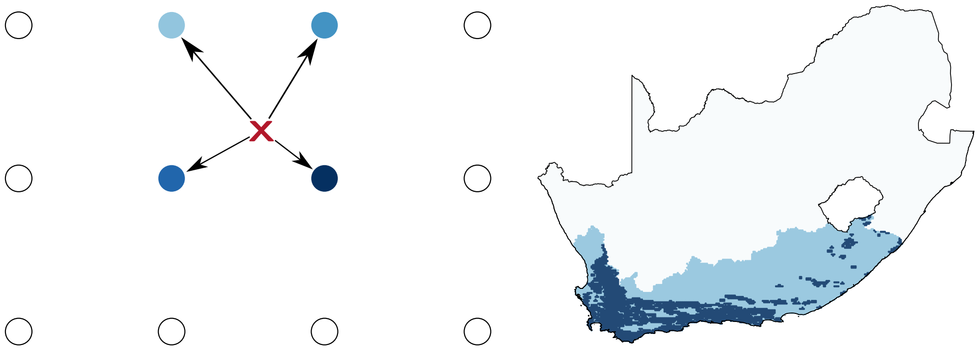


Figure 3.5: Left: A given GPS coordinate is discretised, by fractioning a unit weight of the true GPS location across its four nearest neighbours, based on distance. Right: The Eastern and Western Cape provinces (light blue) with all the Protea Atlas Project records discretised and superimposed (dark blue).

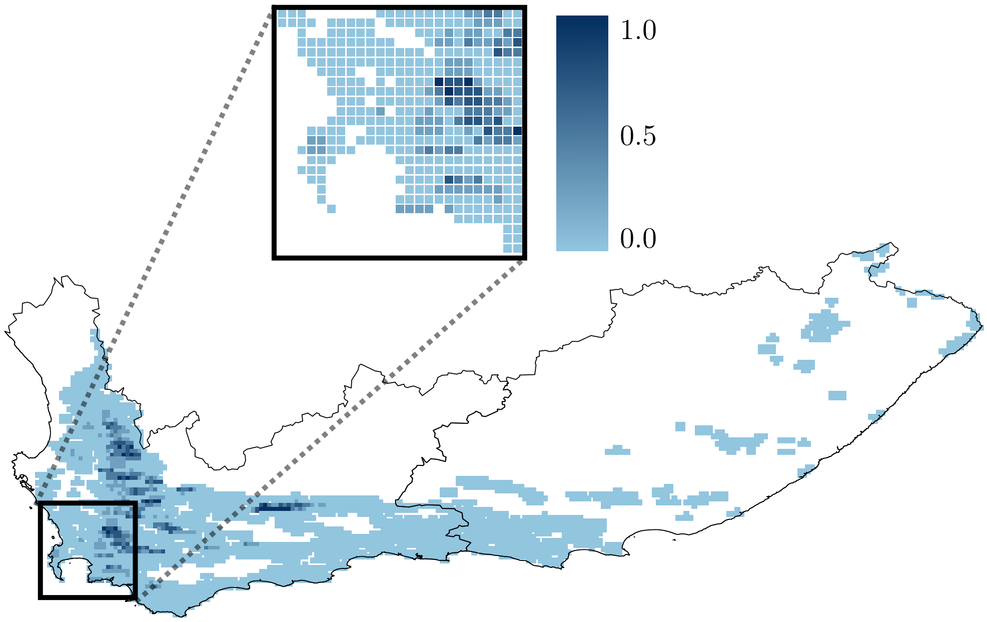


Figure 3.6: The natural distribution of *P. magnifica*. The entirety of blue on the map indicates those regions which were surveyed during the Protea Atlas Project, with the darker shades of blue indicating the presence of *P. magnifica*, according to the density of observations.

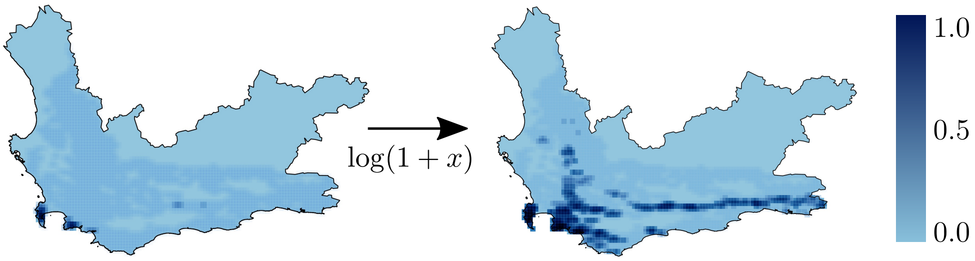


Figure 3.7: On the left we see the weighted distribution of *P. cynaroides* in the Western Cape, while on the right the true extent of the species range is more noticeable after $\log(1+x)$ is applied to its weighted distribution.

and weigh the importance of the observation across its four nearest neighbours, by assigning fractional weights to the neighbours according to their distances to the observation. This has the effect of smoothing the distributions of the various *Protea*, which can be seen in Figure 3.5. We sum the weights assigned to the discretised locations across all the observations for a given species to obtain a distribution maps for each species. For a given species, the magnitude of the weights at a certain discrete location indicates the likelihood of finding the species at that location.

Any location from our initial discretisation of the CFR that does not correspond to any records of the Protea Atlas Project is discarded from the map. The new refined map, as seen in Figure 3.5, now represents our region of study. We thus refine each of the 41 distribution maps to only include this refined set of location points. The process results in a distribution map for each species that comprises 3,206 points, compared to the 11,425 points in the initial discretised CFR.

This does not take care of the problem induced by the discrepancy in the number of Protea Atlas Project samples across the CFR. Regions that were well-sampled during the Protea Atlas Project, such as Table Mountain might be represented in excess, resulting in large weight values. This makes them incomparable with the less-sampled regions, such as the remote regions of the Cederberg. To tackle this, we apply the transformation

$$\log(1+x), \tag{3.1}$$

element-wise, to each distribution map. This function counteracts the incomparable weights which are a result of our method of weight assignment, by boosting the under-represented regions, while repressing the over-represented regions. An illustration of the effect can be seen in Figure 3.7, where we notice on the left that *P. cynaroides* seems to occur essentially only along the South-Western Cape coast. However, once applying the $\log(1+x)$ function on the right, we notice that the true extent of the species' distribution spans most of the CFR. In addition to illuminating the plots of *Protea* distributions, we found through informal experimentation that the transformation in (3.1) seems to improve classification.

The final discretised maps for each species is normalised and flattened into a vector. The result is a table which contains the probability values for each of the 3,206 locations, for each species of *Protea*, and will allow us to find the likelihood of observing a certain species at a given location.

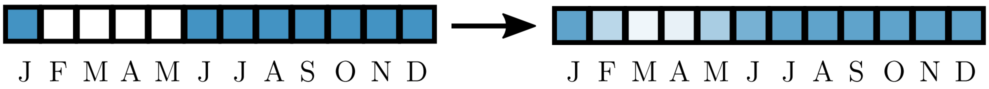


Figure 3.8: Gaussian smoothing applied to the flowering time of *P. magnifica*. Notice some of the cells (representing months of the year) change from no shading to partial shading after the smoothing has been applied.

3.2.2 Elevation data

We use the curated elevation profiles in Rebelo's field guide [43] to set up elevation profiles for each *Protea* species. To this end, we set up a list of discrete elevation intervals, ranging from 0m to $2,700$m above sea level, in increments of 100m.

Note that elevation data along the Cape mountain ranges is often difficult to work with. The Protea Atlas Project data allows for observations with a diameter of 500m from the recorded location, but in a region of that size the elevation profile may change drastically. It is for this reason that we use the field guide, rather than the raw Protea Atlas Project data.

Finally, we apply a one-dimensional Gaussian filter to the binary data. The standard Gaussian function,

$$G(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}, \tag{3.2}$$

is used to construct the filter, where we select $\sigma = 0.5$. This allows for a margin of error on the boundaries of the elevation distributions, when considering at which elevation a particular species is found.

As an example, consider the unlikely (but not completely impossible) event in which *P. magnifica* is observed at 1,100m. Before we apply Gaussian smoothing, this *Protea* would have a zero probability of occurring at this elevation, yet with smoothing, we gain a small, yet valuable probability for this event. Since we are working with real-world data, we should anticipate the event of such unlikely observations.

3.2.3 Flowering-time data

Similar to the elevation data, we use Rebelo's field guide to find the months during which separate species of *Protea* flower. We compile a discrete, binary list for each species, which indicates whether a particular species flowers during a particular month.

Gaussian smoothing, with $\sigma = 0.5$, is applied with "wrap-around" to each species, to allow for the small chance that a *Protea* may flower outside its expected flowering months. This smoothing is seen for *P. magnifica* in Figure 3.8. Note that we normalise the final vectors for both the elevation and date attributes.

— 4 —

Image identification

The goal is to perform *Protea* species identification from an observation, which we assume consists of a single image and additional (but optional) location, elevation and date information. The proposed model has three parts, the first of which deals with the identification of *Protea* species from images by using convolutional neural networks (CNNs).

In this chapter we explain the details of the image identification process, as well as providing background on the basics of neural networks, CNNs, transfer learning and visual attention.

4.1 Neural networks

A neural network is a collection of neurons grouped into layers. The simplest component of a neural network is the neuron, which consists of a set of inputs, weights and an activation function. Such a neuron with n inputs may be represented as

$$\sigma\left(f(x_1, x_2, \ldots, x_n)\right) = \sigma\left(\sum_{i=1}^{n} w_i x_i + b\right), \tag{4.1}$$

where $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ is a vector of input values, $\mathbf{w} = (w_1, w_2, \ldots, w_n)$, a weight vector, b a bias, and σ a nonlinear activation function. The function f represents the weighted sum of the inputs and the weights, with the addition of a bias term.

The activation function takes the output of the neuron and performs a nonlinear transformation, allowing us to model nonlinear functions.

The neurons are assembled to create layers, which are a higher-level building block for neural networks. Conceptually, a layer receives multiple inputs, which it transforms with sets of weight vectors, bias terms and nonlinear activation functions, to produce multiple outputs. These outputs are then received by a subsequent layer as input.

The layers between the first input layer and final output layer are referred to as the hidden layers. An example of the architecture for a basic fully-connected neural network with one hidden layer may be seen in Figure 4.1.

Once we have constructed a neural network, consisting of multiple connected layers, we wish to task the network with a specific goal, such as learning how to identify

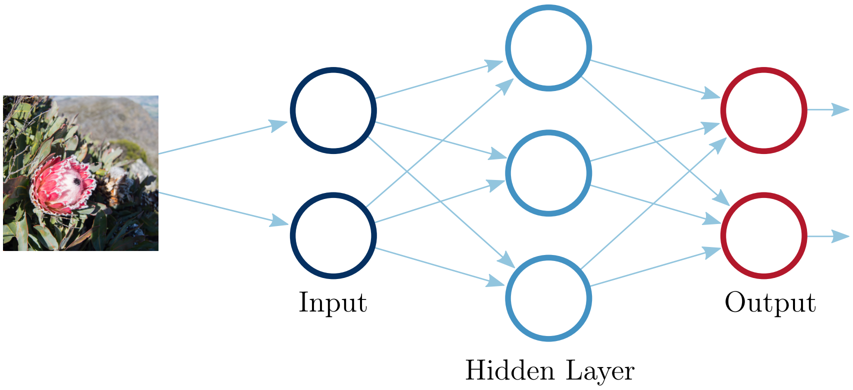


Figure 4.1: A simple feed-forward, fully-connected neural network. The network has two input neurons in the first layer, 3 neurons in the second layer (hidden layer) and two neurons in the final layer.

Protea species from images. To this end, we need to have a final output layer, from which we can infer a class label. We use the softmax function, which allows us to transform the output of the final layer of the network to a probability distribution (non-negative numbers that sum to 1). The softmax function is represented by

$$\sigma(z_i) = \frac{e^{z_i}}{\sum\limits_j e^{z_j}}, \tag{4.2}$$

where z_i is the output of neuron i, while j ranges from 1 to the number of neurons in the output layer of the network (which in our case would be the total number of classes, represented by Q). Notice that the exponential function transforms the output to a positive value, while the division by the sum ensures the values are normalised.

Loss functions

We use a loss function to measure the success of the network in performing its task. The goal of the network is to obtain the smallest possible loss value, which would indicate the smallest divergence between the network's output and the desired output (class labels). A standard loss function for classification is the cross-entropy loss,

$$L_k = L_k(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_{i=1}^{Q} y_i \log(\hat{y}_i), \tag{4.3}$$

where $\mathbf{y}$ represents the true label, while $\hat{\mathbf{y}}$ represents the estimated label, as output of the network for a specific input k. In order to obtain the network's total loss we sum (4.3) across all the inputs, indexed by k. The true label of the data is encoded by a one-hot vector. Notice that when $y_i = 1$ (where i is the position associated with the true class) we require $\hat{y}_i$ to be reasonably close to 1, such that $-y_i \log(\hat{y}_i)$ is reasonably close to 0.

Minimising the loss function

Backpropagation is the workhorse of neural network training and exploits the chain rule to perform gradient based minimisation of the loss function. Popular algorithms

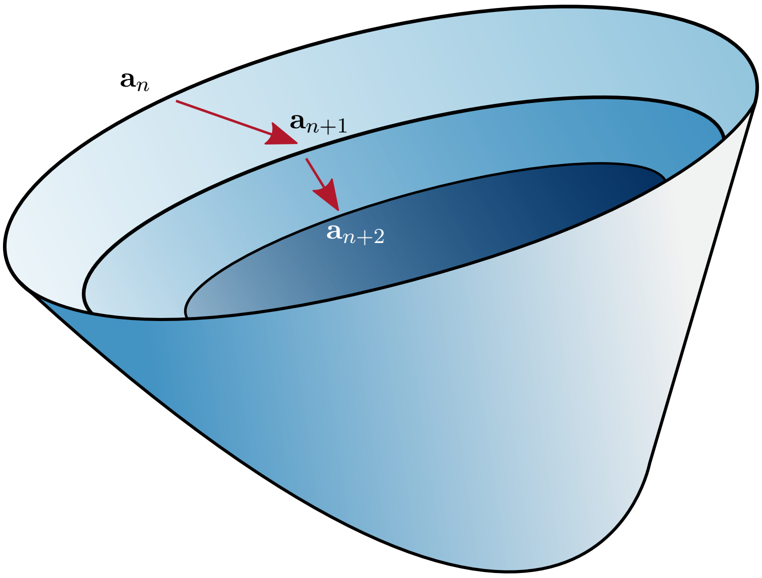


Figure 4.2: An example surface upon which we perform gradient descent, as defined in (4.4). Notice how with each iteration we move closer to the minimum on the surface.

to perform this optimisation include gradient descent, stochastic gradient descent and the Adam optimisation algorithm [28]. Standard gradient descent is defined as

$$\mathbf{a}_{n+1} = \mathbf{a}_n - \gamma \nabla L(\mathbf{a}_n), \tag{4.4}$$

where L is the loss function, $\mathbf{a}_n$ a vector of all the trainable parameters in the network (the weights and biases at each layer) and γ a real-valued constant called the learning rate. With each iteration we adjust the values of the parameters by using backpropagation to find the direction in which the loss function decreases most $(-\nabla L(\mathbf{a}_n))$ and move in that direction with a step size of γ. This concept is illustrated in Figure 4.2, on a loss surface over two parameters.

The size of γ determines the rate at which the network learns. Intuitively, a learning rate that is too large may mean that we never converge to a minimum, due to overstepping. On the other hand, a learning rate that is too small may hinder the ability of the network to find a minimum in a reasonable amount of time. A small learning rate may also inhibit the training algorithm's ability to escape local minima. One possible solution to these extremes is to introduce an adaptive learning rate which changes according to the gradient of the loss surface, as is done in the Adam optimisation algorithm [28].

Regularisation

Neural networks may overfit to training data, which would hinder their ability to generalise to test data. A solution to this is to introduce a form of regularisation, which refers to a set of strategies that help reduce the test error. The main form of regularisation we consider is dropout, where a number of neurons are randomly dropped with a certain probability. This changes the architecture of the network during each training step, with the effect of introducing beneficial noise into the network. It limits the network's ability to overfit to the training data by removing dependency on specific neurons, while encouraging robust representations of the data. Other forms of regularisation include the addition of a penalty term to the loss function, such as L^1 and L^2 regularisation. L^1 regularisation adds a scaled sum of the absolute value of the weights of the network to the loss term, while L^2 regularisation considers the squares of the weights.

A further consideration to regularise a neural network, and to improve computational efficiency, is the use of mini-batches. Here, a number of training samples are grouped together so that the network computes gradients with respect to the network parameters using the mini-batch, as opposed to using all the training samples. For the formulation in (4.4) we compute the gradients by using all of the training data. In comparison, with mini-batches we compute the gradients by using batches of the data. The advantage of using mini-batches is twofold, in that it optimises memory usage, while also introducing advantageous noise in the network gradients. This has the effect of aiding an optimisation algorithm from succumbing to local minima.

Batch normalisation [25] is an additional tool for network optimisation. We may notice that the input distributions of hidden layers change drastically during training as a result of small changes in the parameters of preceding layers. This has the downside of forcing these hidden layers to relearn relevant weights with each training epoch, which requires careful network initialisation and extensive hyperparameter selection to combat. The phenomenon is referred to as internal covariate shift. A solution to this is to normalise the input of each layer, for each mini-batch, through batch normalisation. By normalising the activations of each layer, the parameters the subsequent layers need to learn do not change dramatically during training.

The number of epochs for which all the training data is passed through a network may further help to optimise the performance of a network. We may expect that with too many epochs, the network adjusts its parameters to the training data too well, which would hinder its ability to generalise to the test data.

The techniques introduced above are beneficial for building neural networks which are less susceptible to overfit to training data. In the next section we introduce the concept of a convolutional neural network.

4.2 CNNs

A simple neural network similar to the description above may be tasked to solve our problem of *Protea* identification, but the number of connections needed between the input layer (which would accept all the pixel values of an image) to the first hidden layer would be impractically large. In this section we introduce the concept of convolutional neural networks, which are more efficient for dealing with image data.

A convolutional neural network (CNN), is a specialised neural network. Originally introduced by Yan LeCun [32] in 1989, it has grown to become a dominant building block for vision based problems.

In contrast to standard fully-connected networks, CNNs use the convolution operation in the first number of layers. This allows the network to focus on the structural

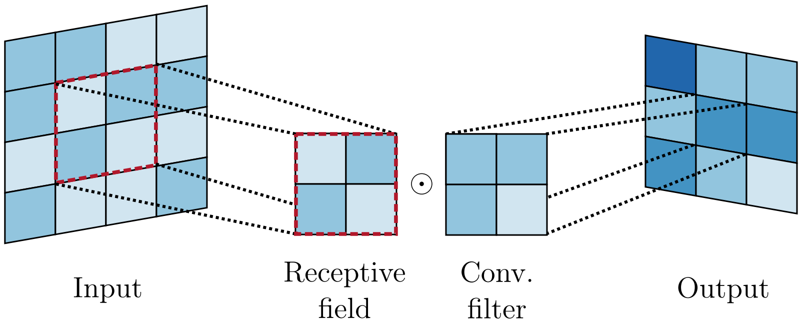


Figure 4.3: A convolutional filter (centre right) is swept across an input image (left), in order to produce the output (right). Notice how the filter acts over a particular portion of the input image at one time, which is referred to as the receptive field (dotted red box, left). This receptive field is multiplied element-wise with the filter and the resultant matrix is summed to produce one value of the output.

composition of the image, on a more local scale, by considering neighbouring pixels in an image. For example, a CNN may learn visual features present in *Protea* flowerheads (for example edges, shapes, colours and textures), which could help to discriminate between separate species.

Since we are working with visual image data (two-dimensional images, where each pixel is represented by an RGB value), we consider the two-dimensional discrete convolution[1]. This may be expressed as

$$(I * K)(x, y) = \sum_{m=-M}^{M} \sum_{n=-N}^{N} I(x + n, y + m)K(n, m), \tag{4.5}$$

where I and K represent our image and filter, respectively. An example may be seen in Figure 4.3.

During training, CNNs learn sets of filters which extract meaningful features (feature maps) for eventual classification. We convolve these sets of filters with the input space, with each of the subsets of the input space considered for a particular output value referred to as a receptive field.

An additional operation called pooling further modifies the output of a convolutional layer by reducing its size. This is achieved by replacing the output of a layer at a certain location, by a summary of the nearby outputs. It helps to reduce the overall number of parameters of the network while also enhancing the ability of the network to be invariant to small translations of the input. We typically use max pooling [63], which replaces the output of a convolved input at each location with the maximum value in a window of preset size around that specific location. Other pooling options include average pooling, or a weighted pooling which weighs each entry in the window on its distance to the central pixel. An example of max pooling may be seen in Figure 4.4.

[1] The formulation in (4.5) is actually cross-correlation, and not convolution. However, in the computer vision community it has become standard to use cross-correlation rather than convolution in CNNs, and to refer to both operations simply as convolution.

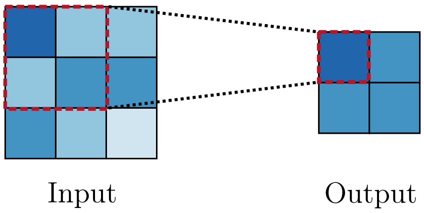


Figure 4.4: An example of max pooling. Notice how the maximum output of each receptive field is taken.

Note that for both the convolutional and pooling filters, we have freedom in the size and stride (the horizontal and vertical translation of the filter at each step). We may also choose the number of filters we wish to learn for each layer.

4.2.1 Advantages of CNNs

There are a number of reasons to use CNNs for the task of image classification, over standard fully-connected neural networks. Firstly, the number of parameters the network needs to learn is greatly reduced. For a fully-connected network, every neuron of a layer is connected to all the neurons of those layers adjacent to it. In contrast, CNNs are sparsely connected, due to the small size of the receptive fields. For example, an image of a *Protea* may consist of millions of pixels, yet the number of pixels required by the filters to isolate local image features useful for classification may be several orders of magnitude smaller.

Secondly, the weight parameters of convolutional filters are shared across the entire input. While standard neural networks have a distinct weight connecting every input neuron to every output neuron, CNNs use the filter weights at each location of an input and consequently learn only one set of weights per filter.

Finally, CNNs are naturally invariant to translation in the image. For example, a CNN may recognise a *Protea* within an image regardless of its location within the image.

An example of a powerful CNN is the Inception-V3 network, which is explained next.

4.2.2 Transfer learning and saliency maps

Transfer learning is the process whereby the weights of a trained neural network are utilised for a new network with a new task. The architecture of the trained network may be copied, and a number of layers may be added and/or removed to form a new network. The weights of the new network are initialised with the corresponding weights of the pretrained network. We have freedom to fix the inherited weights and simply train the weights of the layers that differ from the pretrained network, or we may free all the weights in the network and update the inherited weights during training. We may expect a CNN pretrained on image data to have learnt valuable filters which could be generally applicable to real-world images, including those that recognise rudimentary shapes, lines and colours. By transferring the weights of a pretrained CNN to a new CNN, we are reusing potentially valuable feature extractors for the new network. Whether we fix the transferred weights, or free all the weights

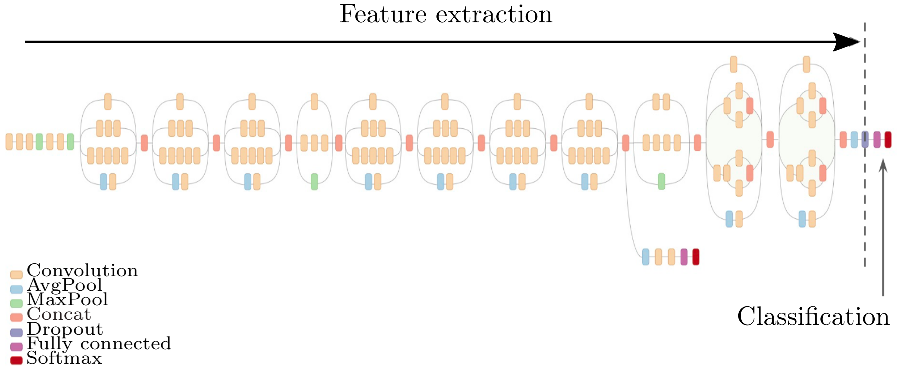


Figure 4.5: The Inception-V3 architecture. Image credits can be found in the appendix.

for training, we need to follow a process of fine-tuning. This refers to the refinement of the new network's parameters (with its new task and data) by using gradient descent, with the pretrained network's parameters as initialisation.

One option for a pretrained network is the Inception-V3 network [48] (displayed in Figure 4.5), which was tasked to learn a thousand classes from over a million images, for the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2015. The network itself is 42 layers deep with about 23 million parameters, and required considerable time to train on multiple GPUs. The value of being able to reuse these weights for a new network is clear, even if one considers only the training time.

As already mentioned, we expect the network to have learnt valuable features within the training images, which allows it to differentiate between classes. For example, we may query the Inception-V3 network (trained on ImageNet [14]) with a photo of a *Protea*. We would expect the early layers in the network to highlight the flower as a prominent object of interest, against the more uninformative background. It does this by focusing its attention to the region of importance in the image, i.e. the the portion of the image containing a *Protea*. The term "saliency map" embodies the idea of visual attention, and is a heatmap which highlights the pixels in a given image which had the greatest contribution to the network's classification. Thus it indicates locations in the image which gained the most attention from the network. For our application we choose the Inception-V3 architecture for its good balance between complexity and performance, as indicated in Figure 4.6.

The above definition of attention may seem somewhat recondite and an alternative explanation from the viewpoint of CNNs may offer more insight. Consider an image of a *Protea*, as well as an arbitrary CNN tasked to identify species in the genus. In order for the network to accurately identify the species, it needs to emphasise (or focus its attention on) the portion of image which contains the inflorescence, and to suppress (or ignore) the other parts of the image. Through training, we expect the network to learn suitable convolutional filters that will generate this sort of saliency map. These filters would fire at locations in the image which the network has learnt to be useful for classification.

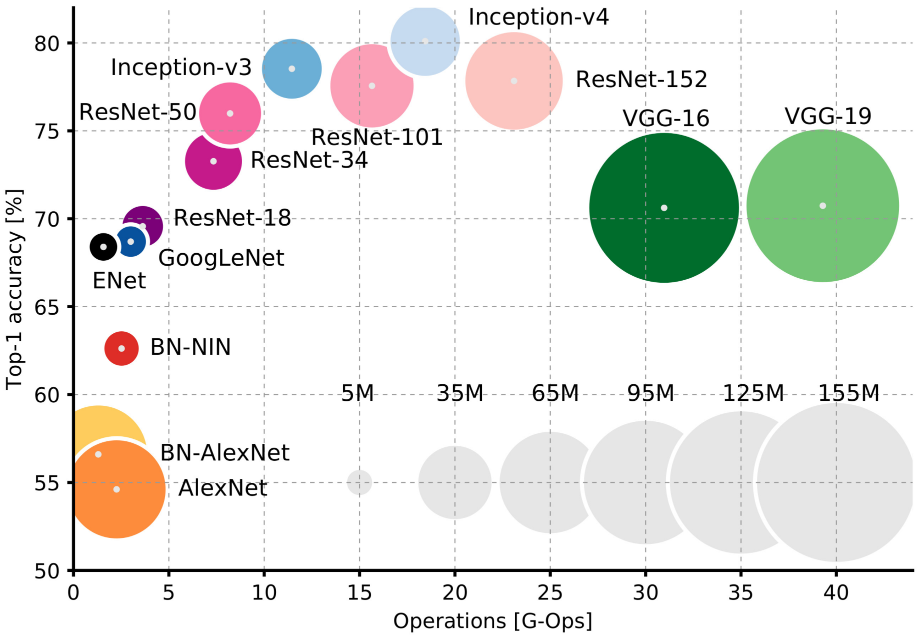


Figure 4.6: A comparison of the top-1 accuracy, number of operations and number of parameters of popular CNN architectures. The size of each circles is proportional to the number of parameters in the network. Notice how Inception-V3 has one of the best top-1 accuracies, with a comparatively small number of parameters and operations. Image from [7].

We may take this idea of saliency maps further with transfer learning. Rather than using a relatively small network trained from scratch on a limited dataset, we can consider a powerful network pretrained on a separate larger dataset. The Inception-V3 network, for example, does not have prior knowledge of *Protea* species, yet we can still utilise the convolutional filters which the network has learnt over hours of training, on a very large set of images. This idea hinges on the hypothesis that such a network has learnt general filters which are able to extract regions of importance regardless of whether it has seen data of the new set of classes before.

In the next section we introduce our specialised CNN to tackle the *Protea* identification problem.

4.3 CNN with attention

In this section we introduce our image identification model, which makes use of a CNN and an attention mechanism to transform images to normalised class scores associated with the different *Protea* species. As a first baseline we make use of the Inception-V3 architecture with weights pretrained on ImageNet. We freeze the convolutional layers, replace the last five layers such that a 41-class softmax output is produced, and train the network on the training set of *Protea* images from Section 3.1.

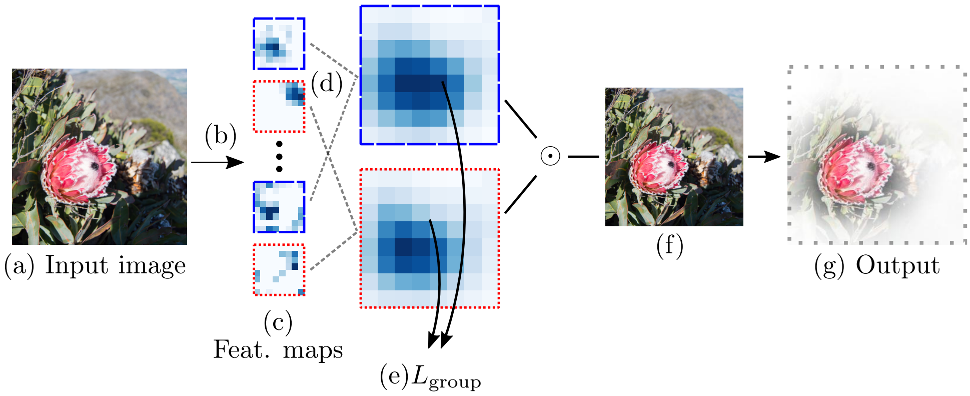


Figure 4.7: An image (a) is passed through the convolutional layers of Inception-V3 (b), yielding 2,048 feature maps (c). These are combined into attention maps through two fully-connected networks (d), learned jointly through the minimisation of a group loss (e). The two maps are added, scaled and multiplied with the original image (f), to produce an attention-boosted image (g).

Prompted by the unconstrained nature of images from field observations, as well as potentially large variations in backgrounds, we opt to explore the inclusion of an attention mechanism, as discussed in Section 4.2. We base this component on the multi-region method of Zheng et al. [62], which learns to find a preset number of attention regions in an image specifically for fine-grained image classification. We extract two regions per image, which we combine and pass to the next phase of the model. Through informal experiments on the training set we found this approach to perform better than a single-region attention model, likely because of the extra constraints that the second region imposes on the first during training (as explained below).

More specifically, a 299 × 299 colour image is fed into the convolutional base of a pretrained Inception-V3 network, yielding 2,048 feature maps each of size 8×8. For simplicity in the mathematics to follow, these feature maps are collectively denoted by $W * X$, where W represents the filter weights in the convolutional layers of the pretrained Inception-V3 network, while X represents the input. For each of the N training images, the corresponding 2,048 feature maps can be thought to represent the peak responses for the particular training image.

As illustrated in Figure 4.7 (d), we create two separate combinations of these feature maps that will form the two attention maps over an input image. The transformation from feature maps to attention maps can be performed by two fully-connected neural networks [62]. In order to initialise these networks, the feature maps are clustered into two groups by k-means on their peak responses over the training set, and then averaged per cluster into attention maps M_1 and M_2. Below, we explain the details of the two fully-connected neural networks, as well as the k-means clustering to initialise these networks.

4.3.1 Creation of the attention maps

We summarise each of the 8×8 feature maps by only considering the coordinate pair (t_x^n, t_y^n) corresponding to the maximum value of this filter, where $n \in \{1, 2, \ldots, N\}$. These coordinate pairs are now grouped by image into a feature channel vector, as follows:

$$F_c = \left[t_x^1, t_y^1, t_x^2, t_y^2, \ldots, t_x^N, t_y^N\right]_c, \tag{4.6}$$

where $c \in \{1, 2, \ldots, 2048\}$. Subsequently, we perform k-means clustering on these F_c vectors, to obtain two vectors of the form

$$I_i = \left[\mathbb{1}\{1\}, \ldots, \mathbb{1}\{c\}, \ldots, \mathbb{1}\{2048\}\right]_i, \tag{4.7}$$

where $\mathbb{1}\{c\}$ equals one if the c^{th} feature channel vector belongs to the i^{th} cluster and zero otherwise, and in our case with two attention regions we have that $i \in \{1, 2\}$.

Reverting to the convolutional output $W * X$ above, we wish to set up a pair of neural networks which takes as input the convolutional outputs, and maps them to a weight vector,

$$D_i(X) = f_i(W * X), \tag{4.8}$$

where $D_i(X) = [d_1, d_2, \ldots, d_{2048}]$ and $i \in \{1, 2\}$. Each d_c corresponds to a value which indicates the relative importance of the feature map c to the cluster i.

In order to obtain the two neural networks f_1 and f_2, we train them to fit to the indicator functions in (4.7), by using a binary cross-entropy loss. This is a special case of cross-entropy loss with two classes ($Q = 2$), as given in (4.3) of Section 4.1. We thus have two networks, f_1 and f_2, which take as input $W * X$ and output I_1 and I_2, in place of D_1 and D_2 respectively.

The networks output weight vectors D_i (initialised to output I_i), and the next goal is to optimise both of these networks to refine the attention regions. The networks are adapted to output attention regions by utilising the weight vectors as follows:

$$M_i(X) = \mathbb{N}\left(\hat{M}_i(X)\right) = \mathbb{N}\left(\sum_{c=1}^{2048} d_c \left[W * X\right]\right), \tag{4.9}$$

where $\mathbb{N}$ indicates a process of normalisation which involves subtracting the minimum value of $\hat{M}_i(X)$ and dividing by the resultant maximum. The values of the attention regions $M_i(X)$ are in the range $[0, 1]$, so we may interpret them as attention scores for the corresponding regions of the input image X.

The model is now optimised by a specifically crafted loss function, as explained next.

4.3.2 Loss and training

The networks f_1 and f_2 are initially trained to reproduce the indicator functions in (4.7), and produce attention maps M_1 and M_2. The networks are further fine-tuned under a grouping loss (L_{group}) that favours tightness within each map and dissimilarity between them. We may motivate the need for tightness by noting that the initialised attention region contains visually important pixels and we would want to keep attention fixed on a coherent region of such pixels. The need for dissimilarity between the attention regions is also important, since we wish for both regions to learn separate, yet visually important attention regions. If we disregard tightness, both filters may shift their attention away from the initialised attention regions, yet if we disregard dissimilarity, both filters may converge to the same attention region.

We compute the grouping loss over attention map i, where $i \in \{1, 2\}$, and may express it as

$$L^{(i)}_{\text{group}} = \sum_{(x,y)} M_i(x,y)\big[(x-p_x)^2+(y-p_y)^2\big]+\lambda \sum_{(x,y)} M_i(x,y)\big[M_{3-i}(x,y)-\alpha\big], \quad (4.10)$$

where $M_i(x, y)$ is the value of attention map i at grid location (x, y), (p_x, p_y) the location of the maximum value of M_i, and α a scalar margin. The importance of the first term (for in-map tightness) relative to the second term (for between-map dissimilarity) is controlled by the hyperparameter λ. Conceptually we see that the first term is relatively small when $M_i(x, y)$ is large in a region near (p_x, p_y), since $x - p_x$ and $y - p_y$ are both small. This may also happen if $M_i(x, y)$ is small when far from (p_x, p_y). The implication is that the first term in (4.10) is small only if the attention is focused in one tight region. In the second term, we may notice that the element-wise multiplication and subsequent summation of the two maps is only small if the maps are dissimilar. The index $3 - i$ is simply a short way to indicate the other mask: if $i = 1$ we have $3 - i = 2$ and if $i = 2$ we have $3 - i = 1$. We do not want corresponding locations in the maps to be large, since this would make element-wise multiplication and subsequent summation large.

As illustrated in Figure 4.7, we train f_1 and f_2 under L_{group} to produce M_1 and M_2. These maps are then added, normalised to the range $[0, 1]$ and scaled to the dimensions of the input image to form the attention map $M_1 + M_2$. The new map is multiplied element-wise with the original image, to highlight regions of attention in the original image.

By passing all the training images through the attention network, we are able to produce a set of images reminiscent of the original training images, with the important attention regions (which would hopefully be mainly the *Protrea* inflorescences) emphasised. The resulting attention-boosted images are used to train a CNN similar to the fine-tuned Inception-V3 network described at the beginning of this section. A network such as Inception-V3 is already well-suited to image classification problems. By isolating the visual structures in the training dataset which are important for identifying different species, we hope that the task of classification is simplified for the Inception-V3 network.

We note that the two components described above, namely the attention map extractor and the attention-boosted image classifier, can be optimised end-to-end or alternately for a number of iterations (similar to what is done in [62]).

— 5 —

Subgenus network and attributes model

In this chapter we describe the subgenus network and attributes model, which form the second and third components of our complete *Protea* identifier. The subgenus network is an image based classifier that relies on a CNN, while the attributes model constitutes a naive Bayes model.

The manner in which we combine the attention based CNN from Chapter 4 with the subgenus network and attributes model is also discussed.

5.1 Subgenus network

The 41 species of *Protea* in our dataset can be grouped into 14 subgenera according to common traits, as exemplified for the Spoon-bract subgenus in Figure 5.1. The details of subgenus grouping may be found in [43] and are summarised in Figure 5.2. The second part of our model attempts to classify a given image into one of these subgenera. It can be regarded slightly easier than the full 41-class problem, due to the 14 classes being less fine-grained and more distinct, the data being less unbalanced, and the availability of more samples per class. The aim is to boost the 41-class output of the attention based CNN by incorporating the 14-class output of the subgenus network. The idea is that the 14-class subgenera scores may scale the 41-class species output, thereby increasing the scores of those classes corresponding to a specific subgenera which the subgenus network is more sure of.

We employ a pretrained Inception-V3 network, replace the last five fully-connected layers, impose a 14-class softmax layer as output, and train the new layers with our

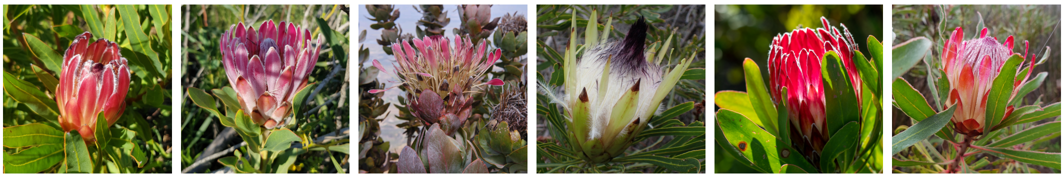


Figure 5.1: Examples of each of the Spoon-bract Sugarbushes included in the study. The inner involucral bracts are all spoon-shaped, which gave rise to the name of the subgenus.

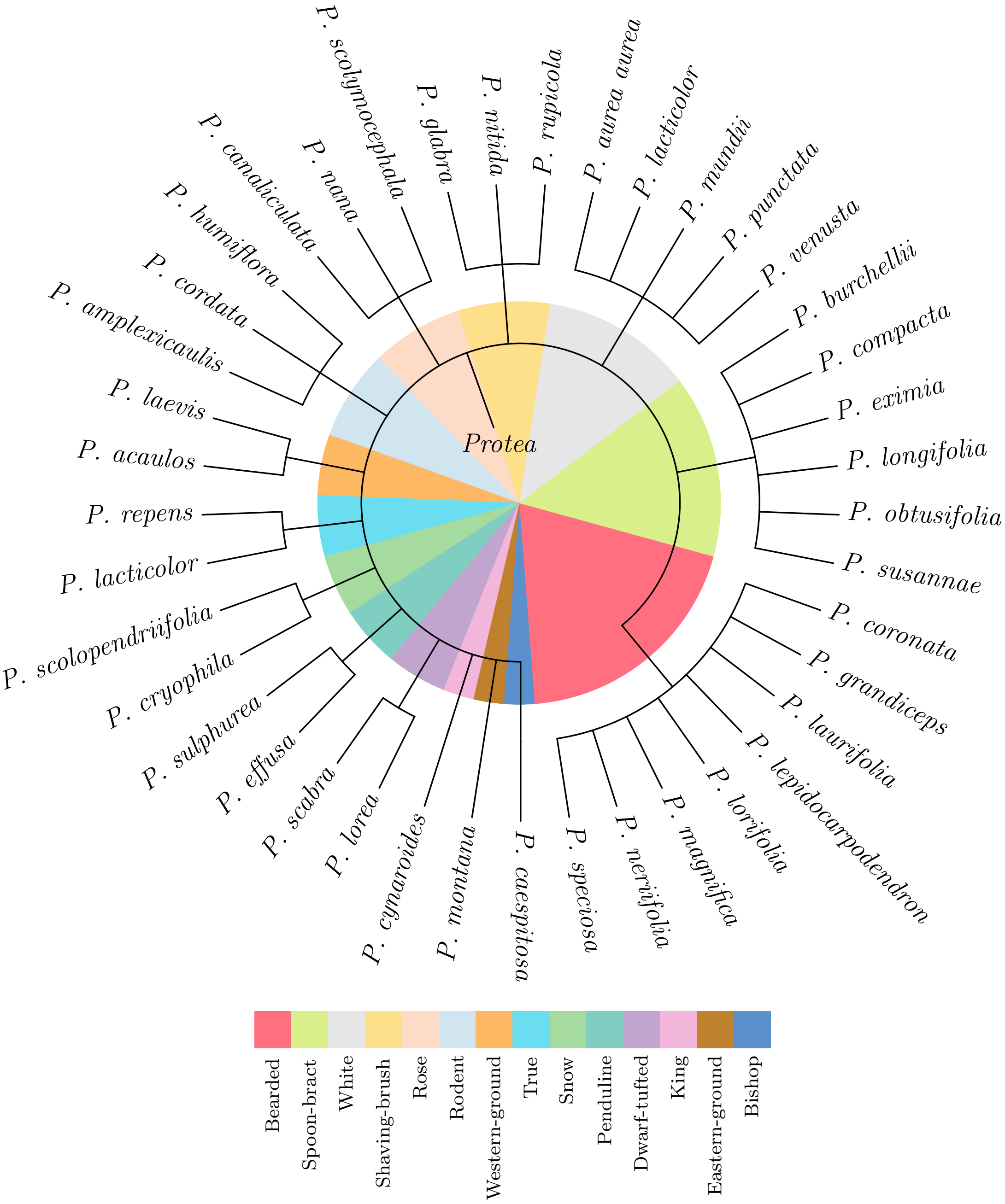


Figure 5.2: The taxonomic tree shows the 14 subgenera for the 41 species of *Protea* that we consider. Each colour represents a separate subgenus, as detailed by the legend.

data. Relabelling our training data from species to subgenera is straightforward, using Figure 5.2.

The subgenus classifier produces 14 class scores for a given image, which we turn into scores over the 41 *Protea* species of our original problem by splitting the score of a subgenus among its children species, equally. This process is exemplified in Figure 5.3, where we construct scores for a theoretical subgenus network with 3 subgenera over 6 species. The first subgenus contains 2 species, the second contains 1

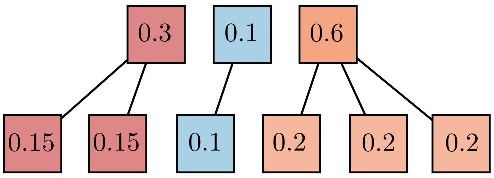


Figure 5.3: Example to show how output is scaled to 6 classes from 3 classes. The top row indicates the scores from an example subgenus network and the bottom row the scaled scores.

and the third contains 3. Each of the final scores is obtained by dividing the subgenus scores by the number of children species.

To obtain the scores for each of the 41 species from the 14 subgenus network scores is a straightforward generalisation of the example above. We calculate

$$P_i = \frac{S_i}{n_i}, \tag{5.1}$$

where P_i is the score the subgenera network assigns to *Protea* species i, S_i is the score of the subgenus, as predicted by the network, and n_i is the total number of species within the subgenus. Here we essentially assume a uniform distribution over the species given the subgenus.

An alternative would be to incorporate the class imbalance over the species in a particular subgenus, but there is a risk of overcompensation since the species-level CNN with which the subgenus classifier is to be combined might already be learning the class imbalance.

We incorporate the 41 scores produced by the scaled subgenus network with the 41 scores of the attention based CNN from Chapter 4, through element-wise multiplication. If the subgenus network has a low score for a certain subgenus, it will have low scores for each of its children species, and the scores of all those species within the subgenus are lowered. The subgenus networks effectively reweights the scores of the attention based CNN in order to give preference to those species which the subgenus network is confident about.

5.2 Attributes model

For the third part of our model we consider the availability of three attributes accompanying an image, namely location, elevation and date. These attributes are usually available on a citizen science platform such as iNaturalist. The location can be mapped to our discrete grid map from Section 3.2, by assigning the true location to one of the 3,206 constructed discrete locations. Similarly, the elevation is binned into one of the 27 discrete intervals according to which interval the true elevation is contained in. Since we consider only observations of *Protea* species in flower, the date can be interpreted as an observation of flowering time.

We combine the three attributes x_1, x_2 and x_3, which we regard as random variables, with a simple naive Bayes model which assumes conditional independence between them:

$$p(y_i \mid x_1, x_2, x_3) \propto p(y_i)p(x_1 \mid y_i)p(x_2 \mid y_i)p(x_3 \mid y_i), \tag{5.2}$$

where y_i is the event of the observation being species i. The conditionals $p(x_j \mid y_i)$ are straightforward implementations of the probability tables we constructed from the Protea Atlas Project in Section 3.2.

The prior $p(y_i)$ is a distribution over species before any attributes are observed. Both the attention based CNN from Section 4.3 and the subgenus CNN from Section 5.1 take image data as input and ultimately produce a 41-dimensional score vector as output. Being normalised, this score vector may be regarded as a set of probabilities indicating the likelihood of the given image being a certain species. Since both these networks carry no information of the attributes, we may regard the output to be a prior for the naive Bayes model. Effectively, this allows us to combine the attributes model with the image based classifiers.

We note that the attributes model in (5.2) can be easily altered to incorporate a subset of observed attributes (or none at all, in which case we simply return the prior $p(y_i)$). In the next chapter we experiment with a uniform prior (which gives a purely attribute based classifier), and also priors obtained from the attention based CNN network and the subgenus network. By considering the prior obtained from the image based classifiers, we are able to incorporate the image and attribute data to form our complete *Protea* identifier.

— 6 —

Experimental results

Our aim is to identify the *Protea* species in a given image with optional location, elevation and date information. In this chapter we report on the test performance of various versions of our model, that incorporate different subsets of the proposed components. All CNN classification models are trained with cross-entropy loss and the Adam optimiser [28] with its default learning rate of 0.001, unless otherwise stated. No further hyperparameter optimisation is performed on any of the networks.

6.1 Performace measures

Performance of trained models is measured in three ways: (1) top-1 accuracy, which is simply the ratio of correctly identified species over the entire test set; (2) top-3 accuracy, which is the ratio of test samples for which the correct species appeared in the model's top three scores (useful in a semi-automated, recommender-type environment); and (3) recall, which in our context is average per-class accuracy. Specifically, for recall we measure the top-1 accuracy separately for each of the 41 classes and then average the results. This weighs the classes uniformly and effectively ignores the class imbalance to give a better indication of whether rare species are correctly identified.

6.2 Image preparation

It is standard practice to augment the set of images before they are delivered to a CNN for training, by applying various image transformations to the original set.

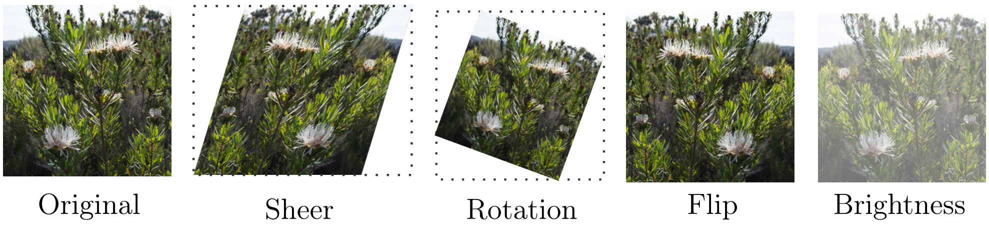


Figure 6.1: At the start of each epoch, the original training set is randomly transformed and replaced with the transformed version of itself.

Before we apply such transformations, we centrally crop and resize the training images, keeping the aspect ratio fixed since all the networks expect an input of dimension $299 \times 299 \times 3$ (a square image with three colour channels). Further, the Inception-V3 architecture expects the numerical values of the $299 \times 299 \times 3$ input to be normalised to the range $[-2, 0]$.

For each epoch of training, we then replace each image in the training set with a randomly transformed version of itself. This means that for two consecutive epochs, the network will never see the exact same data, but only transformed versions of data it has seen before. It is important to note that the total number of training images is never increased. For image transformation we perform random shears, rotations (within 15°), horizontal flips and brightness adjustments, examples of which can be seen in Figure 6.1.

6.3 Baseline image identification

As a baseline for image classification, we replaced and trained the fully-connected layers of a standard Inception-V3 network [48]. Specifically, the $8 \times 8 \times 2,048$ dimensional output of the last convolutional layer is flattened, passed through a $1,024$ dimensional fully-connected layer, whereafter dropout with a 50% drop rate is performed, as well as batch normalisation. The 41-dimensional output is obtained with a softmax layer. The convolutional layers of the pretrained network are fixed, and only the final fully-connected layers are trained. As mentioned, we use the standard cross-entropy loss, Adam optimiser with a learning rate of 0.001 and implement training set augmentation as explained in Section 6.2. Further, we implement early stopping, favouring both high accuracy and high recall on the test set. We should make it clear that this is the only place where we utilise the test set for early stopping, and we do so here only to establish an optimistic baseline against which to compare our final models. The test set does not influence the training of any further models.

Results are shown in Table 6.1, where a random classifier taking the class imbalance into account is also evaluated. Accuracy is almost double the recall, indicating that this baseline network might have a bias for the more commonly occurring species. We compare these results to those obtained in the following section, where we use the addition of an attention mechanism.

Model	Top-1	Top-3	Recall
Random classifier	6.35%	18.11%	2.44%
Baseline CNN	30.32%	59.29%	13.51%
CNN with attention (CNN-A)	55.06%	77.15%	35.59%

Table 6.1: Test performance comparison of baseline models to that of the CNN-A model. We see superior results for the CNN-A model.

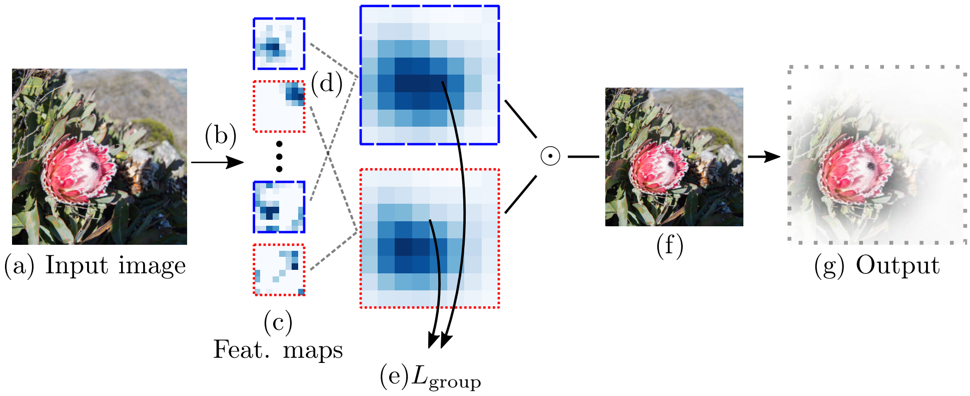


Figure 6.2: An image (a) is passed through the convolutional layers of Inception-V3 (b), yielding 2,048 feature maps (c). These are combined into attention maps through two fully-connected networks (d), learned jointly through the minimisation of a group loss (e). The two maps are added, scaled and multiplied with the original image (f), to produce an attention-boosted image (g).

6.4 CNN with attention

We now discuss the attention based CNN, which we call CNN-A, as well as the details of training the various components of this model and some qualitative results.

6.4.1 Training the CNN-A

As explained in Chapter 4, the CNN-A consists of three main components: (1) a pretrained Inception-V3 subnetwork, which extracts relevant feature maps from the images; (2) an attention subnetwork which combines the feature maps to produce two attention maps; and (3) another network based on Inception-V3, trained on the attention regions to produce a 41-dimensional softmax output. A schematic summary of the first two of these components can be seen in Figure 6.2.

For the first component, we fix all the weights of the Inception-V3 network as trained on ImageNet. We do the same for the convolutional layers of the third component. It is thus only the weights of the attention subnetwork and the final fully-connected layers of the third subnetwork which are trained. By leaving the Inception-V3 weights fixed we rely on the pretrained filters to extract features. This also provides a form of regularisation, since the pretrained Inception-V3 is trained on a very large dataset.

As mentioned in Chapter 4, the attention subnetwork is trained to reproduce attention maps M_1 and M_2, which are then further fine-tuned under a grouping loss L_{group} that favours tightness within each map and dissimilarity between them.

The loss is expressed as

$$L^{(i)}_{\text{group}} = \sum_{(x,y)} M_i(x,y)\left[(x-p_x)^2+(y-p_y)^2\right]+\lambda\sum_{(x,y)} M_i(x,y)\left[M_{3-i}(x,y)-\alpha\right], \quad (6.1)$$

for attention map M_1 and M_2, with α favouring in-map tightness and λ favouring dissimilarity between the two attention maps. These λ and α parameters are set to 2 and 0.02, respectively, as recommended in [62]. We sum the filters $M_1 + M_2$ to determine the final attention region, since both M_1 and M_2 may contain valuable information on the placement of the inflorescences in the image. We should note that we have freedom in the number of attention regions to consider. We discovered through some experimentation on the training set that one attention region did not emphasise enough image information for efficient classification. On the other hand, it was decided that three (or more) attention regions would saturate the amount of the original image which is highlighted, and we would risk losing the discriminating information provided by the attention mechanism.

It was found through informal experimentation on the training set that a learning rate of 10^{-6} produced attention regions that seem adequate, when trained for 10 epochs. In Figure 6.3 we display the regions of attention (M_1, M_2 and $M_1 + M_2$) superimposed on the original image, for a number of training epochs, for a specific example of *P. rupicola.* This example is interesting in that the image contains multiple inflorescences, yet the attention mechanism shifts focus to three of the five in the image. We also notice upon inspection of epoch 0 (a linear combination of the feature maps produced by the pretrained Inception-V3 network) that the initialisation already performs well in isolating the main *Protea* plant from the background. During training, the attention subnetwork iteratively shifts its attention to a handful of inflorescences on the plant (in this example, at least). Further examples of final attention regions ($M_1 + M_2$) may be seen in Figure 6.4.

6.4.2 Performance of the CNN-A

The CNN-A model in Table 6.1 includes the attention mechanism, and performs markedly better than the baselines in terms of top-1 accuracy and recall. An important performance metric for comparison is the recall, which we know ignores the class imbalance. We see an increase of 22.08% in recall, which indicates that the CNN-A model generalises better to our fine-grained and unbalanced dataset. However, with a recall of 35.59% there is room for improvement. We can turn to the confusion matrix[1] in Figure 6.5, where two observations stand out.

Firstly, the classes which are underrepresented in the training set are often identified as other related species, or as species for which much more training data is available. For example, we see that no images of *P. lacticolor* are correctly identified. Instead, all of them are identified to be *P. mundii* (a sister species), *P. eximia* (more dominant in the training set), *P. nitida* (similarly coloured and more dominant) or *P. repens* (more dominant). The same trend may be seen for *P. caespitosa, P. canaliculata, P. cryophila, P. grandiceps, P. laevis, P. lanceolata, P. scabra* and *P. sulphurea.*

The second observation is that species well represented in the training set (*P. neriifolia, P. nitida, P. cynaroides, P. repens*, etc.) are not often misidentified. However, other species are often misidentified for them — apparent from the dark columns in the confusion matrix in Figure 6.5. It is safe to assume that these well represented species account for the relatively large overall accuracy of 55.06%, since they dominate the total true positives across the test set. It also explains the lower recall, as a result of the underrepresented species being misclassified.

[1] We provide plots for log(1 + confusion matrix) throughout, since they give a clearer visual indication of the false positives.

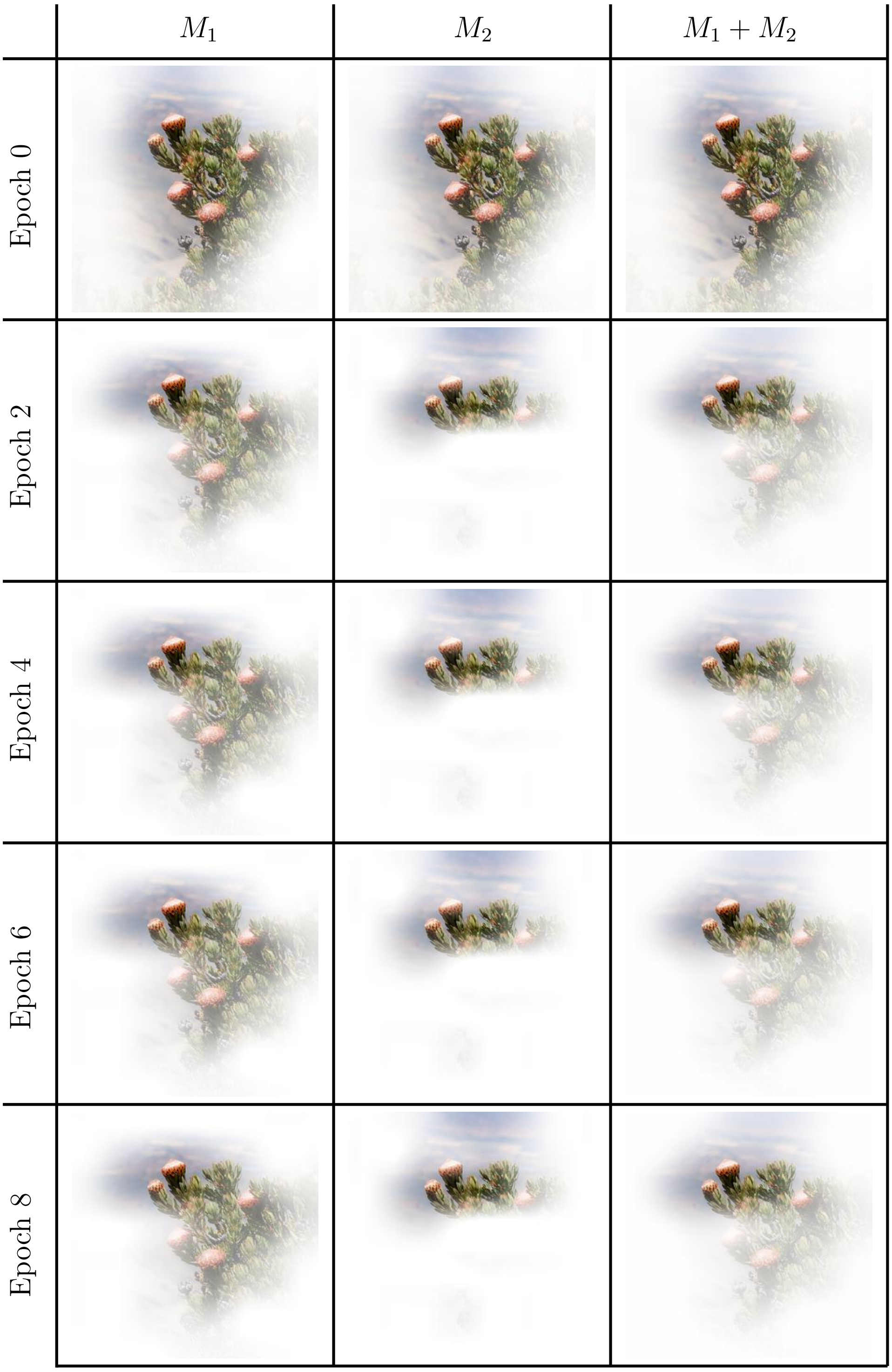


Figure 6.3: The attention subnetwork learns regions of attention through the minimisation of L_{group}. We display the M_1 and M_2 filters superimposed on the original image, in addition to the superposition of the $M_1 + M_2$ filter which is used by the classification subnetwork.

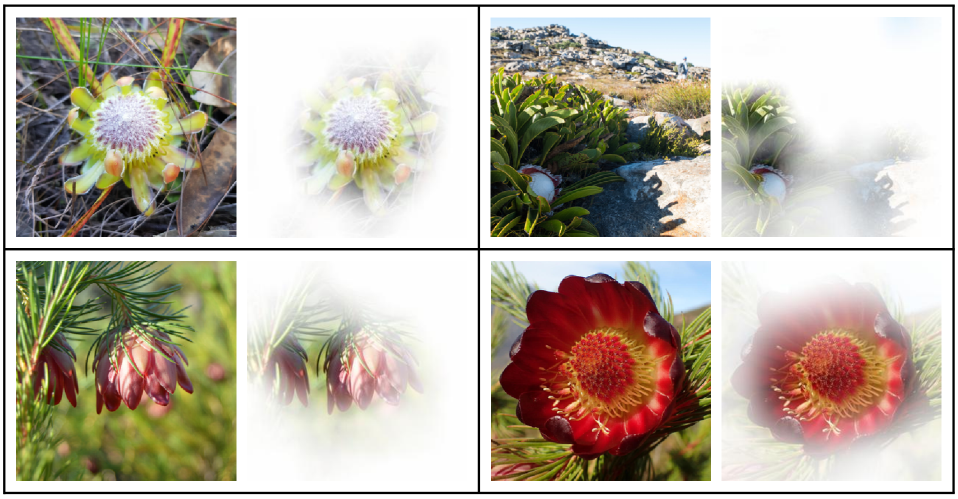


Figure 6.4: Example attention regions for sample images from the test set, as extracted from the attention subnetwork.

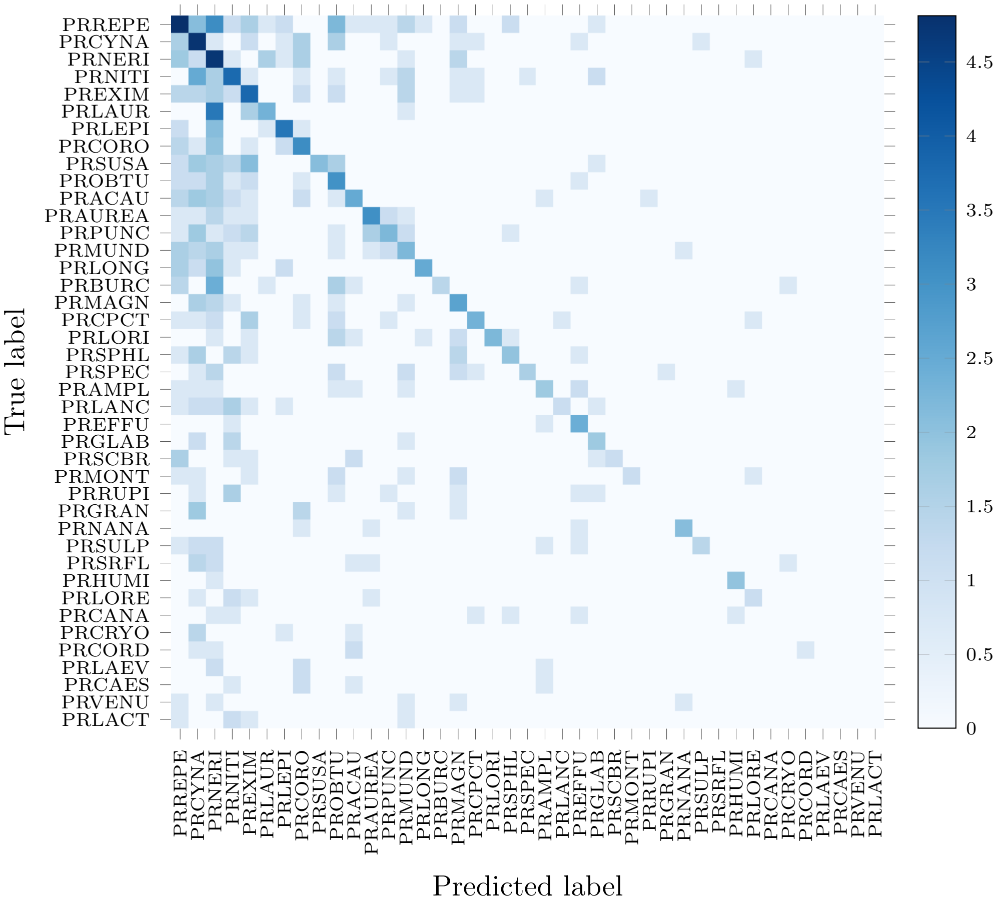


Figure 6.5: We plot the confusion matrix over the test set for the CNN-A model. Notice the general trend of accurate classification on the main diagonal, as well as the trend for common species (e.g. *Protea cynaroides*) to dominate the predicted labels.

6.5 Subgenus network

The subgenus network described in Section 5.1, which we refer to as Subg, on its own achieves a top-1 accuracy of 63.94% and a recall of 41.57%, on the task of classifying test images into the 14 subgenera. These values are not directly comparable to those in Table 6.1, since the subgenus network solves a different problem. That said, it is an easier problem and we would expect performance to be better than the baselines.

We experimented with an attention mechanism in this network as well, but found no significant change in performance. It might be due to the simplified nature of the problem, which already leads to a relatively good accuracy.

6.6 Attributes model

The attributes model described in Section 5.2 implements a naive Bayes model and takes into account location, elevation and date information. We consider two types of priors: one that is uniform (which gives a purely attribute based classifier), and also priors obtained from the CNN-A network and the subgenus network. The results for the attributes model are shown in Table 6.2, and a full discussion follows in Section 6.7.

It it worth emphasising the manner in which we incorporate the image based models (CNN-A and Subg) into the attribute model (Attr). It is done by regarding the CNN outputs as a prior to the Bayes model over the attributes, since the CNN has not seen any attributes. Although the use of the image based classifiers as a prior to the attributes model is sensible in terms of probability theory, it has the minor downside of implying that the main model is the attributes model. In practical terms, the attention network is the workhorse for the full model, as corroborated in Table 6.2 where we notice the higher recall and accuracy values for the attention network over the attributes model with a uniform prior. Intuitively we might consider the attributes to be a refinement of the attention network, and not the other way around.

Model		Top-1	Top-3	Recall
Random classifier		6.35%	18.11%	2.44%
CNN (without attention)		30.32%	59.29%	13.51%
CNN-A (with attention)		55.06%	77.15%	35.59%
Attr with uniform prior		25.86%	60.75%	34.84%
Attr with Subg prior		47.28%	76.78%	55.39%
Attr with CNN-A prior	*no Subg*	65.77%	83.35%	65.83%
CNN-A with Subg scaling	*no Attr*	56.73%	78.86%	35.43%
Attr with CNN and Subg prior	*no attention*	56.64%	80.45%	51.91%
Attr with CNN-A and Subg prior	*full model*	**70.43%**	**85.80%**	**66.88%**

Table 6.2: Test performance comparison of baseline models, naive Bayes models that use only the attributes, and various versions of our model. The best performance is achieved by our full model that combines a CNN with attention, a subgenus network, and attributes (Attr with CNN-A and Subg prior).

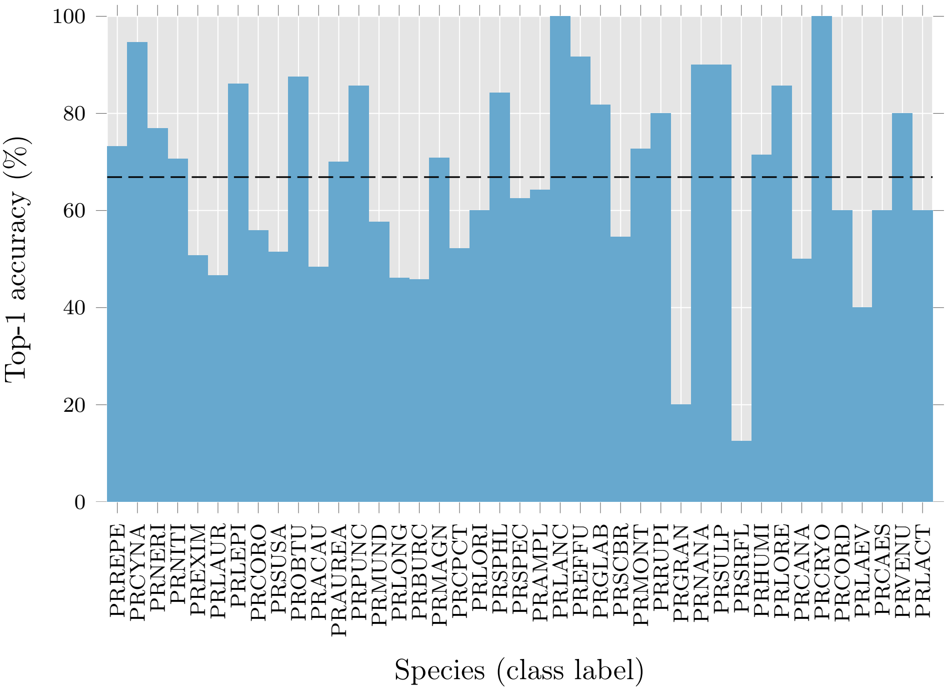


Figure 6.6: The top-1 accuracy for each species of *Protea*, as obtained on the test set using our full model. The horizontal line indicates the average per-class accuracy (which is the recall value of 66.88% in Table 6.2). We order the species along the x-axis according to the number of training images available, as is done in all previous plots.

6.7 Full *Protea* identification model

In this section we consider the combination of attributes (Attr), the subgenus network (Subg) and the attention network (CNN-A), to form our full *Protea* identification model. Results from various combinations are shown in the lower section of Table 6.2, with individual per-class accuracies given in Figure 6.6.

It may be observed that individually the attributes model and subgenus network increase the performance of CNN-A, with attributes having the greatest effect. The combination of all three components outperforms all other versions. The effect of attention on the model is also apparent when comparing the last two rows of Table 6.2. We see an increase of almost 15% in recall for the model with attention (Attr with CNN-A and Subg prior), in comparison to the model without attention (Attr with CNN and Subg prior).

We can extend the analysis of accuracy by considering the confusion matrix of the full model on the test set, as shown in Figure 6.7. In comparison to the confusion matrix for CNN-A on its own (Figure 6.5), we may highlight a number of noticeable improvements.

Firstly, we note that every class has at least a few correct identifications. Our model has improved from not being able to identify certain species at all. This fact may be emphasised by Figure 6.6, where only *P. grandiceps* and *P. scolopendriifolia* remain poorly recognised. It is interesting, considering that these two species have 32 and

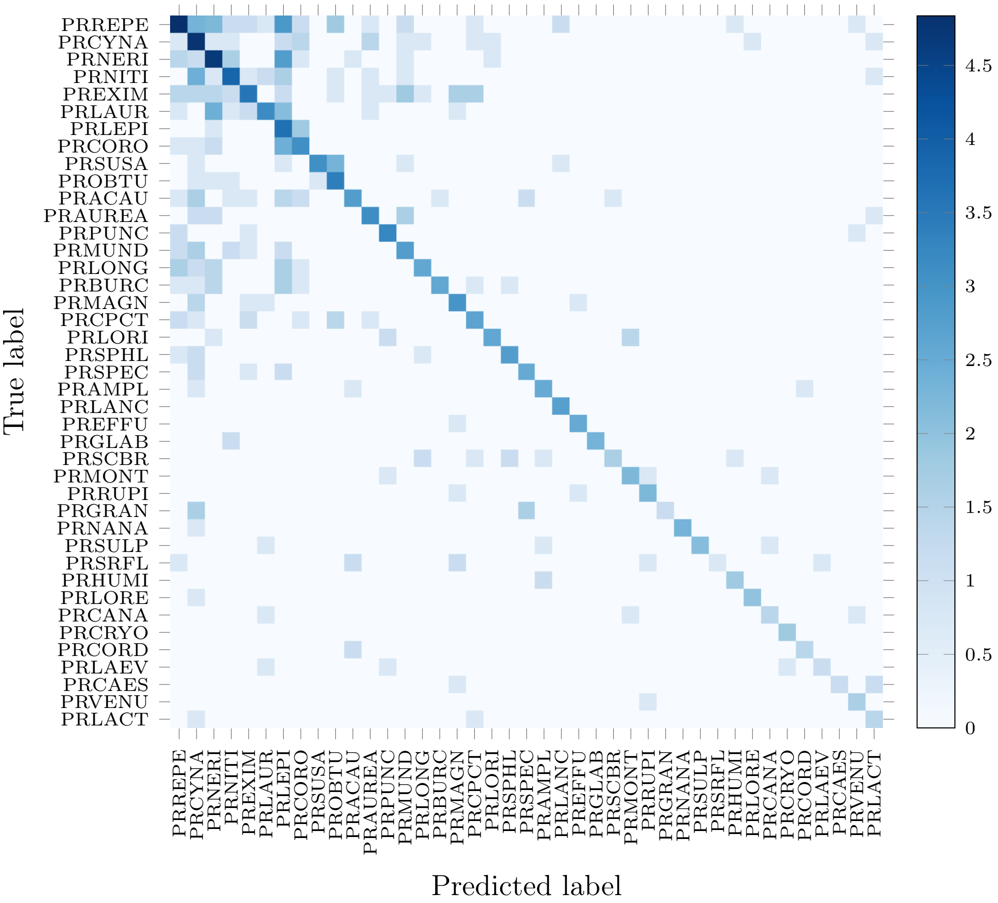


Figure 6.7: We plot the confusion matrix for the full model (Attr with CNN-A and Subg prior), which shows superior performance in identifying every *Protea* species. Notice how each class is represented by a number of correct identifications, as well as the absence of prominent columns.

27 images in their training set, respectively, which is more than what a number of other species have. Speculatively, *P. grandiceps* and *P. scolopendriifolia* both have large distributions in the CFR, and their training samples are limited. This indicates that the two species occur widely, but are not seen frequently. It may be reasonable to expect low accuracies, due to low likelihoods of being observed. A further consideration may simply be that the training images of these species are not sufficiently representative.

The second improvement in the confusion matrix can be seen in the reduction of those prominent columns we saw in Figure 6.5. The implication is that fewer species are mistakenly identified for the common species. This is emphasises by Figure 6.6 where we notice that the long tail in the training data is not carried through to the top-1 accuracies. We do still see misclassifications, where species are predicted to be the common *P. cynaroides* or *P. repens*, yet this may always be expected with long-tailed and limited training data.

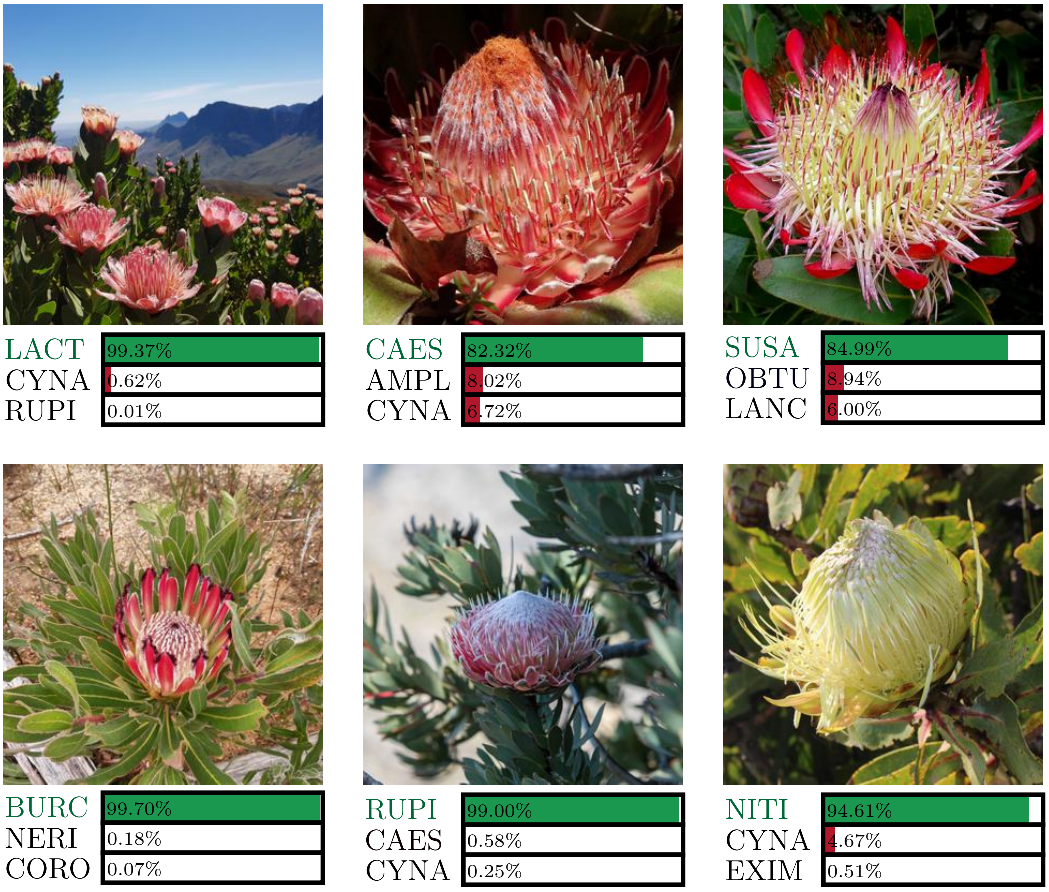


Figure 6.8: We showcase six varied examples from the test set where the full model expertly identifies the *Protea* species. In each case we show the three most probable species as predicted by the model, with corresponding output probabilities. Here, and in the following figures, green indicates the correct class and red indicates incorrect classes.

Figure 6.8 shows a number of correct identifications made by our full model on test images. For these examples the model is also highly certain of its predictions. For example, the second and third predictions for the image of *P. susannae* are species which broadly share their natural distribution with *P. susannae*. We also notice that many of the second and third choices include *P. cynaroides*, which occurs throughout the CFR and is also one of the most represented species in the training set. We should expect the model to have somewhat of a bias towards it.

In Figure 6.9 we provide six observations for which the model predicted the correct label as either the second or third most likely label. The first observation is of *P. venusta*, which occurs high in the mountains of the Swartberg. It is a range restricted species and occurs amongst *P. rupicola* (which is itself restricted by range, as indicated in Figure 6.10) and *P. punctata* (which is a sister species). Further, considering that all three of these species occur at high elevations in the Swartberg, in addition to the fact that *P. venusta* is one of the most underrepresented species in the training set, it is perhaps not surprising that the model misclassified this observation.

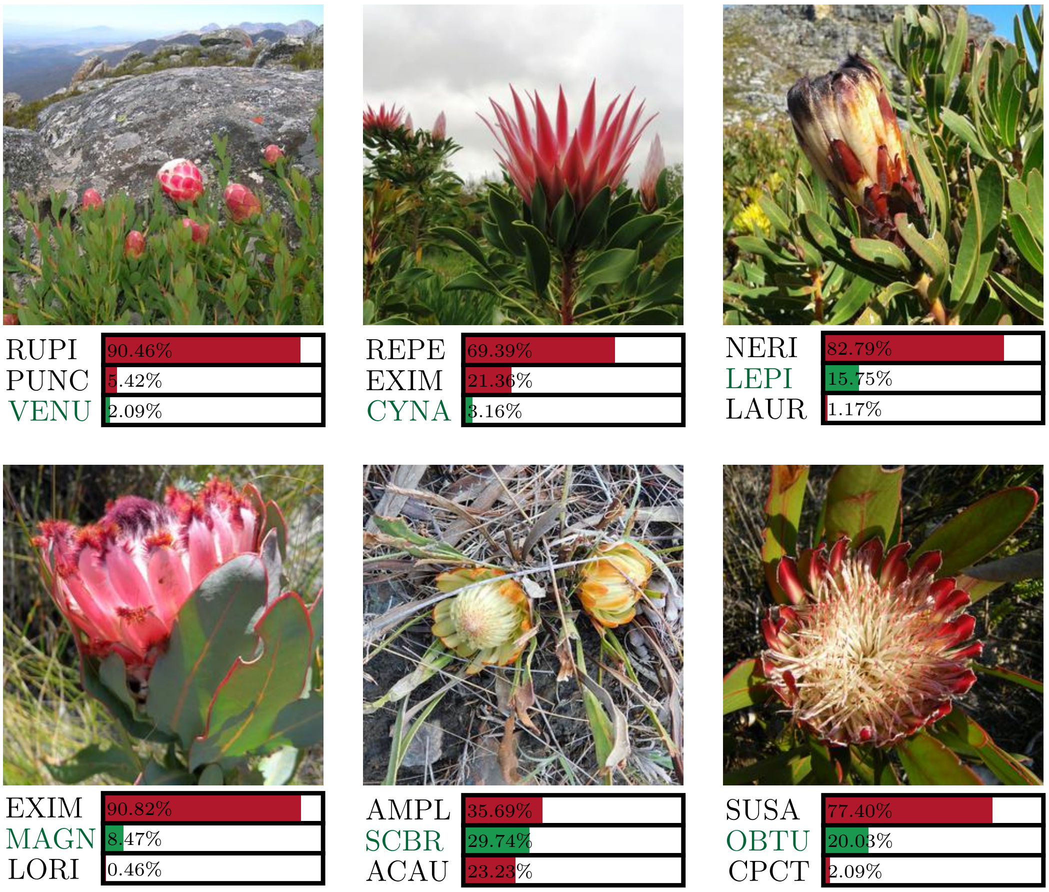


Figure 6.9: These six observations are not correctly identified by the model, but the correct labels do feature in the top three predictions.

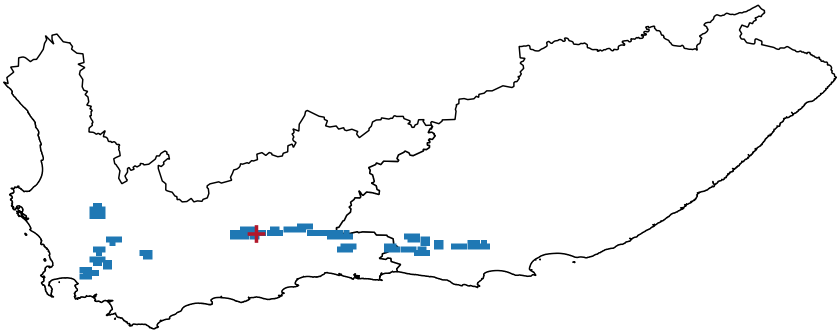


Figure 6.10: The distribution of *P. venusta* (red cross) amongst the distribution of *P. rupicola* (blue) emphasises how the identification of a rare and range restricted species is complicated by overlapping distributions with other rare species.

Another interesting observation is that of *P. lepidocarpodendron* on the top right of Figure 6.9. As illustrated in Figure 3.3, *P. lepidocarpodendron* is easily confused for *P. neriifolia* or *P. laurifolia*. One easy way to tell them apart is to consider their distributions, which are mostly disjoint. However, species such as *P. neriifolia* have invaded the natural distribution of *P. lepidocarpodendron*, which may explain the model's confusion for these two species. *P. neriifolia* is also one of the most represented species in the training set, which complicates matters further.

Ground Sugarbushes[2] present their own set of challenges, as exemplified in the bottom centre of Figure 6.9. Here an observation of *P. scabra* is mistaken to be *P. amplexicaulis*. This species is not as well represented as *P. amplexicaulis* or *P. acaulos* (the most observed Ground Sugarbush), while the inflorescence alone is quite similar to that of *P. acaluos*.

Similar trends may be observed for *P. obtusifolia* on the bottom right, *P. magnifica* on the bottom left and *P. cynaroides* at the top centre. All these species share distributions with those they are confused for, as well as sharing visual characteristics.

The significance of attributes

From a biological point of view it may be important to consider the relevance of individual attributes (and combinations thereof) for the performance of the model. Specifically, we may ask whether location, elevation or the date attribute contribute the most towards the performance of the model.

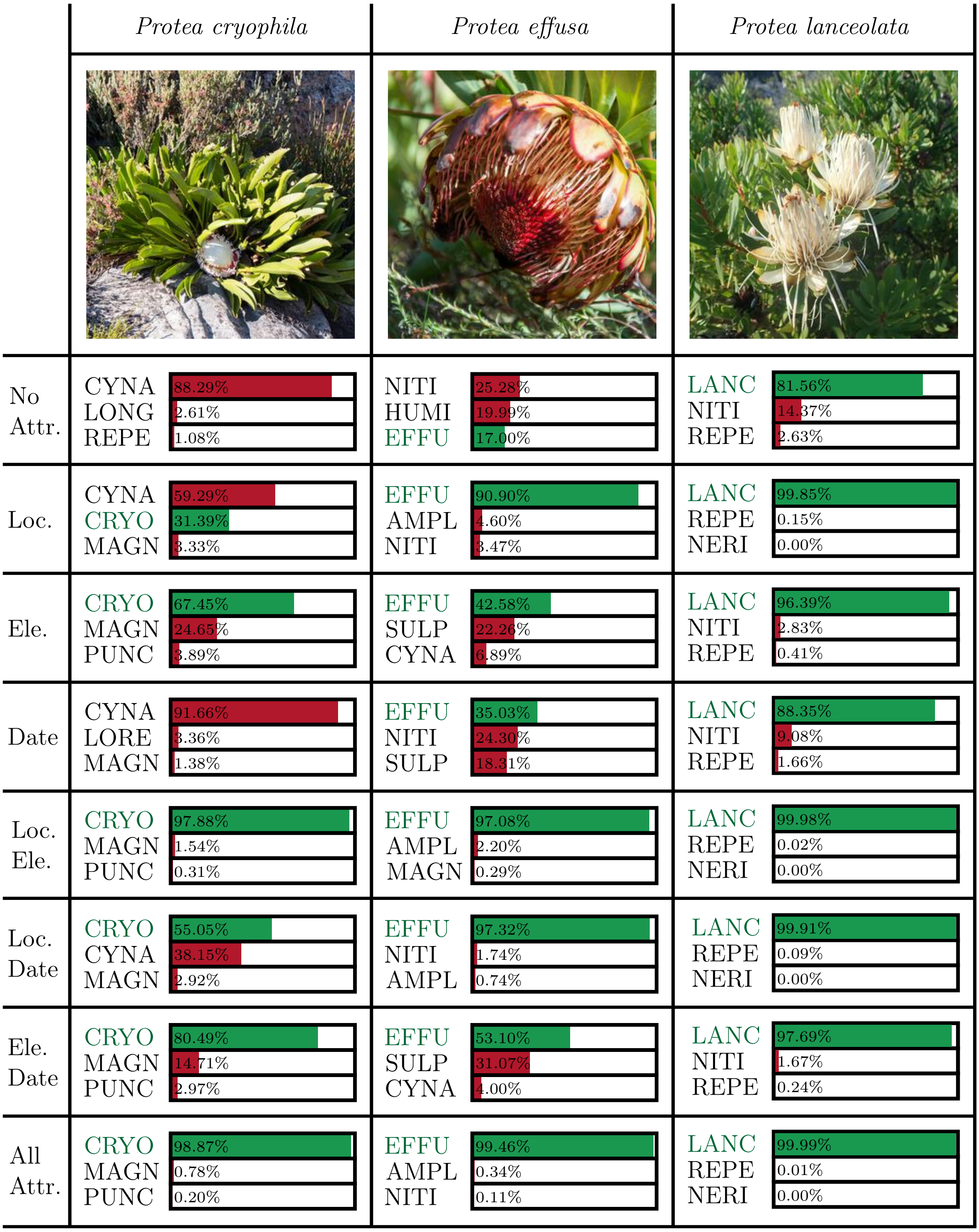
To this end, we plot the recall for our model in Figure 6.12, with a single attribute removed at a time with some probability. For each attribute, we run multiple experiments to find the average recall when a certain attribute (location, elevation or date) is removed from a percentage of the test set, at random.

With the inclusion of all three attributes our model performs with a recall of 66.88%. But, as we start to remove only the location attribute from a percentage of the test observations at random, the recall drops significantly. When we remove the location attribute from 40% of the observations, the recall has dropped by roughly 6%. In comparison, when the elevation or date attributes are excluded from 40% of the observations, we do not see a significant drop in recall at all. These two attributes have a negligible effect on the recall when compared to location.

We may have anticipated this result, if we consider the vast number of locations at which *Protea* species may be present, in comparison to the small number of elevation and date brackets. As mentioned before, there are over $3,000$ discretised locations, yet only 27 elevation and 12 date brackets.

In Figure 6.11 we examine the significance of attributes for a number of example observations. Firstly, consider the observation of the near threatened *P. cryophila*. Its closest relative outside of its subgenus group is *P. cynaroides*, for which it is confused when no attributes are included. The species are similar in appearance, so this may be expected. With the inclusion of location, the model still predicts the observation to be *P. cynaroides*, yet the true label now features in the top three re-

[2] An informal name given to those species of *Protea* which grow low on the ground.



	Protea cryophila	*Protea effusa*	*Protea lanceolata*
No Attr.	CYNA 88.29% LONG 2.61% REPE 1.08%	NITI 25.28% HUMI 19.99% EFFU 17.00%	LANC 81.56% NITI 14.37% REPE 2.63%
Loc.	CYNA 59.29% CRYO 31.39% MAGN 3.33%	EFFU 90.90% AMPL 4.60% NITI 3.47%	LANC 99.85% REPE 0.15% NERI 0.00%
Ele.	CRYO 67.45% MAGN 24.65% PUNC 3.89%	EFFU 42.58% SULP 22.26% CYNA 6.89%	LANC 96.39% NITI 2.83% REPE 0.41%
Date	CYNA 91.66% LORE 3.36% MAGN 1.38%	EFFU 35.03% NITI 24.30% SULP 18.31%	LANC 88.35% NITI 9.08% REPE 1.66%
Loc. Ele.	CRYO 97.88% MAGN 1.54% PUNC 0.31%	EFFU 97.08% AMPL 2.20% MAGN 0.29%	LANC 99.98% REPE 0.02% NERI 0.00%
Loc. Date	CRYO 55.05% CYNA 38.15% MAGN 2.92%	EFFU 97.32% NITI 1.74% AMPL 0.74%	LANC 99.91% REPE 0.09% NERI 0.00%
Ele. Date	CRYO 80.49% MAGN 14.71% PUNC 2.97%	EFFU 53.10% SULP 31.07% CYNA 4.00%	LANC 97.69% NITI 1.67% REPE 0.24%
All Attr.	CRYO 98.87% MAGN 0.78% PUNC 0.20%	EFFU 99.46% AMPL 0.34% NITI 0.11%	LANC 99.99% REPE 0.01% NERI 0.00%

Figure 6.11: Example observations to test the model's performance upon the inclusion of different combinations of attributes. In the case of no attributes, predictions are made purely from image data (the CNN-A model scaled with the subgenus network).

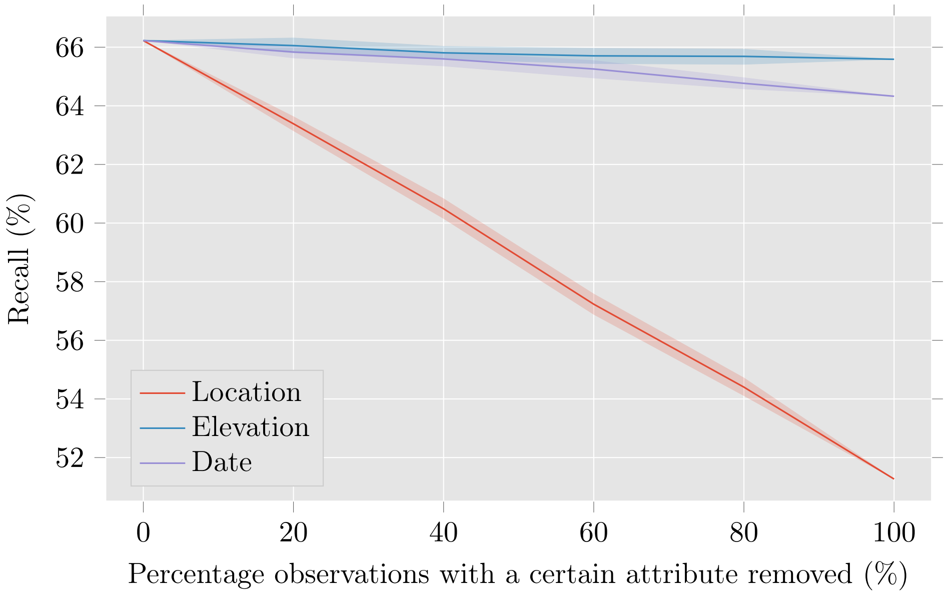


Figure 6.12: We compare the full model's performance upon the random removal of either the location, elevation or date attribute from a certain percentage of the test data. The graph shows the mean (solid lines) ± one standard deviation (shaded regions) from 30 runs.

sults with marginal certainty. With the inclusion of elevation information, the model conclusively predicts the observation to be *P. cryophila*, with *P. magnifica* featuring too. If one considers that *P. cryophila* and *P. magnifica* are both high-altitude montane species, whereas *P. cynaroides* is not, the sudden change is understandable. The date attribute is clearly not significant, even though *P. cryophila* flowers at an unusual time (during the heat of summer). This may be explained by the inclination of *P. cynaroides* to flower at any time of the year, as a result of its wide natural distribution. As expected, with the inclusion of both elevation and location, or any additional combination that includes these two attributes, the model is more successful at identification.

The second example is that of *P. effusa*. We notice that there is no conclusive identification without attributes, yet the correct label does feature within the top three. The inclusion of any attributes sways the identification towards the correct label, with location again having the greatest effect.

The third example showcases an observation where the visual system is highly convinced of the correct label. As a result, the inclusion of any attribute simply makes this identification even more conclusive.

— 7 —

Conclusion

We considered the problem of *Protea* species identification from an image and optional information specifying the location, elevation and date of the observation (which we collectively refer to as attributes). The contribution of the thesis is two-fold: we firstly introduce a challenging dataset for fine-grained image classification, and secondly propose an identification model that consists of a CNN with attention, a second CNN to classify on the coarser subgenera-level and rescale the output of the first CNN, and a probabilistic model to condition the identification on the observed attributes.

Our attempt with this thesis is to provide data and introduce a challenging problem, and we view the final *Protea* identification model as a baseline upon which future work can improve. Our aim was also to understand the effects of the various components in our model; individually and jointly. Given the specific nature of our data, we have not yet been able to meaningfully compare our approach to a completely separate one.

For the dataset we scraped the crowd-sourced website iNaturalist, and created a labelled set of 4,849 images across 41 species of *Protea*. Each of these images includes a location, elevation and date. A number of factors complicate the classification of the species from images, namely the fine granularity in the classes, the unbalanced nature of the data, and the large variation and lack of standardisation in how the images were taken.

In order to address these problems, we reviewed the literature of fine-grained image recognition and firstly implemented an attention based CNN (CNN-A) which uses a pretrained Inception-V3 network to isolate regions of attention (hopefully containing the *Protea* inflorescence) within an input image. The attention region is isolated, and a separate CNN is trained with the attention regions as input. The attention based CNN performs markedly better than a baseline CNN on the test data. The second component features a pretrained CNN which classifies an input image into one of 14 subgenera, whereafter it redistributes the output to a 41-dimensional output.

The final component features a probabilistic model which implements naive Bayes on the attribute data. We consider a uniform prior, which allows the model to act as a standalone classifier, as well as a prior inferred from the image based classifiers. By considering a prior obtained from the image based classifiers, we are able to construct the full model which combines the attribute and image data.

The proposed combination of these three parts performs reasonably well on test data, and can form a basis for future studies. Notably, the full model obtains a top-3 accuracy of 85.80% and a recall of 66.88%, managing to identify most *Protea* species to a reasonable accuracy. The model is able to counter the ill effects of the long tail distribution and has no discernible preference for well-represented species.

Although the results of the full model are promising, our implementation of the models might not be ready for deployment. With regards to the code, it is imperative that libraries are constructed to allow for modular training of the various models. Specifically, code needs to be implemented to allows for quick acquisition of data off platforms such as iNaturalist. The models need to be more flexible with regard to the number of species or taxonomic families that are considered. Importantly, the libraries need to allow for the various models to be combined with ease, and each of the models should be easy to train on their own.

Currently, the full model makes use of three separate CNNs and requires considerable computational power for training. The hope is to replace the computationally-intensive Inception-V3 networks with lightweight variants, which could allow the deployment of these models on mobile devices at remote locations in the wild.

Although the thesis focuses solely on the genus *Protea*, we need to emphasise that the work can be generalised to any taxonomic group of fauna or flora. We may examine different taxonomic levels of classification by considering species, in addition to their parent subgenera. As an example, our study could easily be raised to the family *Proteaceae* by considering genera such as *Protea, Serruria* and *Leucospermum.* This could be extended to even higher taxonomic levels.

References

[1] Aditya, K., Nityananda, J., Bangpeng, Y. and Li, F. (2011). Novel dataset for fine-grained image categorization. In: *First Workshop on Fine-Grained Visual Categorization, IEEE Conference on Computer Vision and Pattern Recognition*. Colorado Springs, CO.

[2] Apriyanti, D., Arymurthy, A. and Handoko, L. (2013). Identification of orchid species using content-based flower image retrieval. *International Conference on Computer, Control, Informatics and its Applications: "Recent Challenges in Computer, Control and Informatics"*, pp. 53–57.

[3] Barré, P., Stöver, B., Müller, K. and Steinhage, V. (2017). LeafNet: a computer vision system for automatic plant species identification. *Ecological Informatics*, vol. 40, pp. 50–56. ISSN 15749541.

[4] Beery, S., van Horn, G., Mac Aodha, O. and Perona, P. (2019). The iWildCam 2018 challenge dataset. *arXiv preprint arXiv:1904.05986*.

[5] Berg, T., Liu, J., Woo Lee, S., Alexander, M., Jacobs, D. and Belhumeur, P. (2014). BirdSnap: large-scale fine-grained visual categorization of birds. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2011–2018.

[6] Bonnet, P., Goëau, H., Hang, S., Lasseck, M., Šulc, M., Malécot, V., Jauzein, P., Melet, J.-C., You, C. and Joly, A. (2018). Plant identification: experts vs machines in the era of deep learning. In: *Multimedia Tools and Applications for Environmental & Biodiversity Informatics*, pp. 131–149. Springer.

[7] Canziani, A., Paszke, A. and Culurciello, E. (2016). An analysis of deep neural network models for practical applications. *arXiv preprint arXiv:1605.07678*.

[8] Ceballos, G., Ehrlich, P., Barnosky, A., García, A., Pringle, R. and Palmer, T. (2015). Accelerated modern human–induced species losses: entering the sixth mass extinction. *Science Advances*, vol. 1, no. 5, pp. 89–96.

[9] Chandler, M., See, L., Buesching, C., Cousins, J., Gillies, C., Kays, R., Newman, C., Pereira, H. and Tiago, P. (2017). Involving citizen scientists in biodiversity observation. In: *The GEO Handbook on Biodiversity Observation Networks*, pp. 211–237. Springer.

[10] Cho, S. (2012). Content-based structural recognition for flower image classification. *IEEE Conference on Industrial Electronics and Applications*, pp. 541–546.

[11] Cho, S. and Lim, P. (2006). A novel virus infection clustering for flower images identification. *International Conference on Pattern Recognition*, vol. 2, pp. 1038–1041.

[12] Cope, J., Corney, D., Clark, J., Remagnino, P. and Wilkin, P. (2012). Plant species identification using digital morphometrics: a review. *Expert Systems with Applications*, vol. 39, no. 8, pp. 7562–7573. ISSN 09574174.

[13] Cowling, R., Pressey, R., Rouget, M. and Lombard, A. (2003). A conservation plan for a global biodiversity hotspot — the Cape Floristic Region, South Africa. *Biological Conservation*, vol. 112, no. 1-2, pp. 191–216.

[14] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K. and Fei-Fei, L. (2009). ImageNet: a large-scale hierarchical image database. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255.

[15] Elevation-API web application (2019). `www.elevation-api.io/`. Accessed: 2019-10-05.

[16] Fu, J., Zheng, H. and Mei, T. (2017). Look closer to see better: recurrent attention convolutional neural network for fine-grained image recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4476–4484.

[17] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y. (2014). Generative adversarial nets. *Advances in Neural Information Processing Systems*, pp. 2672–2680.

[18] He, K., Zhang, X., Ren, S. and Sun, J. (2016). Deep residual learning for image recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778.

[19] Hong, S. and Choi, L. (2012). Automatic recognition of flowers through color and edge based contour detection. *International Conference on Image Processing Theory, Tools and Applications*, pp. 141–146.

[20] Hopkins, G. and Freckleton, R. (2002). Declines in the numbers of amateur and professional taxonomists: implications for conservation. In: *Animal Conservation forum*, vol. 5, pp. 245–249. Cambridge University Press.

[21] Hou, S., Feng, Y. and Wang, Z. (2017). VegFru: a domain-specific dataset for fine-grained visual categorization. *IEEE International Conference on Computer Vision*, pp. 541–549.

[22] Hsu, T., Lee, C. and Chen, L. (2011). An interactive flower image recognition system. *Multimedia Tools and Applications*, vol. 53, no. 1, pp. 53–73.

[23] Huang, R., Jin, S., Kim, J. and Hong, K. (2009). Flower image recognition using difference image entropy. *International Conference on Advances in Mobile Computing and Multimedia*, pp. 618–621.

[24] iNaturalist web application (2019). `www.inaturalist.org`. Accessed: 2019-10-05.

[25] Ioffe, S. and Szegedy, C. (2015). Batch normalization: accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

[26] Jamil, N., Hussin, N., Nordin, S. and Awang, K. (2015). Automatic plant identification: is shape the key feature? *Procedia Computer Science*, vol. 76, pp. 436–442.

[27] Jonathan, K., Michael, S., Jia, D. and Li, F. (2013). 3D object representations for fine-grained categorization. In: *IEEE Workshop on 3D Representation and Recognition*. Sydney, Australia.

[28] Kingma, D. and Ba, J. (2014). Adam: a method for stochastic optimization. *arXiv preprint arXiv:1412.6980.*

[29] Krause, J., Sapp, B., Howard, A., Zhou, H., Toshev, A., Duerig, T., Philbin, J. and Fei-Fei, L. (2016). The unreasonable effectiveness of noisy data for fine-grained recognition. *European Conference on Computer Vision*, pp. 301–320.

[30] Kress, W., Garcia-Robledo, C., Soares, J., Jacobs, D., Wilson, K., Lopez, I. and Belhumeur, P. (2018). Citizen science and climate change: mapping the range expansions of native and exotic plants with the mobile app LeafSnap. *BioScience*, vol. 68, no. 5, pp. 348–358.

[31] Krizhevsky, A., Sutskever, I. and Hinton, G. (2012). ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, pp. 1097–1105.

[32] LeCun, Y., Haffner, P., Bottou, L. and Bengio, Y. (1999). Object recognition with gradient-based learning. In: *Shape, Contour and Grouping in Computer Vision*, pp. 319–345. Springer.

[33] Lin, T., Roychowdhury, A. and Maji, S. (2015). Bilinear CNN models for fine-grained visual recognition. *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1449–1457.

[34] Maji, S., Rahtu, E., Kannala, J., Blaschko, M. and Vedaldi, A. (2013). Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151.*

[35] Najafabadi, M.M., Villanustre, F., Khoshgoftaar, T.M., Seliya, N., Wald, R. and Muharemagic, E. (2015). Deep learning applications and challenges in big data analytics. *Journal of Big Data*, vol. 2.

[36] Nilsback, M. and Zisserman, A. (2006). A visual vocabulary for flower classification. *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 1447–1454.

[37] Nilsback, M. and Zisserman, A. (2008). Automated flower classification over a large number of classes. *Indian Conference on Computer Vision, Graphics and Image Processing*, pp. 722–729.

[38] Nilsback, M. and Zisserman, A. (2010). Delving deeper into the whorl of flower segmentation. *Image and Vision Computing*, vol. 28, no. 6, pp. 1049–1062.

[39] Phyu, K., Kutics, A. and Nakagawa, A. (2012). Self-adaptive feature extraction scheme for mobile image retrieval of flowers. *8th International Conference on Signal Image Technology and Internet Based Systems*, pp. 366–373.

[40] Pl@ntNet (2019). `plantnet.org`. Accessed: 2019-10-05.

[41] Qi, W., Liu, X. and Zhao, J. (2012). Flower classification based on local and spatial visual cues. *IEEE International Conference on Computer Science and Automation Engineering*, vol. 3, pp. 670–674.

[42] Rebelo, T. (2004). Protea Atlas. `http://hdl.handle.net/20.500.12143/5287`. Accessed: 2019-07-26.

[43] Rebelo, T., Paterson-Jones, C. and Page, N. (2001). *SASOL Proteas: a field guide to the Proteas of Southern Africa.* Fernwood Press in association with the National Botanical Institute.

[44] Seek (2019). `www.inaturalist.org/pages/seek_app`. Accessed: 2019-10-05.

[45] Silvertown, J. (2009). A new dawn for citizen science. *Trends in Ecology and Evolution*, vol. 24, no. 9, pp. 467–471.

[46] Simon, M. and Rodner, E. (2015). Neural activation constellations: unsupervised part model discovery with convolutional networks. In: *IEEE International Conference on Computer Vision*, pp. 1143–1151.

[47] Sun, M., Yuan, Y., Zhou, F. and Ding, E. (2018). Multi-attention multi-class constraint for fine-grained image recognition. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, pp. 834–850.

[48] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. and Wojna, Z. (2016). Rethinking the Inception architecture for computer vision. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2818–2826.

[49] Tang, K., Paluri, M., Fei-Fei, L., Fergus, R. and Bourdev, L. (2015). Improving image classification with location context. *IEEE International Conference on Computer Vision*, pp. 1008–1016.

[50] Valente, L., Reeves, G., Schnitzler, J., Mason, I., Fay, M., Rebelo, T., Chase, M. and Barraclough, T. (2010). Diversification of the African genus *Protea* (Proteaceae) in the Cape biodiversity hotspot and beyond: equal rates in different biomes. *Evolution: International Journal of Organic Evolution*, vol. 64, no. 3, pp. 745–760.

[51] Van Horn, G., Mac Aodha, O., Song, Y., Cui, Y., Sun, C., Shepard, A., Adam, H., Perona, P. and Belongie, S. (2018). The iNaturalist species classification and detection dataset. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8769–8778.

[52] Wah, C., Branson, S., Welinder, P., Perona, P. and Belongie, S. (2011). The Caltech-UCSD Birds-200-2011 Dataset. Tech. Rep. CNS-TR-2011-001, California Institute of Technology.

[53] Wäldchen, J. and Mäder, P. (2018). *Plant Species Identification Using Computer Vision Techniques: A Systematic Literature Review*, vol. 25. Springer Netherlands. ISBN 0123456789.

[54] Wäldchen, J., Rzanny, M., Seeland, M. and Mäder, P. (2018). Automated plant species identification — trends and future directions. *PLOS Computational Biology*, vol. 14, no. 4.

[55] Wang, Z., Chen, J. and Hoi, S. (2019). Deep learning for image super-resolution: a survey. *arXiv preprint arXiv:1902.06068*.

[56] Wei, X., Wu, J. and Cui, Q. (2019). Deep learning for fine-grained image analysis: a survey. *arXiv preprint arXiv:1907.03069*.

[57] Weinstein, B. (2018). A computer vision for animal ecology. *Journal of Animal Ecology*, vol. 87, no. 3, pp. 533–545.

[58] Wu, S., Bao, F., Xu, E., Wang, Y., Chang, Y. and Xiang, Q. (2007). A leaf recognition algorithm for plant classification using probabilistic neural network. *IEEE International Symposium on Signal Processing and Information Technology*, pp. 11–16.

[59] Yanikoglu, B., Aptoula, E. and Tirkaz, C. (2014). Automatic plant identification from photographs. *Machine Vision and Applications*, vol. 25, no. 6, pp. 1369–1383.

[60] Zawbaa, H., Abbass, M., Basha, S., Hazman, M. and Hassenian, A. (2014). An automatic flower classification approach using machine learning algorithms. *International Conference on Advances in Computing, Communications and Informatics*, pp. 895–901.

[61] Zhang, C., Zhou, P., Li, C. and Liu, L. (2015). A convolutional neural network for leaves recognition using data augmentation. *IEEE International Conference on Computer and Information Technology*, pp. 2143–2150.

[62] Zheng, H., Fu, J., Mei, T. and Luo, J. (2017). Learning multi-attention convolutional neural network for fine-grained image recognition. *IEEE International Conference on Computer Vision*, pp. 5209–5217.

[63] Zhou, Y. and Chellappa, R. (1988). Computation of optical flow using a neural network. *IEEE International Conference on Neural Networks*, pp. 71–78.

[64] Zhuang, B., Liu, L., Li, Y., Shen, C. and Reid, I. (2017). Attend in groups: a weakly-supervised deep learning framework for learning from web data. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2915–2924.

Appendix

Subgenera

True sugarbushes

Protea repens
Protea lanceolata

Bearded sugarbushes

Protea coronata
Protea grandiceps
Protea laurifolia
Protea lepidocarpodendron
Protea lorifolia
Protea magnifica
Protea neriifolia
Protea speciosa

Bishop Sugarbush

Protea caespitosa

Dwarf-tufted sugarbushes

Protea lorea
Protea scabra

Eastern ground sugarbushes

Protea montana

King sugarbush

Protea cynaroides

Penduline sugarbushes

Protea effusa
Protea sulphurea

Rodent sugarbushes

Protea amplexicaulis
Protea cordata
Protea humiflora

Rose sugarbushes

Protea canaliculata
Protea nana
Protea scolymocephala

Shaving-brush sugarbushes

Protea glabra
Protea nitida
Protea rupicola

Snow sugarbushes

Protea cryophila
Protea scolopendriifolia

Spoon-bract sugarbushes

Protea burchellii
Protea compacta
Protea eximia
Protea longifolia
Protea obtusifolia
Protea susannae

Western-ground

Protea acaulos
Protea laevis

White sugarbushes

Protea aurea aurea
Protea lacticolor
Protea mundii
Protea punctata
Protea venusta

Image attributions

We include the links and licensing information for all the photographs used, which are not property of the author.

Figure 5.1, (4) from left:
Protea longifolia
(`https://www.inaturalist.org/photos/29803695`)
by Marian Oliver, licensed under CC BY-NC 4.0
(`https://creativecommons.org/licenses/by-nc/4.0/`).

Figure 5.1, (5) from left:
Protea obtusifolia
(`https://www.inaturalist.org/photos/48749864`)
by Magriet Brink, licensed under CC BY-SA 4.0
(`https://creativecommons.org/licenses/by-sa/4.0/`).

Figure 6.4, bottom right:
Protea nana
(`https://www.inaturalist.org/photos/15932993`)
by Marian Oliver, licensed under CC BY-NC 4.0
(`https://creativecommons.org/licenses/by-nc/4.0/`).

Figure 6.8, top centre:
Protea caespitosa
(`https://www.inaturalist.org/photos/24353697`)
by Marian Oliver, licensed under CC BY-NC 4.0
(`https://creativecommons.org/licenses/by-nc/4.0/`).

Figure 6.8, top right:
Protea obtusifolia
(`https://www.inaturalist.org/photos/15558398`)
by Di Turner, licensed under CC BY-NC 4.0
(`https://creativecommons.org/licenses/by-nc/4.0/`).

Figure 6.8, bottom left:
Protea burchellii
(`https://www.inaturalist.org/photos/43700859`)
by Werner Theron, licensed under CC BY-NC 4.0
(`https://creativecommons.org/licenses/by-nc/4.0/`).

Figure 6.8, bottom right:
Protea nitida
(`https://www.inaturalist.org/photos/15300239`)
by Tony Rebelo, licensed under CC BY-SA 4.0
(`https://creativecommons.org/licenses/by-sa/4.0/`).

Figure 6.9, top right:
Protea venusta
(https://www.inaturalist.org/photos/15304062)
by Nick Helme, licensed under CC BY-SA 4.0
(https://creativecommons.org/licenses/by-sa/4.0/).

Figure 6.9, top centre:
Protea cynaroides
(https://www.inaturalist.org/photos/15787209)
by linkie, licensed under CC BY 4.0
(https://creativecommons.org/licenses/by/4.0/).

Figure 6.9, bottom left:
Protea magnifica
(https://www.inaturalist.org/photos/15593236)
by Di Turner, licensed under CC BY-NC 4.0
(https://creativecommons.org/licenses/by-nc/4.0/).

Figure 6.9, bottom centre:
Protea scabra
(https://www.inaturalist.org/photos/24079931)
by Klaus Wehrlin, licensed under CC BY-NC 4.0
(https://creativecommons.org/licenses/by-nc/4.0/).

Figure 6.9, bottom right:
Protea obtusifolia
(https://www.inaturalist.org/photos/15541642)
by Andrew Massyn, licensed under CC BY-NC 4.0
(https://creativecommons.org/licenses/by-nc/4.0/).

Figure 6.11, centre:
Protea effusa
(https://www.inaturalist.org/photos/15464302)
by Magriet Brink, licensed under CC BY-SA 4.0
(https://creativecommons.org/licenses/by-sa/4.0/).

Figure 6.11, right:
Protea lanceolata
(https://www.inaturalist.org/photos/41130686)
by Di Turner, licensed under CC0 1.0
(https://creativecommons.org/publicdomain/zero/1.0/).

All other images present, which are property of the author (Peter Thompson), licenced under CC BY-NC 4.0 (https://creativecommons.org/licenses/by-nc/4.0/) include:

Figure 1.1:
Protea lacticolor, The Triplets, Jonkershoek (April 2018).

Figure 1.2:
(left) *Protea punctata*, Swartberg Pass (June 2018),
(centre left) *Protea lacticolor*, Swartboskloof, Jonkershoek (March 2019),
(centre right) *Protea cynaroides*, Slopes of Misty Point, Swellendam (December 2017),
(right) *Protea cynaroides*, Swartboskloof, Jonkershoek (March 2019).

Figure 1.4:
Protea magnifica, Haelhoeksneeukop (October 2017).

Figure 2.6:
(left) Snow on Victoria Peak, Jonkershoek (July 2017),
(right) Rain Daisies (*Dimorphotheca pluvialis*) at Postberg, West Coast National Park (August 2017).

Figure 3.1: Moving from top left, across and down.
Protea cryophila, Sneeuberg, Cederberg (January 2019),
Protea effusa, Matroosberg (July 2019),
Protea eximia, Koumashoek (December 2016),
Protea glabra, Oorlogskloof (Septermber 2018),
Protea humiflora, De Wetsberg (July 2019),
Protea lacticolor, The Triplets, Jonkershoek (March 2019),
Protea laevis, Sneeuberg, Cederberg (January 2019),
Protea nitida, Jonkershoek (June 2019),
Protea rupicola, Mast Peak, Kammanassie (December 2017),
Protea scolymocephala, Intersection of N1 and N7, Cape Town (July 2019),
Protea amplexicaulis, Matroosberg (July 2019),
Protea compacta, Luscerne farm, near Stanford (June 2018).

Figure 3.4:
(left) *Protea laurifolia*, Miaspoort (October 2017),
(centre) *Protea lepidocarpodendron*, Camps Bay (August 2018),
(right) *Protea neriifolia*, Jonkershoek (June 2019).

Figure 3.4:
(left) *Protea magnifica*, Bertsberg (August 2019),
(centre) *Protea magnifica*, Kromrivier Dome (June 2019),
(right) *Protea magnifica*, Haelhoeksneeukop (October 2017).

Figure 5.1: Moving from left.
(1) *Protea burchellii*, Paradyskloof (June 2019),
(2) *Protea compacta*, Karwyderskraal (August 2019),
(3) *Protea eximia*, Rabiesberg (October 2019),
(6) *Protea susannae*, N2 Albertiniea (June 2019).

Figure 6.1:
Protea lanceolata, Friemersheim (June 2016).

Figure 6.4:
(top left) *Protea acaulos*, Steenboksberg (July 2018),
(top right) *Protea cryophila*, Sneeuberg, Cederberg (January 2019),
(bottom left) *Protea nana*, Steenboksberg (July 2018).

Figure 6.3:
Protea rupicola, Blesberg, Groot Swartberg (December 2018).

Figure 6.8:
(top left) *Protea lacticolor* The Triplets, Jonkershoek (March 2019),
(bottom centre) *Protea rupicola* Banghoek Peak, Jonkershoek (November 2018).

Figure 6.11:
(left) *Protea cryophila*, Sneeuberg, Cederberg (January 2019).

The modified diagram of the Inception-V3 network in Figure 4.5 was taken from `https://codelabs.developers.google.com/codelabs/mlimmersion-image-flowerstxf/img/bfea25ba557fbffc.png`.