Advances in Fine-grained Visual Categorization

D.Phil Thesis

Robotics Research Group
Department of Engineering Science
University of Oxford



Yuning Chai
New College
2015

Supervised by
Prof. Andrew Zisserman and Prof. Victor Lempitsky

Yuning Chai
New College

Doctor of Philosophy
January 2015

Advances in Fine-grained Visual Categorization

Abstract

The objective of this work is to improve performance in fine-grained visual categorization (FGVC). In particular, we are interested in the large-scale classification between hundreds of different flower, bird, dog species. FGVC is challenging due to high intra-class variances caused by deformation, view angle, illumination and occlusion, and low inter-class variance since some categories only differ in detail that only experts notice. Applications include field guides, automatic image annotation, one-click shopping app and 3D reconstruction.

At the start, we discuss the importance of foreground segmentation in FGVC, where we focus on the unsupervised segmentation of image training sets into foreground and background in order to improve image classification performance. To this end, we introduce a new scalable, alternation-based algorithm for co-segmentation, *BiCoS*, which is simpler than many of its predecessors, and yet has superior performance on standard benchmark image datasets. Next, we extend BiCos to a new model, *TriCoS*, that adds a class-discriminitiveness term directly into the segmentation objective. The new term aims at removing image regions that, although appearing as foreground, do not contribute to the discrimination between classes.

We also propose a model that combines parts alignment and foreground segmentation into a unified convex framework. The model is called *Symbiotic* in that part discovery/localization is helped by segmentation and, conversely, the segmentation is helped by the detection (e.g. part layout). The joined system improves over what can be achieved with an analogous system that runs segmentation and part-localization independently.

Finally, we built a new flower dataset consisting of 26,798 high quality images collected by ourselves and 187,559 images gathered from existing datasets. The construction of this dataset follows a strict biological taxonomy. We also evaluate the impact of using the Amazon Mechanical Turk (AMT) service for filtering fine-grained data.

This thesis is submitted to the Department of Engineering Science, University of Oxford, in fulfilment of the requirements for the degree of Doctor of Philosophy. This thesis is entirely my own work, and except where otherwise stated, describes my own research.

Yuning Chai, New College

Acknowledgements

I am extremely grateful to both of my supervisors, Prof. Andrew Zisserman and Prof. Victor Lempitsky, for the excellent guidance and persistent encourangement during the past years. My DPhil would not have been possible without Andrew's enthusiasm, optimism and dedication, or Victor's endless stream of brilliant ideas. I am looking forward to being inspired by them in the future, no matter where I go.

My everyday life was greatly enriched by my wonderful labmates. First, I'd like to thank Varun, who helped me setting up both a computer and a new home when I first started in Oxford. Special thanks go to Karen, who not only likes to share his vast knowledge in computer vision, but was also there to advise me on computers, cameras and road bikes. I'd like to thank Carl, Elliot and Duncan for entertaining and challenging me in the virtual world of computer games after work, and I will always remember the fun we had at VGG dinners, especially the weirdest topics we discussed, thanks to Relja, Yusuf, Amr, Ken, Omkar, Carlos, Mircea, Esa, Andrea, Arpit, Mihn, Meelis, L'ubor, Nataraj, Eric, Matthew, Tomas and Max. I'd like to wish all the best to the new DPhil students and postdocs, Karel, Yujie, Joon, Mohsan and Hakan.

Most importantly, I want to thank my girlfriend Phebee for giving me a reason to smile everytime I come home. And of course, I'd like to dedicate this thesis to my parents, Jiayun Zhang and Qing Chai.

Contents

1 Introduction **1**

1.1 Objectives and motivation 1

1.2 Challenges 4

1.3 Contributions and thesis outline 6

1.4 Publications 7

2 Literature review **9**

2.1 The origin 10

2.2 Parts alignment 13

2.2.1 Unsupervised parts learning 13

2.2.2 Supervised parts learning 16

2.3 Foreground segmentation 19

2.4 Improvements in visual descriptors 21

2.4.1 Soft assignment and locality-constrained linear coding 21

2.4.2 Fisher vectors 22

2.5 Convolutional neural networks 23

2.5.1 Using CNN as descriptor 25

2.5.2 Fine-tuning 25

2.5.3 CNN for FGVC 26

2.6 Human-in-the-loop 27

2.7 Commercially available applications 30

2.8 Notable datasets 31

2.8.1 Oxford Flowers 32

2.8.2 Caltech UCSD Birds 32
2.8.3 ImageNet and Stanford Dogs 34

3 Classification with BiCoS **36**
3.1 Segmentation methods 40
3.1.1 Image-level segmentation: GrabCut 40
3.1.2 Class-level co-segmentation 41
3.1.3 Co-segmenting multiple class image sets 44
3.2 Other recent co-segmentation works 49
3.3 Experimental results 50
3.3.1 Oxford Flowers 17 52
3.3.2 Oxford Flowers 102 54
3.3.3 Caltech-UCSD Birds 200 55
3.3.4 Weizmann Horses and Caltech-4 56
3.4 Discussion 58

4 Classification with TriCoS **59**
4.1 Segmentation Methods 61
4.1.1 Objectiveness-based Initialization 61
4.1.2 Dataset-level Class-Discriminative Segmentation 63
4.1.3 Implementation Details 65
4.2 Experimental Results 66
4.2.1 Caltech-UCSD Birds 200-2010 68
4.2.2 Stanford Dogs 69
4.2.3 Oxford Flowers 102 72
4.3 Discussion 73

5 Classification with symbiotic model **74**
5.1 Symbiotic Segmentation and Localization 76
5.1.1 Model fitting 78
5.1.2 Learning the Model 82

5.2 Related Work 83
5.3 Experimental Results 84
5.3.1 Results and Comparisons 86
5.3.2 Experiments with Extra Annotation 91
5.4 FGVC challenge 2013 92
5.5 Conclusion 97

6 Developing a new flower dataset 98
6.1 The plant world 98
6.2 Motivations 100
6.3 Challenges 101
6.4 Data sources 104
6.4.1 Interest groups 104
6.4.2 Commissioned 106
6.4.3 Mined from existing datasets (ImageNet) 109
6.5 Experiments with AMT 110
6.5.1 Setup 111
6.5.2 Playing with annotations 113
6.6 Statistics and benchmark 116

7 Conclusion 119
7.1 Achievements 119
7.2 Future work 121
7.2.1 Models and training 121
7.2.2 Data is still the king 123

Appendices 124

A Oxford Flowers 261 125

B Oxford Flowers ImageNet 234 134

Bibliography 142

Chapter 1

Introduction

1.1 Objectives and motivation

Having machines able to visually recognize objects automatically is one of the core challenges in computer vision. The goal is to identify and elaborate what is going on in an image or video based on visual cues. In most cases, this task narrows down to classify the input into one of a set of given categories or labels, hence it is also called a classification, categorization or labeling problem.

The fine-grained visual categorization (FGVC) is a relatively new addition to the family of visual recognition. Unlike traditional categorization problems, such as discrimination between dogs and cats, or cars and bikes, FGVC focuses on the identification of the actual dog species or car models. This new task is a significant step-up from traditional works, since fine-grained or so-called sub-ordinate categories often look more alike to each other and cause more confusion during the classification. The usefulness however also stepped-up greatly. Automatically identifying a bike from a sea of cars is impressive, but any human can achieve that. But if the person is asked

which flower species an image contains, most people would need extra help, which includes asking an expert or buying field guides. Therefore, automatically identifying fine-grained categories is very useful.

In FGVC, the overarching entity is called the *domain* within which the categorization is performed. The sub-ordinate classes that are to be discriminated are the *categories*. Sometimes, domain specific names, such as *species* for flowers or birds, or *models* for car types are used instead of categories.

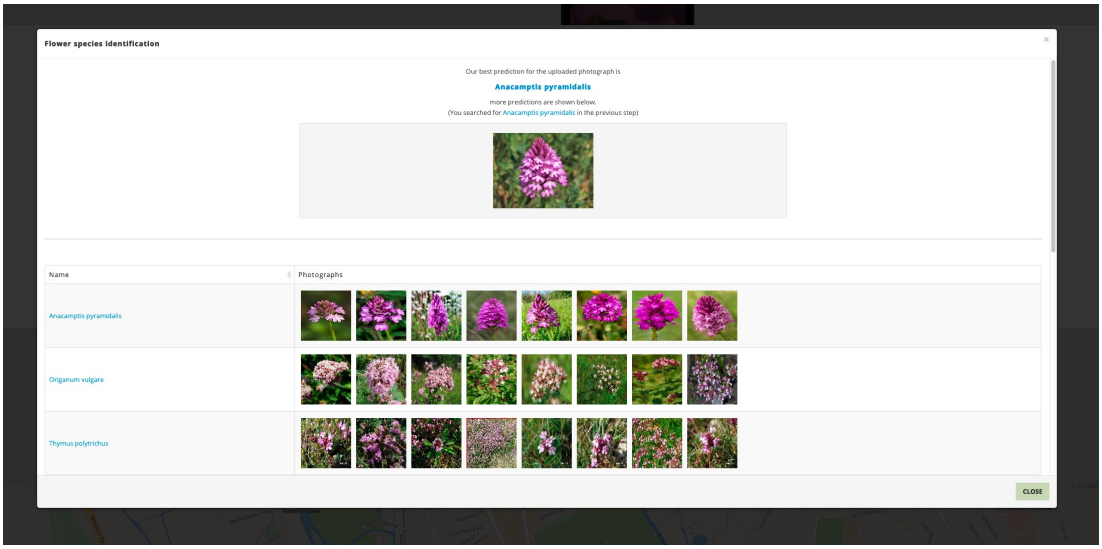
FGVC has a wide variety of application areas, some of which are listed below:

Automatic field guides. Mobile phone apps for flower, bird, dogs identification. Instead of finding an expert, all the user needs to do is to take a picture of the flower, and gets more information about it including names, where to buy and other properties right away. A demo app we developed for flower classification is shown in Fig. 1.1.

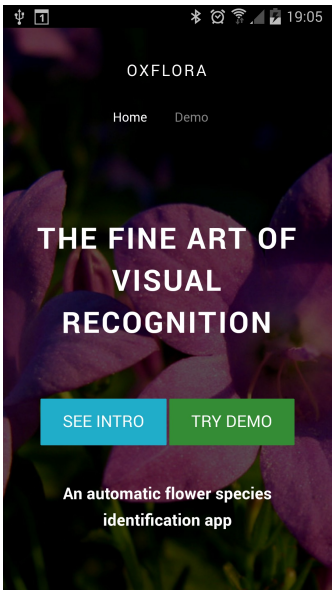
Auto-tagging for image retrieval systems. Currently, most labeled images on the web are annotated via its surrounding text on their HTML pages. Therefore, images without any text description rarely appear in search results despite being relevant. FGVC can automatically annotate fine-level labels to these images and help them appearing in search results. This does not only apply to web images, but also your personal albums. For example, Google+ [1] lets you search for keywords within your album without you having to tag your photos manually beforehand.

One-click shopping app. Similar to the field guides, shopping apps aim to bring customers faster to the item they are looking for, which directly increases the conversion rate and the revenue. The customer simply takes a photograph of an item and can purchase that item instantly. Amazon recently introduced a mobile phone that leverages

[1] plus.google.com



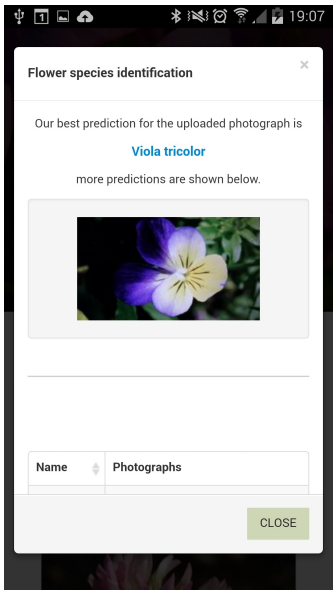
(a) Browser: classification results



(b) Phone: landing page



(c) Phone: integrated camera



(d) Phone: classification results

Figure 1.1: The flower recognition system we build using the data presented in Chap. 6. a) The photo at the top center is a query image, correctly classified as *Anacamptis pyramidalis*. Top 3 classification results with sample training images are shown at the bottom. b) The app also works on a smartphone. c) User can use the smartphone to take a flower image. d) The image is uploaded to our server and classification results are returned and shown on the smartphone.

this technology [2].

Robotics. FGVC can help robots to understand their environment more accurately. For example, a household robot would need to understand in detail what is in the fridge to make the best breakfast!

1.2 Challenges

Data collection. One of the main challenges in FGVC already starts at the data collection to train the system. As mentioned before, one beauty of FGVC is its ability to give the user expert-level answers that the non-expert user would not know. However, this also means that the training set has to contain these expert answers in the first place. Nowadays, datasets are procured by third-party services such as the Amazon Mechanical Turk (AMT) for its scalability and low cost. Images are annotated by non-expert people, who mostly are not paid very well for the effort. The alternative is to hire experts to do the annotation, but this way is not very scalable and hiring experts can be very expensive. How to collect an FGVC dataset at a low cost, high speed and high accuracy poses the first serious challenge in FGVC.

High intra-class and low inter-class variance. Many species in FGVC can only be separated by subtle details, e.g., black vs. white color on the top of a bird's head, which is easy to miss. Fig. 1.2 shows four pairs of these cases in the flowers domain, where the species of each pair are different, although the appearances are similar. Moreover, two images belonging to the same species can have different pose, view point, deformation, color, etc., making them look very different. Fig. 1.3 contains 8 images belonging to the same species, and the FGVC classifier needs to predict the same label

[2] developer.amazon.com/public/solutions/devices/fire-phone/docs/understanding-firefly



(a) Petunia (b) Silverbush (c) Water Lily (d) Lotus

(e) Globe Thistle (f) Artichoke (g) Cyclamen (h) Magnolia

Figure 1.2: Visually similar flower species. As shown in pairs, images from different flower species may even look indistinguishable for non-trained human eyes.

for all of them. High inter-class variance, in conjunction with low intra-class variance poses a very tough challenge that is intrinsic for any FGVC problems.

Less informative background. In the generic object categorization, the background of an image often contributes to the classification. Think of classifying between airplanes and cars, where the background sky or street is highly informative. However in most cases in FGVC, background is not that informative and is a source of noise. For example in the flower domain, the background often consists of dirt and leaves.

Image conditions. Finally, FGVC suffers from the same problems inherited from general visual recognition, such as illumination, clutter, occlusion. Illumination can greatly differ based on shadows, reflectance, time of the day or even time of the year. Clutter describes confusing background. This is especially the case if the target object was not the intended foreground when the image was taken. Occlusion is when only part of the object of interest is in the image. Finally, since many of the final applications for FGVC are mobile phone based, the image quality also suffers from low resolution, out of focus, blur, etc..



Figure 1.3: Visually dissimilar flowers within the same species as shown in the example of sword lily. Sword lilies have a high variation in both color and shape, which makes the recognition problem very challenging.

1.3 Contributions and thesis outline

In this section, we outline the chapters of this thesis.

Literature review. Chap. 2. We elaborate the story of FGVC and some of the important advances in FGVC, and generic visual classification that affected FGVC in a significant way. We also overview some of the existing applications that made use of the FGVC technology.

Classification with BiCoS . Chap. 3. Foreground segmentation has proven to be useful for FGVC. However, training a foreground segmenter requires accurate pixel-level foreground segmentation of a training set, which is expensive to obtain. In this work, we explored the use of *co-segmentation*, which automatically segments out the foreground area of a set of images containing the same foreground object. We introduced a new, very scalable co-segmentation method that represents the task at appropriate levels - pixels and color distributions for individual images, and super-pixels with learnable features at the level of sharing across the image set.

Classification with TriCoS. Chap. 4. Although BiCoS was designed to improve

FGVC accuracy, the fine-grained classification stage does not appear in the co-segmentation objective and hence only indirectly affects the FGVC goal. TriCoS aimed to change that by adding a class-discriminative term/level onto BiCoS. As a result, the modified co-segmentation method tries to segment out parts of the image that appear in all images and do not contribute to the discrimination.

Classification with symbiotic model. Chap. 5. Parts alignment and foreground segmentation are both important for FGVC. However, they are not independent from each other. A good segmentation can lead to better parts alignment, while well detected parts can refine the segmentation. To this end, we developed a method that does both tasks jointly, via a novel bi-convex energy minimization framework. The algorithm iterates between finding the best segmentation given the parts and locating the parts given the segmentation.

Developing a new flower dataset. Chap. 6 discusses the creation of a new, and very large flower dataset. It also contains experiments regarding using the Amazon Mechanical Turk service to filter the data for a fine-grained system. Finally, it provides some experimental baselines for future reference.

1.4 Publications

The contributions discussed in Sec. 1.3 are published as papers below:

- BiCoS: A Bi-level Co-Segmentation Method for Image Classification, *International Conference on Computer Vision*, 2011
- TriCoS: A Tri-level Class-Discriminative Co-segmentation Method for Image Classification, *European Conference on Computer Vision*, 2012

- Co-Segmentation for Fine Grained Visual Categorization, *Fine-grained Visual Categorization Workshop at CVPR*, 2013
- Symbiotic Segmentation and Part Localization for Fine-Grained Categorization, *International Conference on Computer Vision*, 2013

Chapter 2

Literature review

This chapter presents an overview of the exciting story of FGVC. In 2006, Nilsback and Zisserman (2006) introduced an object classification dataset consisting only of flower species. This problem was particularly challenging given the high intra-class and low inter-class variance nature among the flowers, which prevented older classification methods from being applied successfully. The authors gave this new branch of image classification the name "fine-grained visual categorization (FGVC)". In the recent years, FGVC has gained tremendous popularity and is now covering domains across flowers, leaves, birds, dogs and many more. There has also been commercial applications based on FGVC, which allowed people around the world to benefit from this emerging technology.

In Sec. 2.1, we will discuss the early days of FGVC, which was dominated by Bag-of-words (BoW) methods and hand-defined domain-specific strategies. Various directions then emerged towards solving FGVC and making it useful for the general public, these are concluded in Sec. 2.2 to Sec. 2.7. Finally, we give an overview of the most important FGVC datasets and challenges in Sec. 2.8.

2.1 The origin

The first FGVC work by Nilsback and Zisserman (2006) focused on flower species categorization. The authors published a dataset consisting of 17 flowers species, with each species having 80 images. Although there has been other object categorization datasets such as the Caltech256 (Griffin et al. (2007)), this new problem was highly different from any precedented ones. First, as discussed before, only people with certain domain-specific expertise could tell one species from another. Secondly, flowers are non-rigid objects and each instance of a species looks different, rendering many commonly accepted methods such as feature matching less successful. Finally, some flower species can look similar and can only be distinguished in subtle details, making parts localization and accurate foreground segmentation very important.

In 2006, the Bag-of-words (BoW) classification pipeline was widely in use. The approach first extracts hand crafted features, e.g. SIFT (Lowe (1999)), from the input image, then encodes the features via BoW and feeds the fixed-sized encoded vector into a classifier.

BoW is originally a text retrieval approach (Manning et al. (2008)), where the goal is to search for relevant documents based on keywords. In BoW, each document is represented as an N_w-dimensional histogram of word occurrences, where N_w is the number of all words in a language. The value of each bin in the histogram is equal to the number of times a word appears in the text document. Via BoW, a document containing any number of words can be encoded into a feature vector of fixed size, which is highly desirable for a number of classifiers and retrievers.

Sivic and Zisserman (2003) later adapted BoW for image retrieval and introduced BoW to other compute vision areas as well. But unlike in text retrieval, where each

text document can be broken down into words in a dictionary, there is no such natural partition for images. Sivic and Zisserman (2003) developed the concept of *visual words*, which mimicks the words in a language to create a *visual dictionary* that can be used for BoW encoding. The visual dictionary needs to be learned from a training set of images, e.g. via performing K-Means on a set of local visual features from these images. When computing the BoW representation, each local visual feature is assigned to the closest visual word in the dictionary, yielding a fix-sized representation of the entire image like in the text retrieval case.

BoW's success on images retrieval was carried over to image classification by Csurka et al. (2004). The authors combined BoW with another popular tool, the support-vector-machine (SVM) to train a system that can predict labels based on input images.

The FGVC pipeline developed by Nilsback and Zisserman (2006) used this BoW scheme in conjunction with a number of different local features, including SIFT (Lowe (1999)), HOG (Dalal and Triggs (2005)) and some hand crafted features covering the boundary and color of the flower. In their research, the authors recognized the importance of accurately segmenting out the foreground area prior to applying BoW and the classifier. They developed a flower domain specific model fitting algorithm as depicted by Fig. 2.1. The method first applies a generic color model onto the image to compute a rough segmentation. It then leverages the common knowledge that the outline of a flower is somewhat consistent. Namely, it has local maxima in the form of a petal peak and local minima where two petals intersect. Giving the rough segmentation, one can find these extrema and estimate the center of the flower. Finally, the method iterates between estimating better extrema and therefore cleaner segmentation, and re-estimating the flower center position. In the early days of FGVC, methods were developed for a specific domain only, like how this segmentation method only works

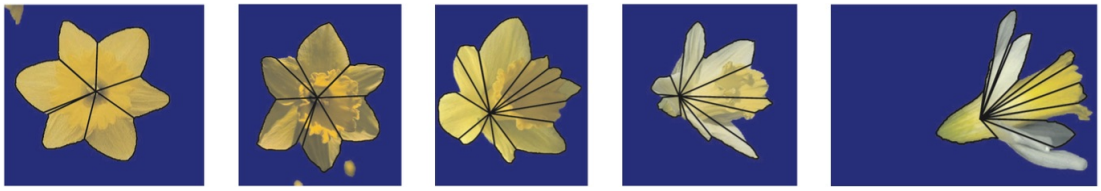


Figure 2.1: Flower petal segmentation by Nilsback (2009). It first computes a rough segmentation based on a generic color model, and then iterates between estimating the flower center and geometric fitting petal models around the center.

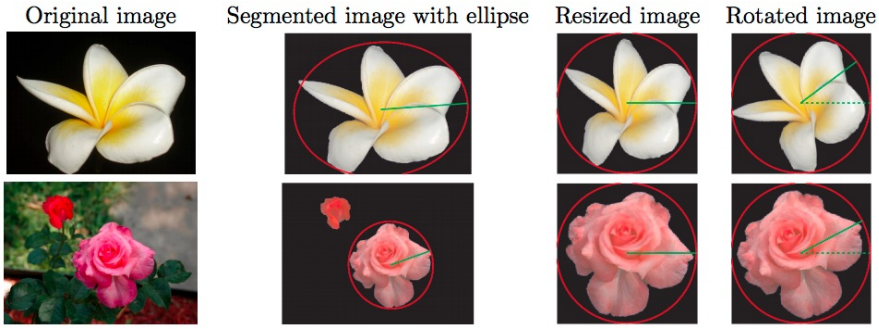


Figure 2.2: Flower alignment method by Nilsback (2009). An ellipse is first fitted to the convex hull of the segmentation, followed by resizing the ellipse to a circle with unit radius and rotation according to the tip of the petals. Aligned flowers can be compared with each other in more detail, hence improving the overall categorization accuracy.

for flowers. In the more recent times, methods are usually shown to work well on a number of different FGVC domains.

In the later works by Nilsback (2009), the authors explored techniques to accurately align the already segmented flowers prior to classification. First, an ellipse is fitted to the convex hull of the foreground segmentation. The ellipse is then transformed into a circle of unit radius and rotated according to the tip of the petals. Some sample pose alignments are shown in Fig. 2.2. Alignment, including parts alignment discussed in Sec. 2.2, has since become an integral part of FGVC.

2.2 Parts alignment

Alignment, as discussed towards the end of Sec. 2.1, is needed in FGVC since it allows detailed comparisons for discriminating between two similar classes. However, not all fine-grained domains are as geometrically consistent as flowers. A bird image, for example, has many degrees of freedom with the head, wings and feet. And the camera angle from which a bird image is taken also varies greatly, whereas the camera angle of a flower image is mostly frontal. It has become clear that global alignment methods such as the one discussed in Sec. 2.1 cannot be easily generalized over other domains.

Therefore, a range of methods emerged to only align local *parts* instead. Parts are locations that look consistent across images with the same label. The head of a bird is a good part, since it is rarely deformed and appears only in certain poses, facing left, right or frontal. Parts not only need to be consistent, but also need to be discriminative, since the ultimate classification is done using these detected parts.

In the FGVC literature, there are two types of parts learning. The supervised parts learning is based on hand-annotated keypoints in the dataset and thus learns parts that make sense to human, whereas unsupervised parts learning tries to maximize both the repeatability and discriminability of the parts automatically, without any help of hand-annotated keypoints. We will describe in more detail the unsupervised parts learning, followed by its supervised counterpart.

2.2.1 Unsupervised parts learning

One recent work in unsupervised parts learning for FGVC is that of Yang et al. (2012). The authors treated the unknown parts locations within all images as latent variables in the objective function, which optimizes for a set of detector templates. The ob-



Figure 2.3: Each rows shows a bird part learned by Yang et al. (2012) in an unsupervised way. The black boxes indicate the window with the highest part score. The method seems to have learned human understandable bird anatomies such as the head, back and tail.

jective function optimizes for fitness between the templates and training images, co-occurrence of templates and diversity between the templates. Accordingly, the training procedure operates in an iterative fashion: 1) it finds parts locations within the training data, 2) trains the detector templates based on the found parts locations, 3) updates the so-called template relation term which gives a prior probability of a set of templates co-occurring. Some of the parts learned for the birds domain is shown in Fig. 2.3. It is very surprising to see that the algorithm has learned human defined concepts such as the head, back or tail in an unsupervised manner.

During test time, all detector templates are applied on the test image. Features are then extracted from each activated part template and concatenated into a final feature vector for linear classification. The learning framework starts by initializing all templates with various sizes and evenly spaced locations within a single image.

The deformable part (DPM) model by Felzenszwalb et al. (2010) was not designed to learn parts for FGVC, but it turned out to perform pretty well in this area too. DPM was introduced in the context for object detection, in particular, it gained significant attention during the PASCAL VOC detection challenges, where the DPM and its derivative methods dominated the top spots for years. The DPM consists of a mixture of detectors, with each detector having a root HOG (Dalal and Triggs (2005)) filter, a set of parts HOG filters and a spring model for the connectivity between the part locations and the root location. The mixture enables the detector to handle multi-modal objects, for bike detection for example there is a set of filters for the frontal view of a bike and another set for the side view of a bike. The DPM detector follows a sliding window approach, which places the root filter at evenly-spaced location at various scales. For each of these root candidates, parts are placed according to the best configuration, and non-maximum suppression is applied to the candidates to find the best ones.

Zhang et al. (2013) was one of the first to benefit from the DPM for FGVC. The authors trained a DPM model for the target fine-grained domain. During test time, detected DPM parts are encoded and concatenated into a single visual descriptor, which in turn is fed into a classifier. Within the DPM model, it is natural that more than one of the part detectors will describe the same physical part. Zhang et al. (2013) used human annotations to associate these part detectors with their physical correspondences. This way, heads of a bird can be compared against each other even if they are detected by different part detectors. This pipeline is called *Weakly supervised* DPM in the illustration in Fig. 2.4. The sister pipeline *Strongly supervised* DPM essentially trains the DPM parts in a supervised way, thus avoiding the mentioned correspondence problem in the first place.

Another interesting work in unsupervised parts learning is the one by Yao et al.

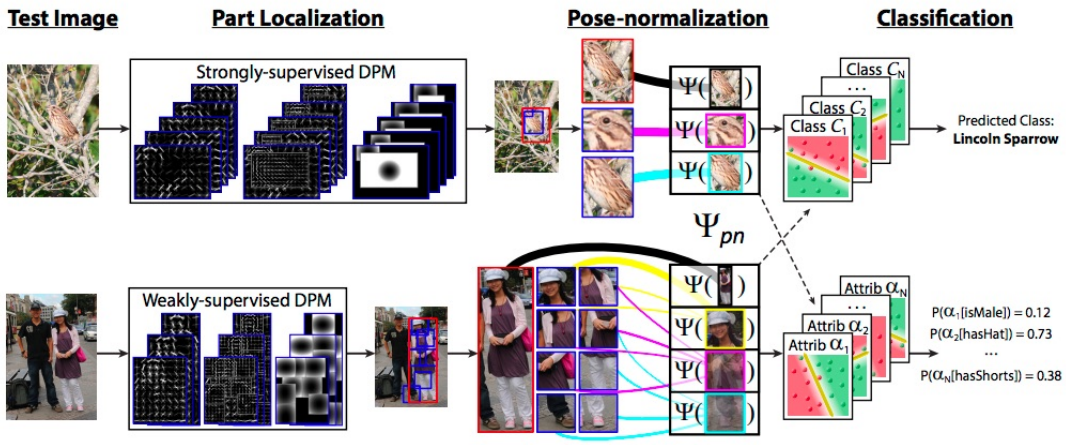


Figure 2.4: Zhang et al. (2013) applied the DPM for bird species classification. Top: the part filters are trained in a supervised way using hand-annotated part coordinates. Bottom: parts are trained without part supervision. Part activations are later fused according to the hand-annotated part coordinates.

(2011). The authors proposed a random forest-based algorithm for fine-grained classification. Unlike conventional random forests, which use the same feature vector throughout entire trees, the authors let each tree node learn the location of the most discriminative features and use those combined with some inherited information from the parent node to make the decision. During the traversal of the tree, information of different locations is pulled together which results in the final decision. One limitation of this approach is the requirement of a bounding box during test time in order to have a position normalized image.

2.2.2 Supervised parts learning

Some datasets provide 2D or even 3D keypoint locations in each training image, thus enabling supervised parts learning that leverages these additional annotations to build semantically meaningful part detectors.

The Poselet line of research Bourdev and Malik (2009) is a recent addition to this

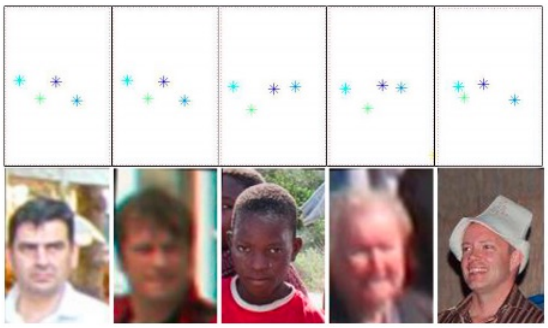


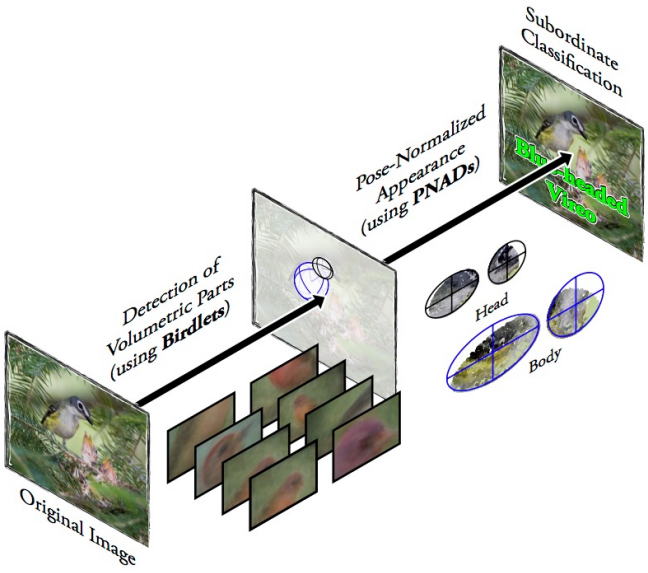
Figure 2.5: Poselets (Bourdev and Malik (2009)) are trained by grouping samples with similar keypoints configuration. Top: five Poselets with different keypoints configurations. Bottom: images assigned to each of the five Poselets configurations. Note there might be many Poselets for the same anatomical parts. In this example, all Poselets describe the human face.

framework. The original idea was to investigate extensive keypoints annotation. Given hand-annotated keypoints in a training set, the method first finds groups of patches that have similar annotated keypoints configuration in 3D, then trains a template-based detector (the Poselets) for each of the groups. See Fig. 2.5 for an example.

Poselet training relied on accurate 3D keypoints annotation, i.e. a 3D model had to be manually fitted onto the 2D image to obtain the 3D keypoints coordinates. These annotations are very expensive, therefore, other works opted for a sparse 3D annotation or operated directly in the 2D space image space instead.

In the bird species categorization work of Farrell et al. (2011), the authors approximated the bird with two ellipsoids, one for the head and one for the body. Just as in the Poselet, part detectors (Birdlets) were trained based on groups of patches displaying similar pose, which is determined by hand-annotated ellipsoids in the training data. During test time, all detectors were applied to the test image and reliably recovered the ellipsoids of the head and the body. Features were then extracted from these aligned parts and fed into a classifier. See Fig. 2.6 for an overview of the FGVC pipeline.

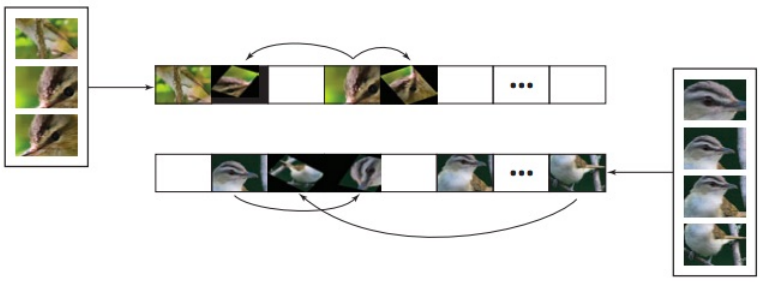
In a follow-up work of Birdlets, Zhang et al. (2012) explored ways to directly use



.

Figure 2.6: Overview of the Birdlet work for birds species categorization Farrell et al. (2011). First, part detectors are applied to the test image to recover the pose (represented as ellipsoids for head and body), features are then extracted from aligned head and body for classification.

Poselet activations for classification. The common way to use parts for classification is to extract features from each activations and stacked them together to obtain a descriptor for the whole image. However, since only a few Poselets are activated per image, the concatenated descriptor is sparse which results in low classification accuracy. Two methods were developed to make the descriptor denser (see Fig. 2.7). The first method is to mock missing parts by warping detected activations, since the pose relationship between two Poselets are known. The second method pools Poselets that describe the same bird locations together, basically reducing the dimensionality of the image descriptor by merging dimensions in a supervised way.



(a) Warp feature kernel



(b) Pooling feature kernel

Figure 2.7: Pooling kernels developed by Zhang et al. (2012) for birds species classification. After Poselets (Birdlets) were applied to an input image, features are extracted per Birdlets and concatenated, yielding a sparse image descriptor. Left: the warp feature kernel tries to mock non-activated Birdlets by apply warping operations on neighboring activations. Right: the pooling feature kernel aggregates Birdlets activations that share the same physical location on a bird.

2.3 Foreground segmentation

Nilsback and Zisserman (2008) showed great improvement in classification accuracy by removing the background of images, since the background appearance is often shared across different fine-grained categories and therefore only serves as a source of confusion.

Later foreground segmentation works formulated the segmentation task into an energy minimization framework and solve it using the famous graph-cut by Boykov and

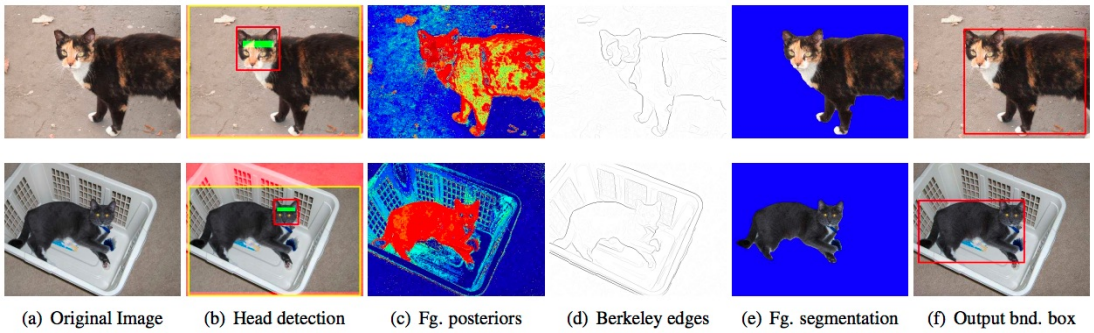


Figure 2.8: Cats and dogs foreground segmentation pipeline by Parkhi et al. (2011). The face is first detected using an DPM Felzenszwalb et al. (2010). The color information of the face is then used as foreground color prior to segment the entire body via a graph-cut framework.

Jolly (2001). Graph-cut is an optimization procedure that solves for the binary segmentation of a weighted graph, with each vertex having a prior probability of being fore- or background and edges indicating whether two connected vertices should have the same label (binary potentials). Graph-cut is widely used in foreground segmentation, semantic segmentation, interactive segmentation, etc..

The work by Parkhi et al. (2011) was an example of using graph-cut for foreground segmentation in conjunction with dog/cat species categorization. The pipeline is depicted in Fig. 2.8. The head of an animal is first detected by a DPM, which has a high chance of success, since the face is hardly deformable. A color model of the face is then propagated through the image, and the output is a map indicating the prior probability of a pixel belonging to fore- or background, which can be used as the unary potentials for graph-cut. The binary potentials are obtained by performing edge detection. Outputs of the foreground segmentation and the induced bounding boxes are shown in the two columns to the right.

2.4 Improvements in visual descriptors

Computing a visual descriptor for an image or a patch often consists of two steps, a local feature extraction step and a feature encoding/pooling step. The former extracts a variable number of local features from either densely or sparsely sampled locations from the input image. The latter then combines these local features into a vector of fixed size so that can be used by a common classifier such as the SVM or a random forest. BoW is the most basic encoding/pooling method.

The area of local features has been dominated by gradient-based descriptors until the rise of convolutional neural nets. These features include the most commonly known SIFT (Lowe (1999)), SURF (Bay et al. (2006)) and HOG (Dalal and Triggs (2005)). There has been new descriptors in the recent years, e.g. Bo et al. (2010), but these do not seem to outperform older ones greatly and consistently.

Stronger improvements could be achieved in the encoding/pooling stage, which will be discussed in this section. There is an extensive review of these encoding methods by Chatfield et al. (2011).

2.4.1 Soft assignment and locality-constrained linear coding

Given a visual dictionary, BoW assigns each feature vector to the dictionary word that is closest to it. This type of *hard* assignment does not take the actual distance between the local features and visual words into account. Moreover, it can be that one feature vector lies in the center of a few words but is randomly assigned to one of them, making the encoding more noise-prone.

The Locality-constrained linear coding (LLC) scheme, developed by Wang et al. (2010), overcomes these weaknesses by *soft* assigning feature vectors to dictionary

words. LLC uses the same setting as BoW. It takes a pre-trained visual dictionary containing K N-dimensional words, a set of feature vectors of the same dimensions, and produces encoded feature vector of fixed K dimensions. Instead of assigning one feature vector to a single visual word, LLC tries to linearly reconstruct the feature vector using the dictionary entries. The reconstruction coefficients, which are floating point values normalized between 0 and 1, then correspond to the encoded values.

After each of the set of local feature vectors has been processed, the encoded vectors need to be pooled into a single feature vector. The most straightforward pooling methods are average and maximum pooling, which are element-wise sum and max operations along the encoded vectors. In reality, for the sake of speed, a feature vector is constructed using its nearest M neighbors instead of using all visual words. BoW can be viewed as an extreme case of LLC, namely when $M = 1$, where the encoded vector is a zero vector with a single 1.

2.4.2 Fisher vectors

Fisher vectors, by Jaakkola and Haussler (1998), and applied to visual search by Perronnin and Dance (2007), further improved the encoding scheme by adding the first and second derivatives of the feature vector towards each visual words.

In this scheme, each local descriptor is assumed to be an *iid* sample from a mixture of Gaussian (GMM). The Fisher vector is constructed by taking the log-likelihood with respect to the GMM's mixture weights, means and variances. This is a tremendous gain in information contained in the encoding compared to BoW and LLC, which is also reflected in the dimension of the encoded feature vector. Given a GMM with K N-dimensional components, the output dimension of Fisher encoding is $(2 \times N + 1) \times K$, a significant increase from K in BoW and LLC. Practically, much less components are

used in a Fisher GMM than in a visual words dictionary to compensate the higher dimension produced by Fisher. Moreover, it was shown in Perronnin et al. (2010) that the derivatives with respect to the mixture weights does not contribute greatly towards better classification performance.

A number of improvements have been proposed by Perronnin et al. (2010), e.g. power normalization. In most cases, power normalization simply takes the signed square root from the encoded vector, which makes encoded vectors more comparable to each other. Taking the square root of local features such as SIFT prior to the encoding, as discovered by Arandjelović and Zisserman (2012), also improves the Fisher vector in many cases.

2.5 Convolutional neural networks

Convolutional neural networks (LeCun et al. (1998)) belong to the family of deep learning, which maps a set of observations (the input) to a set of targets (the output) via multiple non-linear transformations. Other types of deep networks have also been successfully applied to computer vision, these include the Long short-term memory network (LSTM, Gers (2001)), the recurrent neural network (RNN) and the deep belief network (DBN). The CNN is a feed-forward network, where none of its units forms a directed cycle.

The CNN gained its name from the convolutional layers which form the first layers of the network. In computer vision, a convolutional layer is designed to take a 3-dimensional input ($W \times Z \times D$) and applies a 2-D convolution along first two dimensions with K convolution kernels of size ($A \times B \times D$). The K convolutions are then stacked together to form the final output of size $X \times Y \times K$. This type of layer

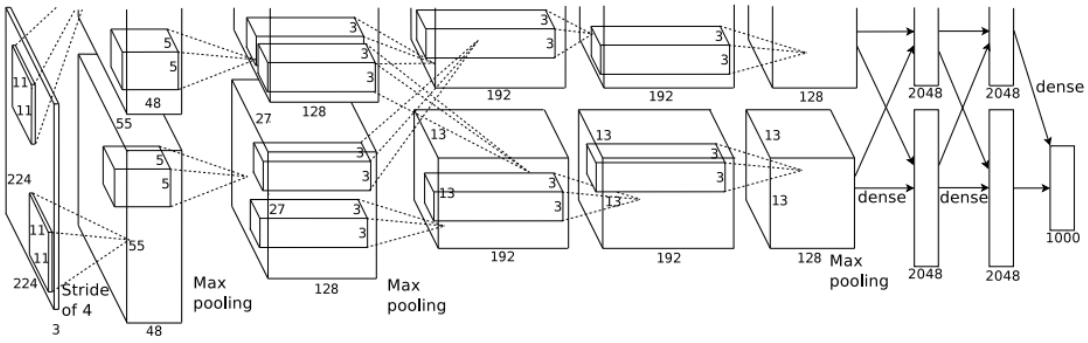


Figure 2.9: The convolutional neural network trained by Krizhevsky et al. (2012) for ImageNet Challenge 2012. It involves of 12 layers in total and 8 layers with trainable parameters. These are 5 convolutional layers and 3 fully connected layers on top of the convolutional ones. Parameters of the middle layers are split in half due to GPU memory constraints. This trick is no longer necessary with more modern GPUs.

mimicks the principle of many traditional local feature extractors, which often applies a hand-defined convolution kernel to the input image to compute local gradients. In the case of a network, these kernels are learned based on training data instead.

CNNs have a long history in machine learning and computer vision, for example, it has been the top performing method in hand-written digits recognition (LeCun et al. (1998)) for years. However, limited computing resource and a lack of large datasets had prevented neural nets being successfully deployed in larger scale object categorizations and the FGVC. It was not until recently that Krizhevsky et al. (2012) successfully trained a CNN for the ImageNet Challenge 2012 thanks to advances in parallel computing (GPU). The authors managing to reduce the top 5 1000-way classification error from 26.1% to 15.3% over to all other state-of-the-art techniques, and the CNN instantly became the center of attention in the computer vision community. The network is depicted in Fig. 2.9.

2.5.1 Using CNN as descriptor

Despite using modern GPUs, training a full CNN from scratch for a large scale problem can take a week or two, which is too slow for many research areas. Moreover, training a CNN of a similar size as in Krizhevsky et al. (2012) requires a tremendous amount of training data, which is not available for many tasks especially for FGVC, where data collection poses a challenge itself.

Several works have experimented with using the outputs of a pre-trained CNN as an image/patch descriptor for a new task other than the data the CNN was trained on. It is assumed that the bottom layers, especially the convolutional layers, correspond to generic image representations while the top layers are task-specific. Therefore, one can simply take the output vector of a middle layer as an image descriptor. This technique is extremely useful for image retrieval, as shown in Babenko et al. (2014), Chatfield et al. (2014), but less useful for image categorization, since it is sub-optimal to the fine-tuning discussed in Sec. 2.5.2.

2.5.2 Fine-tuning

When adapting a network for classification purposes, instead of taking the output of a mid-layer as image descriptor, one can *fine-tune* the network for the new task.

Several works have experimented with fine-tuning a CNN. Although they vary to a certain extent, the underlying principle remains the same as in Sec. 2.5.1, namely, one assumes that the bottom layers, especially the convolutional layers, correspond to generic image representations while the top layers are task-specific. Girshick et al. (2013) fine-tuned the 1000-way ImageNet network for a 21-way PASCAL VOC detection network. The authors first removed the last layer of the network, replaced it

with a new fully connected layer of 21 output elements. Finally, they kept the training the network using the PASCAL VOC data while reducing the overall learning rate of the network by $\frac{1}{10}$. Oquab et al. (2014) deployed a very similar strategy for the PASCAL VOC classification challenge, replacing the final layer with two fully-connected layers.

2.5.3 CNN for FGVC

The work by Zhang et al. (2014), shown in Fig. 2.10, was one of the first to apply CNNs to FGVC. Following the development of R-CNN for detection by Girshick et al. (2013), the idea is to start from interest region proposals (e.g. Uijlings et al. (2009)) and train a CNN-based binary classifier to compute the parts activation for each region in a test image. The authors fine-tuned an ImageNet-based CNN for birds classification, of which the penultimate layer output is used as part descriptors. These CNN-based features were then applied for classifying interest region proposals. Activated region proposals are then concatenated as the final image descriptor for the final fine-grained classification.

Another recent work of using CNN on FGVC is the one by Branson et al. (2014), shown in Fig. 2.11. The authors investigated the invariances of the CNN by comparing the baseline results against results where the parts were pre-warped prior to the classification. The warping was performed using the keypoint locations estimated by a DPM. More importantly, the paper showed extensive experimental results on varying different parameters when using the CNN for the bird classification problem, e.g. different ways to train the CNN and the output of which layer is best to use as the visual descriptor. According to their research, fine-tuning an ImageNet-based network significantly improves over training from scratch due to limited FGVC training data

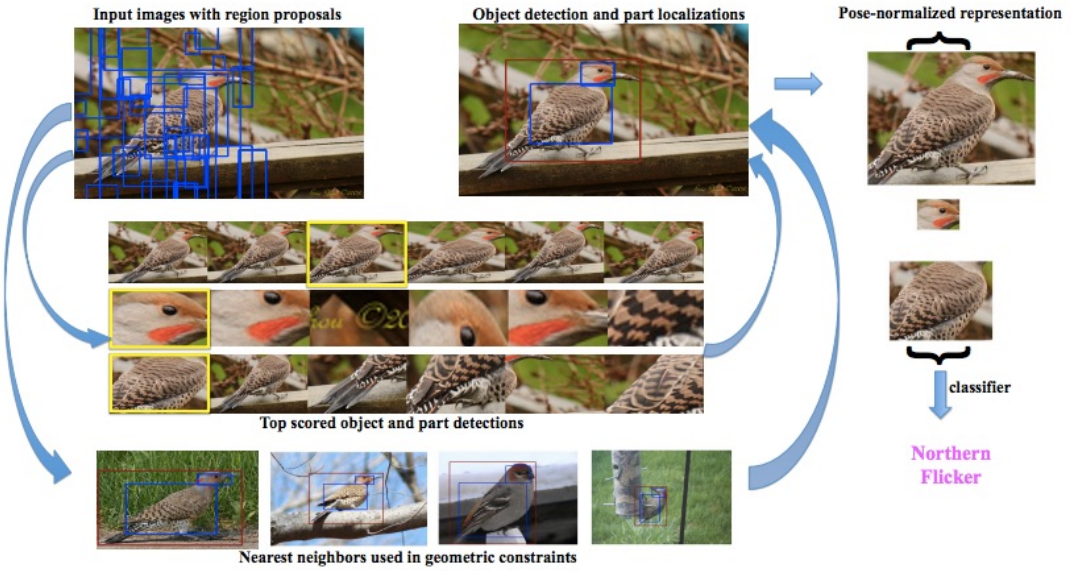


Figure 2.10: Overview of the R-CNN-based parts detection framework by Zhang et al. (2014). Interest regions proposals are first computed, which are encoded into CNN-based descriptors. Those descriptors are used for both parts detection and the final fine-grained classification. Geometric constraints were also included in the parts detection. However, the impact of these constraints is relatively small.

size. And layers 6 and 7 are the best to use as visual descriptors.

2.6 Human-in-the-loop

One of the target application in FGVC is the development of field guides to automatically recognize animals or plants. However, for challenging tasks such as discriminating hundreds of birds classes, the technology still cannot meet consumers' expectations. Therefore, Branson et al. (2010) started to explore the possibility of active classification, or so-called human-in-the-loop classification.

The research human-in-the-loop tries to achieve a high accuracy with as little human involvement as possible. In the first work by Branson et al. (2010), the interaction between the computer vision system and the user is based on pre-defined questions. Upon

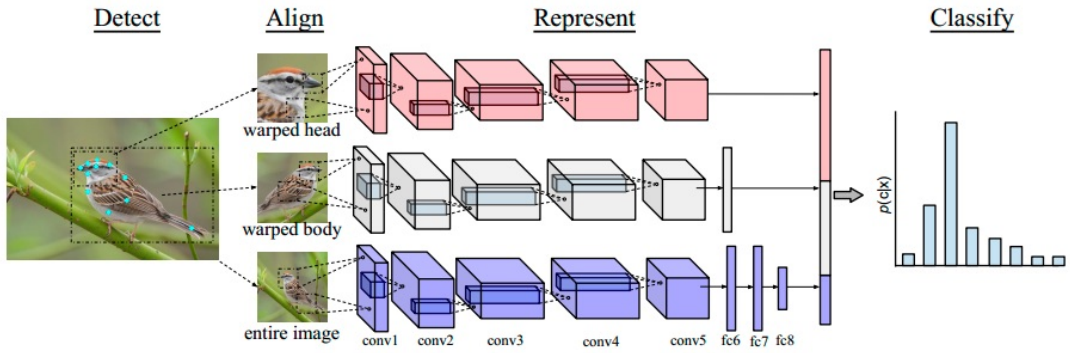


Figure 2.11: In the work by Branson et al. (2014), the authors first warped part patches according to keypoint detections performed by a DPM. Parts are then individually encoded into CNN-based descriptor, which are then used to classify between bird species.

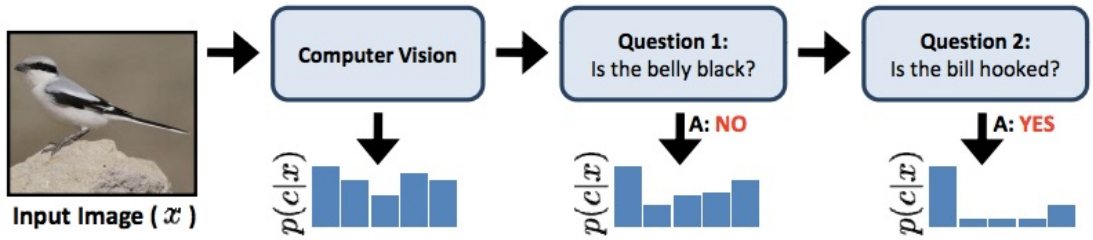


Figure 2.12: Human-in-the-loop pipeline (figure extracted from Branson et al. (2010)). Computer vision is used for generating an initial probability distribution, which is later refined by questions answered by the human.

receiving an input image, the system estimates an initial guess of the target species and selects the question from the pre-defined list of which the user submitted answer could reduce the entropy the most. When the answer is delivered, the initial guess is refined and a new question is selected. This procedure is repeated until saturation. Experiments showed that the top-1 classification accuracy is increased from 20% to 70% after in average 40 binary questions have been answered. Some sample questions are shown in Fig. 2.12.

As discussed in Sec. 2.2, parts localization and alignment is essential to successfully performing FGVC. In a follow-up work of Branson et al. (2010), Wah et al. (2011) extended the human-in-the-loop pipeline by adding localization-based questions to the

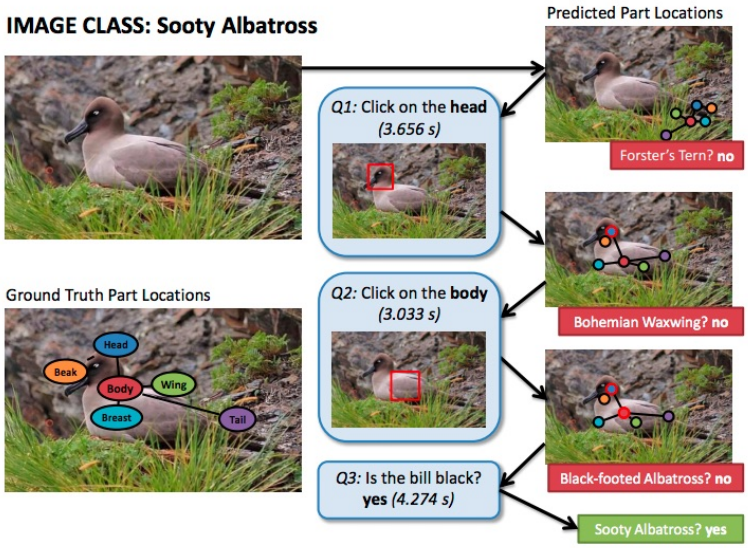


Figure 2.13: Updated human-in-the-loop pipeline by Wah et al. (2011). The authors added localization-based questions alongside binary questions developed by Branson et al. (2010).

pre-defined question pool as shown in Fig. 2.13. The user is asked to, for example, click on the head or the tail of the bird. Parts localization is supported by adding the bird skeleton locations as a latent-variable alongside the class probability distribution in the computer vision system. Another improvement over the earlier work was the formulation of response time into the optimization target. For example, a question that is easier to answer, can now be prioritized over a question that gives better entropy reduction but takes longer to answer.

Wah et al. (2014) is the most recent addition to the human-in-the-loop line of the work. In this work, the user is asked to select the most similar images to the test image from a pool of candidates. This way, the user does not need to understand bird-domain specific questions as in Branson et al. (2010) and only needs to compare images against each other. The graphic interface is shown in Fig. 2.14. A similarity matrix between all competing classes is first learned via a set of pre-collected answers. During test time, after an initial guess is estimated using computer vision only, the similarity matrix

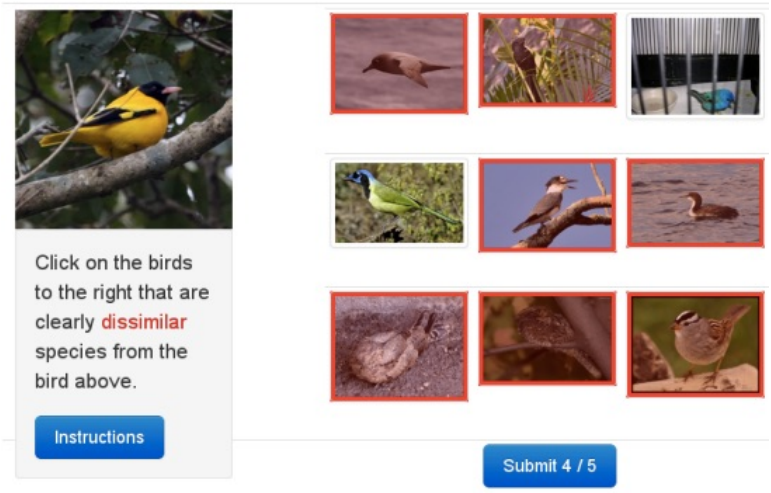


Figure 2.14: Distance metric-based human-in-the-loop pipeline (figure extracted from Wah et al. (2014)). Computer vision is again used for generating an initial probability distribution. However, the human is asked to find the similar images to the test image (left) from a pool of images selected from a database.

helps to select images from the classes that have more or less the same chance of being selected by the user, which would reduce the entropy the most. The similarity matrix is also used to refine the initial guess.

2.7 Commercially available applications

Since the goal of FGVC is to help the greater community to identify objects that they would not be able to identify otherwise, it is not surprising that there are already the first applications, e.g. mobile apps, out there to leverage the current FGVC technologies.

One group that stands out is the joint effort from Columbia University and University of Maryland, who developed Leafsnap [1], Dogsnap [2] and Birdsnap [3], each of which has

[1] http://www.leafsnap.com Kumar et al. (2012)
[2] https://itunes.apple.com/us/app/dogsnap
[3] http://www.birdsnap.com Berg et al. (2014)

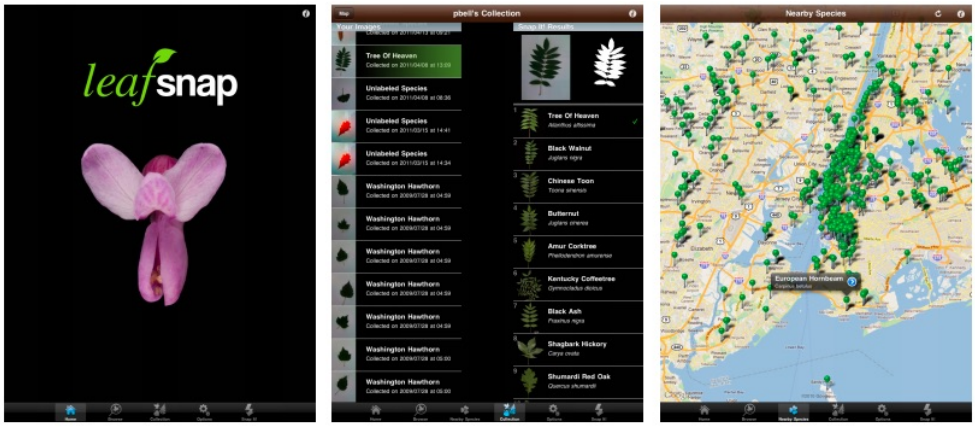


Figure 2.15: Leafsnap mobile application developed by Kumar et al. (2012). The user takes a picture of a leaf, the photo gets uploaded to server and a ranking list (middle image) returned back to the user. The app also offers guide to nearby species (right image).

the functionality to visually identify a particular species within a domain. Also, each app offers guides around the species once it is recognized by the app as shown in Fig. 2.15.

Another app that stands out is PhotoId [4]. It does not offer a stand-alone app like everybody else does. Instead, users can take a picture and tweet [5] it to certain accounts. These accounts implemented the Twitter API and can automatically tweet back once the fine-grained classification has done.

2.8 Notable datasets

This section is an overview of some recent fine-grained related benchmark datasets that will appear in the experiment section of this thesis. We also collected a new and much larger flower dataset during the course of my study, which will be presented in Chap. 6.

[4] http://photoid.naturalvision.de
[5] http://twitter.com

2.8.1 Oxford Flowers

Nilsback et al. has done extensive work on flower species recognition Nilsback (2009), Nilsback and Zisserman (2006, 2007, 2008, 2009), during which they introduced two flower datasets that have served as a long standing benchmark for the fine-grained community, the *Oxford Flowers 17* and the *Oxford Flowers 102*.

The Oxford Flowers 17 dataset (Nilsback and Zisserman (2006)) contains 17 different flower species with 80 images per category. The dataset provides three different data splits with each having 60 training and 20 test images. And 818 out of the 1360 images have hand-annotated ground truth segmentations. The dataset was created by crawling the web for images with the common names of species and manually filtered for quality and correctness by the authors.

The Oxford Flowers 102 dataset (Nilsback and Zisserman (2008)) has 8289 images divided into 102 categories with 40 to 250 images per category. For each category, 10 training and 10 validation images are predefined, while the rest is left for testing. The 102 species chosen are among the most common ones living in the UK. And the data curation follows the same as in Oxford Flowers 17.

As of 2014, both datasets are considered by many as solved, with the very basic version of Fisher vector SVM setup achieving over 80% mean accuracy on Oxford Flowers 102. The few remaining errors are mostly due to the small size of training data (20 images per category).

2.8.2 Caltech UCSD Birds

Amazon Mechanical Turk (AMT) paved ways to get more images curated at little cost. One of the first datasets to have benefited from AMT were the Caltech UCSD Birds.

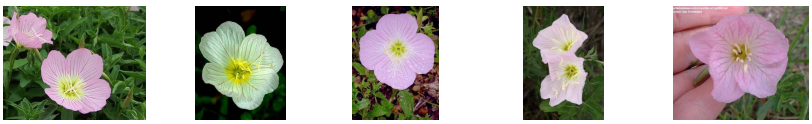
Pink primrose	
Canterbury Bells	
Sweet Pea	

Table 2.1: Images samples from the Oxford Flowers 102 by Nilsback and Zisserman (2008).

Caltech-UCSD Birds 200-2010 dataset by Welinder et al. (2010b) contains 200 classes with each class having exactly 15 training images and 15+ test images, resulting in 6033 images in total. In 2011, the authors doubled the number of train and test images while keeping the same species. The datasets were very challenging when first introduced in 2010, with the baseline accuracy 6% using state of the art algorithms.

The 2010 version also contains rough foreground segmentation masks, bounding boxes and attributes such as *blue wings* or *yellow beaks*, while the 2011 version added parts coordinates for 14 locations such as *left eye* or *tail* to the additional annotations, and increased the number of training and test images to 30. The addition of those parts locations are significant to make the 2011 version one of the most important benchmark datasets in FGVC, since it enabled the training of parts detectors in a supervised way, which turned out to be very promising.

The collection of the datasets, used the AMT service. For the part localization, the authors designed a novel system that can predict the location of the parts based on those already annotated, thus easing the task of the AMT operators and increase the annotation quality.

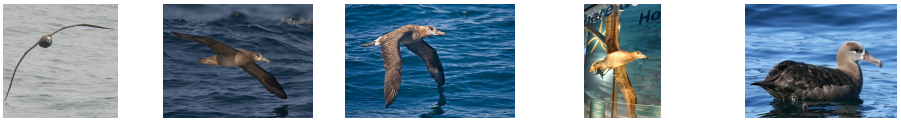
Black footed Albatross	
Least Auklet	
Brewer Blackbird	

Table 2.2: Image samples from the Caltech-UCSD Birds 200 dataset Welinder et al. (2010b).

Overall, as of 2014, accuracies over 75% can be achieved without using any annotations during test time, and over 85% if parts locations are giving during test time. A table containing performances of recent works on this dataset is collected by Branson et al. (2014).

2.8.3 ImageNet and Stanford Dogs

ImageNet by Deng et al. (2009) is one of the most impactful dataset in the recent years. Although it is not a purely fine-grained dataset like the Oxford Flowers or the Caltech-UCSD Birds, subsets of it can often be treated as standalone fine-grained sets, such as the Stanford Dogs dataset Khosla et al. (2011).

ImageNet was built using a word hierarchy in the English language, WordNet [6], which contains (as of 2014) 155'287 words and 206'941 relations between them. The authors of ImageNet retrieved images from the web for most of the nouns in the WordNet, and filtered the images using the AMT service. Prior to the use of the AMT service, downloaded images had to be filtered by authors themselves, which was a

[6] http://wordnet.princeton.edu

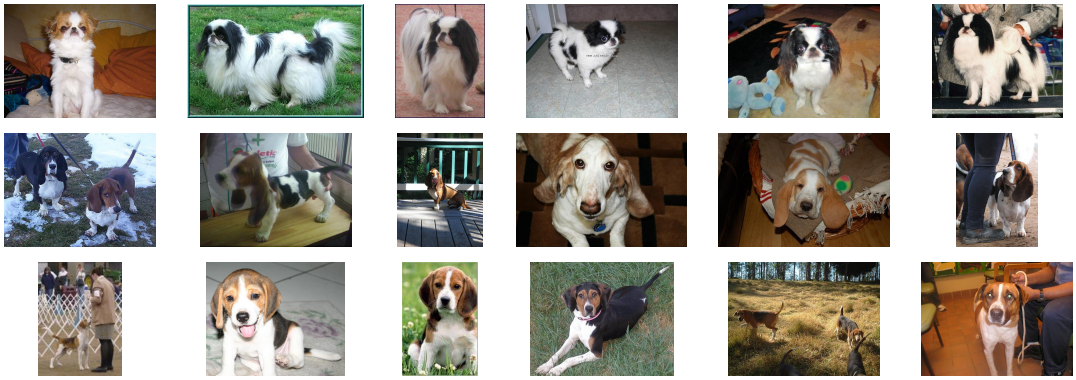


Table 2.3: Image samples from the Stanford Dogs dataset Khosla et al. (2011). Each row shows one of the 200 species.

tedious task and less scalable. ImageNet on the other hand leveraged the AMT and thus produced a dataset that contains orders of magnitude more images than any other existing publicly available ones. As of late 2014, it contains over 14 millions images across 21'841 synsets (categories). Another notable contribution of ImageNet is its yearly classification challenge, which has become the de facto benchmark for the classification community. Great lines of works have emerged through this challenge, most notably the convolutional neural nets Krizhevsky et al. (2012).

Stanford Dogs Khosla et al. (2011) is a purely fine-grained subset of the ImageNet The authors took 120 dog species from the ImageNet hierarchy and had the images filtered once more by the AMT. There are 100 images per class for training and about 80 images for testing. Bounding boxes are also provided.

Chapter 3

Classification with BiCoS

There is abundant evidence that accurate foreground-background segmentation can benefit visual recognition (e.g. Malisiewicz and Efros (2007)). This is particularly true when image classification deals with fine-grained visual categories, e.g. flowers or birds species, as pointed out by Nilsback and Zisserman (2006). In this scenario, background visual features tend to be similar across categories and typically act as a distraction to statistical learning rather than as useful context. Removing background prior to the classification stage therefore gives a significant boost to the overall performance. However, the supervised training of an automatic foreground segmentation tool requires pixel-wise foreground-background annotation of images, which is expensive and tedious to obtain. To this end we introduce a new scalable, alternation-based algorithm for co-segmentation, *BiCoS*, which is simpler than many of its predecessors, and yet has superior performance on standard benchmark image datasets.

Co-segmentation of image collections has recently become a topic of active research (see Hochbaum and Singh (2009), Joulin et al. (2010), Mukherjee et al. (2009), Rother et al. (2006), Vicente et al. (2010, 2011)). Co-segmentation methods consider sets of

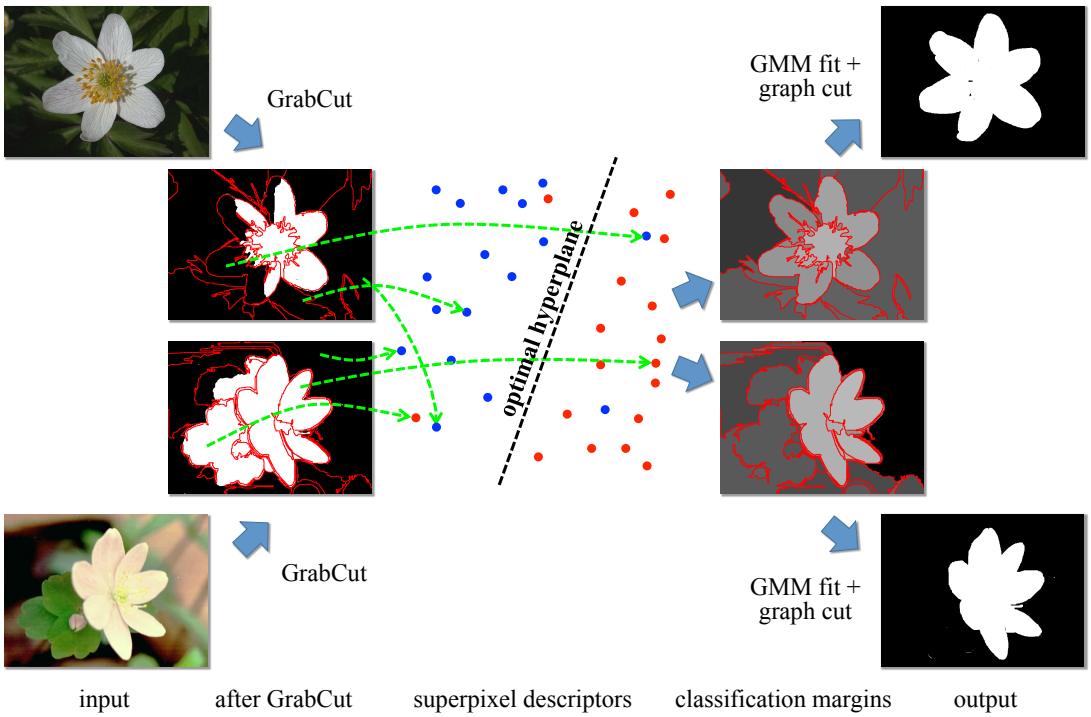


Figure 3.1: The diagram of our method (BiCoS) shown for 2 (out of 15) training images from one class of the Oxford Flowers-17 dataset. BiCoS starts with GrabCut at the pixel-level applied to each image independently. Each superpixel (superimposed) is then assigned to foreground or background and mapped to a descriptor space, where the optimal separating hyperplane is found. The pixels in each image are then assigned values according to the margins of the respective superpixels, and such soft segmentation map is then used to fit color Gaussian mixtures and apply pixel-level graph cut to each image.

images where the appearance of foreground and/or background share some similarities, and try to leverage these similarities to obtain accurate foreground-background segmentations either totally-unsupervised, or with a small amount of interactive supervision (Batra et al. (2010)). Most of the proposed co-segmentation methods (with the exception of Joulin et al. (2010)) assume close similarity of the foreground color histograms essentially requiring foreground objects to be the same through the image set.

Our goal in this method is to develop a method for unsupervised foreground-background co-segmentation of image sets that is scalable to large datasets of thousands of images. Our research is motivated by the desire to use unsupervised co-segmentation for the task of background removal within the training image sets for image classification systems.

The method operates at two levels of representation: at the bottom level, it treats each image separately and uses the well-known GrabCut algorithm by Rother et al. (2004) applied to the RGB values of individual pixels, whereas at the top level a discriminative classification is performed on high-dimensional descriptors of superpixels. The top layer operates on all images jointly and propagates information about foreground and background appearances across the dataset. Fig. 3.1 illustrates the algorithm schematically. This simple algorithm, which we term *BiCoS* for Bi-level Co-Segmentation, scales linearly with the number of images. Unlike most co-segmentation algorithms it does not assume the similarity of either global geometric shape or foreground color distributions throughout the image set.

Motivated by our ultimate application, we also propose a *multi-task* modification of BiCoS for co-segmenting *multiple* image sets each corresponding to a different class. This (BiCoS-MT) algorithm solves the co-segmentation tasks jointly, ensuring

the similarity of the background appearances across the classes.

We then evaluate both BiCoS and BiCoS-MT in the context of building an image classification system. The evaluation proves convincingly that applying unsupervised co-segmentation to the training sets can benefit the classification accuracies. As an outcome, we report better than state-of-the-art classification accuracy on the Oxford-17 and Oxford-102 flowers datasets (see Sec. 2.8) at the time of publication. For these datasets, we compare BiCoS and BiCoS-MT with GrabCut by Rother et al. (2004), a state-of-the-art co-segmentation system by Joulin et al. (2010) as well as with the system of Nilsback and Zisserman (2007) that is designed specifically for flower segmentation and is trained in a supervised way. Despite their simplity, the proposed algorithms came out on top both in terms of the *segmentation accuracy* and in terms of the *classification accuracy* obtained by the visual classifier trained on the segmented training set. We then extend the experimental evaluation to the Caltech-UCSD bird species dataset, which is very difficult for both segmentation and recognition.

Closely related to co-segmentation algorithms, is the group of approaches that deal with an unlabeled image set of an unknown class and aim to simultaneously segment the images and build a loose geometric model of that class (Alexe et al. (2010a), Arora et al. (2007), Liu et al. (2007), Todorovic and Ahuja (2006), Winn and Jojic (2005)). These algorithms are typically evaluated and achieve excellent results on the viewpoint-constrained datasets (Weizmann Horses by Borenstein and Ullman (2002), Caltech-4 by Fei-Fei et al. (2006)), where their geometric modeling is very appropriate. In contrast, BiCoS algorithms do not attempt to build any geometric models and are therefore applicable to image sets of highly non-rigid classes under extreme variations of viewpoints (such as user photographs of flowers taken in unconstrained setting). Nevertheless, we evaluate BiCoS on a number of viewpoint-constrained datasets, find-

ing out that, despite the lack of geometric modeling, the performance of BiCoS is rather competitive with many of the recent geometry-based methods.

3.1 Segmentation methods

We start with a review of GrabCut by Rother et al. (2004) that works at single image level. We then introduce the BiCoS method that is based on the combination of GrabCut and discriminative learning in the superpixel descriptor space. After that, we demonstrate how the proposed method may be modified to work with a dataset of multiple image sets with shared background patterns (a scenario typical for image classification as discussed above).

3.1.1 Image-level segmentation: GrabCut

The GrabCut algorithm by Rother et al. (2004) is a popular tool for the segmentation of single images. It combines two components: a binary-label random field defined on image pixels, and a generative foreground/background classifier that takes pixel RGB values as features. GrabCut proceeds by iterations: starting from some initialization, it alternates between **(1)** The (re)estimation of foreground and background probability densities via Gaussian mixtures given foreground/background pixel labels (Blake et al. (2004)), and **(2)** The graph cut inference of foreground/background labels in the random field with unary terms defined according to the evidence from the Gaussian mixtures and the pairwise terms defined according to local image gradients cues (Boykov and Jolly (2001)).

The assumption behind GrabCut and in particular behind the step (1) in its alternation scheme, is that the RGB distributions of foreground and background are shared across

the entire image (so-called *global color modeling*). As demonstrated in Rother et al. (2004) and subsequent evaluations, GrabCut is capable of producing accurate segmentation even when initialized in a very crude way (e.g. a rectangular mask overlapping the true foreground).

3.1.2 Class-level co-segmentation

Given a set of images of the same class, co-segmentation algorithms attempt to improve the segmentation accuracy by assuming the similarity of the visual appearance of foreground and/or background across the images in the dataset. A number of approaches (Hochbaum and Singh (2009), Mukherjee et al. (2009), Rother et al. (2006), Vicente et al. (2010)) work with pixel RGB values to model and propagate the distributions of visual appearance. However, in realistic scenarios for the vast majority of classes, RGB distributions of foreground and background pixels across the entire set of images overlap too much for such approaches to be useful. In other words, the color-based representation which works very well at the level of individual images is often unsuitable at the dataset level.

Rather than share a simple descriptor (RGB) at the level of individual pixels across the dataset, we share a richer descriptor at the level of super-pixels. This richer descriptor (a high dimensional feature vector) is suitable for discriminative learning at the dataset level. Thus, each image is partitioned into a set of superpixels (via the graph-based method by Felzenszwalb and Huttenlocher (2004), preferred for its time efficiency) and described by a real-valued descriptor (with the dimensionality $D = 1076$) stacked from five fairly standard sub-descriptors, representing the super-pixels' color distribution, SIFT (Lowe (1999)) distribution, size, location within the image and shape (more details below).

The super-pixels are then classified into foreground and background using standard linear support vector machine (SVM) training. Assume that a set $\{\mathbf{x}_1, \mathbf{x}_2 \dots \mathbf{x}_N\}$ of N descriptors corresponding to all superpixels are given. Let also $\{y_1, y_2 \dots y_N\}$ be the binary labels where $y_i = +1$ ($y_i = -1$) corresponds to superpixels with the majority of pixels assigned to foreground (background). Then, the separating hyperplane $\mathbf{w}$ in the descriptor space may be obtained by solving the standard SVM (convex quadratic) optimization program:

$$\frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{N} \ell(y_i \cdot \mathbf{w}^T \mathbf{x}_i) \longrightarrow \min_{\mathbf{w}}, \tag{3.1}$$

where $\ell(t) = \max(0, 1-t)$ is the hinge loss function, and C is the regularization constant set to 10 in all our experiments.

Our co-segmentation method then integrates this discriminative learning step that propagates the information about appearance distributions across images with GrabCut steps that propagate the information within images. Overall, BiCoS has the following steps (Fig. 3.1):

1. **Initialization at the image level.** BiCoS starts with GrabCut being applied to each image independently. GrabCut here is initialized with the rectangle in the center (50% of the image size, unless noted otherwise) assigned to the foreground and the rest to background. The output of this step is a pixel level segmentation of each image.

2. **Propagation at the dataset level.** Each superpixel is assigned in each image to either foreground or background depending on the label of the majority of its pixels. Given the set of superpixels and their labels (over all images), the separating hyperplane $\mathbf{w}$ is then learnt using a linear SVM. Each superpixel is

then re-classified according to its sign w.r.t. to the hyperplane (so that superpixels on the "wrong" side of the hyperplane effectively switch label). The output of this step is a linear classifier and a classification of all super-pixels across the dataset.

3. **Update at the image level.** Within each image independently, the Gaussian mixture distribution are reestimated from the new foreground and background regions using the pixels that belong to superpixels classified as foreground or background in the previous step. Each pixel is weighted according to the distance from the separating hyperplane, i.e. the margin $\mathbf{w}^T\mathbf{x}_i$. Given the new mixtures, a single graph cut (Boykov and Jolly (2001)) is then applied with the potentials defined as in GrabCut. The output of this step is again a pixel level segmentation of each image.

One can iterate the sequence of steps 2 and 3. For efficiency reasons, however, we adopted the single-pass version of BiCoS for our experiments (unless noted otherwise). Overall, scalability to large datasets is an attractive property of the method. Essentially, the scaling is linear in the number of images: the SVM in step 2 can be trained in linear time in the number of superpixels (Shalev-Shwartz et al. (2007)) and all other steps are done independently within each image.

We have used a discriminative classifier. Due to the high dimensionality of superpixel descriptors and obvious interdependencies in their dimensions, generative modeling and classification of foreground and background distributions would be problematic, although one can use naive-Bayes approximation (Alexe et al. (2010a)) as well as topic models (Cao and Fei-Fei (2007), Russell et al. (2006)).

Implementation details. We use the popular Felzenswalb's code for superpixel segmentation by Felzenszwalb and Huttenlocher (2004), with sigma=1, k=200, minSize=640.

Parameter values were chosen on another, but similar, problem, and are used unchanged for all segmentation experiments. We observe that the system is not too sensitive to the method of superpixel segmentation or its parameter settings.

The superpixel descriptor is composed from the following parts: (1) a 200-dimensional color and a 800-dimensional SIFT histograms that are obtained by vector-quantizing and pooling the densely-sampled *Lab* color values and multi-scaled dense SIFT descriptors respectively. (2) The size is encoded as a tiny 2 bins-histogram with the 1 value placed depending on whether the superpixel is "big" or "small" (the meaning of "big" and "small" is obtained by clustering the sizes of superpixels in the training set into two clusters). (3) The location in the image is represented with a 36-D descriptor obtained by linearly down-sampling the image-size binary mask, indicating pixels belonging to the superpixel, to 6×6 size. (4) The shape descriptor is obtained in the same way, except that the mask is cropped with the superpixel bounding box prior to down-sampling. (5) Finally, two 0/1 values encoding whether the superpixel contains the center of the image and whether it touches the boundary of the image. The resulting descriptor is quite high-dimensional (1076-D), but is relatively cheap to compute and permits the similarity computation by a simple dot product (hence no need for the kernelization of the algorithms). The vector-quantization is done with locally-constrained linear coding Wang et al. (2010).

3.1.3 Co-segmenting multiple class image sets

The GrabCut algorithm achieves a remarkable accuracy by making foreground and background regions of an image share appearance distributions. Co-segmentation algorithms including BiCoS are capable of improving segmentations by enforcing the sharing of such distributions across the images of the same class. In this subsection,

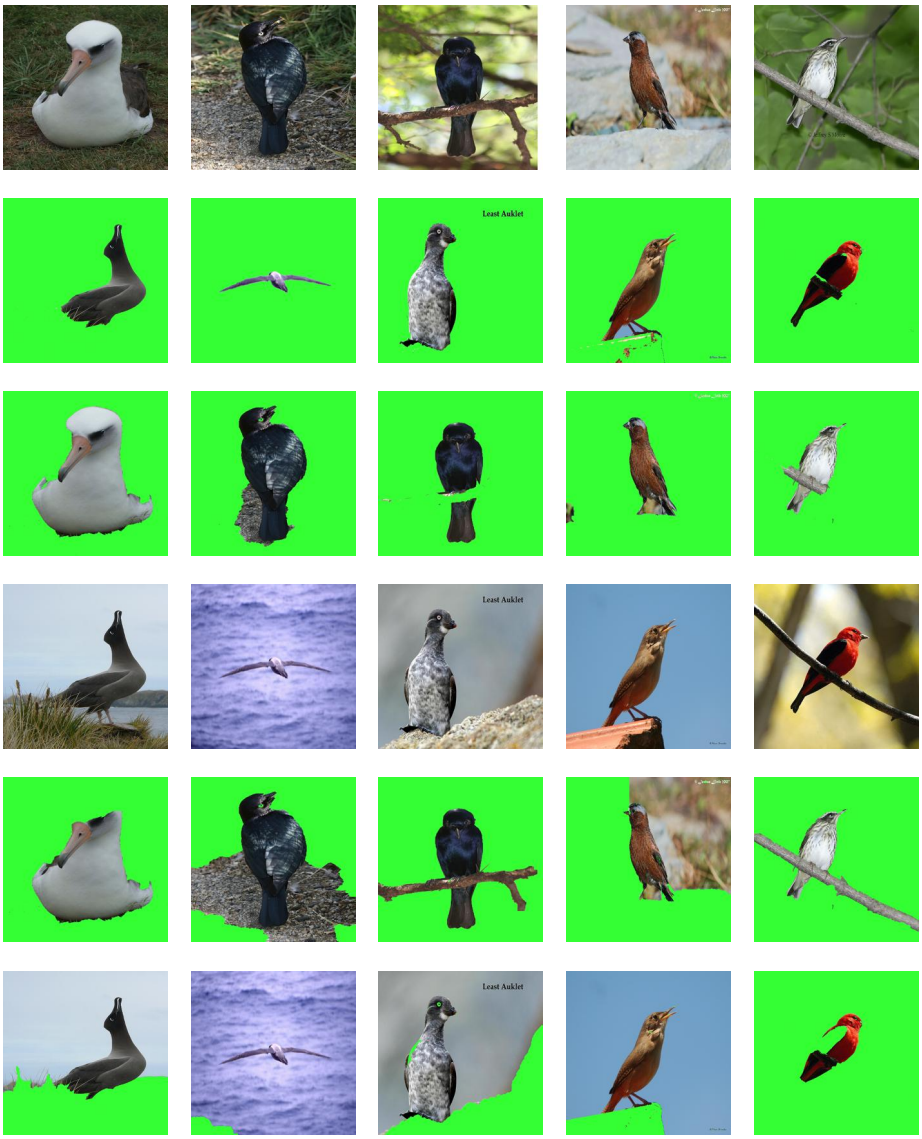


Figure 3.2: Bird segmentations. Every 3 rows: (top) original images, (center) using BiCoS and (bottom) using BiCoS-MT. In the top 5 images, BiCoS-MT performs better because it takes into account background elements (*e.g.* tree branches) in the background of images from other classes. But due the existence of confusing foreground elements (*e.g.* blue birds) in other classes, sky can be mis-classified using BiCoS-MT.

we make one step further and consider the scenario where one is given a dataset that contains multiple sets of images $\mathcal{S}_1, \mathcal{S}_2 \ldots \mathcal{S}_K$ corresponding to K classes, with each set $\mathcal{S}_k$ containing N_k images ($\mathcal{S}_k = \{\mathcal{I}_1, \mathcal{I}_2, \ldots \mathcal{I}_{N_k}\}$), where N_k may be as small as 1 and as large as several hundred.

Clearly, such datasets may be segmented by running co-segmentation algorithm for each set S_k independently. However, one may attempt to improve the segmentation even further by enforcing appearance sharing across classes. As discussed in the introduction, we assume that the ultimate goal in this scenario is the improvement of the recognition accuracy for classifiers trained from the respective training sets after background removal. The consequence of this observation is that a "good" co-segmentation process should have a tendency to assign regions with the appearance patterns that are ubiquitous across multiple image sets to background (as these patterns would not be discriminative but rather confusing for class discrimination).

Our second approach (BiCoS-MT) enforces background sharing by modifying the SVM formulation in BiCoS. For each class-specific dataset S_k, the algorithm finds a separate hyperplane that discriminates between the background and the foreground appearances typical for that class. The hyperplanes for different classes are however not computed independently. Instead, for the class k, the separating hyperplane is found as the sum of two vectors $\mathbf{w}_k + \mathbf{w}_-$, where $\mathbf{w}_k$ is the class-specific part and $\mathbf{w}_-$ is the component shared across all classes. The objective of the joint SVM formulation is then naturally defined as:

$$\frac{\mu}{2}||\mathbf{w}_-||^2 + \sum_{k=1}^{K}\left[\frac{1}{2}||\mathbf{w}_k||^2 + C\sum_{i=1}^{N_k}\ell\left(y_i^k \cdot (\mathbf{w}_k + \mathbf{w}_-)^T\mathbf{x}_i^k\right)\right] \rightarrow \min_{\{\mathbf{w}_1 \ldots \mathbf{w}_K, \mathbf{w}_-\}}. \quad (3.2)$$

Thus, (3.2) is defined as a sum of K independent SVM objectives for each class along with the vector component $\mathbf{w}_-$ (and its regularization term) shared across the SVMs.

(In (3.2), $\mathbf{x}_i^k$ and y_i^k denote superpixels within the dataset $\mathcal{S}_k$; N_k denotes the number of superpixels in image set $\mathcal{S}_k$; μ is an additional regularization parameter set to $\frac{1}{2}$ in all our experiments.)

Since we want to encourage the sharing of background and discourage the sharing of the foreground appearance patterns, we impose additional constraints on (3.2):

$$\mathbf{w}_k^j \geq 0, \quad \forall j = 1 \dots D \tag{3.3}$$

$$\mathbf{w}_-^j \leq 0, \quad \forall j = 1 \dots D \tag{3.4}$$

An intuition behind the constraints (3.4) is that they make the separating hyperplanes for different classes share negative coefficients (that are indicative of background superpixels), while positive coefficients can be inferred for each class independently. With respect to the original SVM formulation, it is easy to show that for the case of a single class $K = 1$ and equal regularization on $\mathbf{w}_1$ and $\mathbf{w}_-$ (i.e. when $\mu = 1$), the standard SVM formulation (3.1) and the new formulation (3.2)–(3.4) are exactly equivalent and produce the same separating hyperplane.

Based on the augmented SVM formulation, we devise the BiCoS-MT algorithm that uses the joint program (3.2)–(3.4) to estimate the separating hyperplane $\mathbf{w}_k + \mathbf{w}_-$ for each class k at step 2. The rest of the steps are unchanged from BiCoS. We note that the joint SVM formulation can be naturally interpreted as an instance of multi-task learning (Caruana (1997), Evgeniou and Pontil (2004)) (hence the name BiCoS-MT). Within BiCoS-MT, each task corresponds to learning a superpixel classifier for a particular image set $\mathcal{S}_k$. We also note that "softer" version of background sharing may be implemented, where each class possesses both positive component $\mathbf{w}_k^+$ and negative component $\mathbf{w}_k^-$ specific to that class, whereas the extra term $\nu \sum_k ||\mathbf{w}_k^- - \mathbf{w}^-||^2$ penalizes the deviation of the negative components from their joint mean vector

$\mathbf{w}^-$ that does not enter in any other terms (c.f. Evgeniou and Pontil (2004)).

The program (3.2)–(3.4) may be rather large (many tens of thousands of variables in some of our experiments), and we therefore implemented the modification of a popular stochastic gradient descent-based method (Pegasos by Shalev-Shwartz et al. (2007)) to handle the task. This allows (3.2)–(3.4) to scale to a large number of classes and/or a large number of images per class.

Algorithm 1 Modified Pegasos algorithm for BiCoS MT

Require: $[\mathbf{w}_{-,1}, \mathbf{w}_{1\ldots K,1}, S_{1\ldots K}, \lambda_{1\ldots K}, \mu, T, N]$
  for $(t = 1, 2, \ldots, T)$ **do**
    for $(k = 1, 2, \ldots, K)$ **do**
      Choose $A_{k,t} \in S_k$, where $|A_{k,t}| = N$
      Set $A^+_{k,t} = \{(\mathbf{x}, y) \in A_{k,t} : y\langle \mathbf{w}_{k,t} + \mathbf{w}_{-,t}, \mathbf{x}\rangle < 1\}$
      Set $\eta_{k,t} = 1/(\lambda_k t)$
      Set $\nu_{k,t} = 1/(\mu t)$
      $\mathbf{w}_{k,t+\frac{1}{2}} = (1 - \eta_{k,t}\lambda_k)\mathbf{w}_{k,t} + \frac{\eta_{k,t}}{N} \sum_{(\mathbf{x},y)\in A^+_{k,t}} \mathbf{x}y$
      $\mathbf{w}_{-,t+\frac{1}{2}} = (1 - \nu_{k,t}\mu)\mathbf{w}_{-,t} + \frac{\nu_{k,t}}{N} \sum_{(\mathbf{x},y)\in A^+_{k,t}} \mathbf{x}y$
      Replace all negative elements in $\mathbf{w}_{k,t+\frac{1}{2}}$ with 0
      Replace all positive elements in $\mathbf{w}_{-,t+\frac{1}{2}}$ with 0
      $\mathbf{w}_{k,t+1} = \min\{1, \frac{1/\sqrt{\lambda_k}}{||\mathbf{w}_{k,t+\frac{1}{2}}||}\}\mathbf{w}_{k,t+\frac{1}{2}}$
      $\mathbf{w}_{-,t+1} = \min\{1, \frac{1/\sqrt{\mu}}{||\mathbf{w}_{-,t+\frac{1}{2}}||}\}\mathbf{w}_{-,t+\frac{1}{2}}$
    end for
  end for
  return $[\mathbf{w}_{-,T+1}, \mathbf{w}_{1\ldots K,T+1}]$

A detailed pseudocode of the modified Pegasos solver for (3.2)–(3.4) is in Alg. 1.

As illustrated in Fig. 3.2, sharing the background among all classes may lead to better segmentation, but may also cause mis-segmentation. But for the majority of our experiment setups, BiCoS-MT has an advantage over BiCoS.

3.2 Other recent co-segmentation works

The discriminative learning in the space of superpixel descriptors used at the top level has been employed within *semantic segmentation* methods such as Fulkerson et al. (2009), Gould et al. (2009), Hoiem et al. (2005), which are trained in a supervised way.

Among the unsupervised co-segmentation algorithms, discriminative learning on superpixels is used by Joulin et al. (2010). The optimization framework of Joulin et al. (2010) simultaneously enforces spatial smoothness within each image as well as finding the foreground-background boundary in the superpixel space. Unlike Joulin et al. (2010), BiCoS decouples spatial smoothness enforcement and classification of superpixels, so that these two steps are performed consecutively rather than simultaneously. We demonstrate experimentally that, despite the sub-optimality that such alternation-based approach might bring, BiCoS consistently attains higher segmentation accuracies, while being applicable to much larger image sets than Joulin et al. (2010).

Alexe et al. (2010a) is another very recent work closely related to ours, as their system also uses superpixels to propagate information across multiple images. Such propagation is however achieved through binary-label Conditional Random Field (CRF) with unary potentials derived in a generative fashion as opposed to discriminative learning (SVM) used within BiCoS. Despite the use of explicit geometric modeling within Alexe et al. (2010a), the experimental comparison revealed that BiCoS is able to achieve similar or higher segmentation accuracy for several viewpoint-constrained datasets, where their loose geometric model is appropriate.

Another co-segmentation work that adopts an alternation strategy similar to the bi-level architecture of BiCoS is Batra et al. (2010). Their appearance models are however

based purely on color (and hence are too limited for many scenarios). Their focus is also on interactive user supervision, rather than the fully unsupervised scenario used by most other co-segmentation works, as well as ours.

3.3 Experimental results

Performance measures. During the evaluation of co-segmentation systems, we are interested in two measures: the segmentation accuracy and the accuracy of the recognition achieved by a visual classifier trained on the segmented dataset, and we report these measures for three image classification datasets and several methods.

Given a ground truth foreground mask, the segmentation accuracy can be expressed in several ways. One is the ratio between the size of the intersection area between the estimated foreground and ground truth foreground over the size of their union (**Seg. I**). The other is the percentage of pixels classified correctly as either foreground or background (**Seg. II**). We report both numbers for our methods, and whichever is published for the competitors. For the average recognition accuracy (**Rec.**), we take the mean value of the relative numbers of correctly classified images of each category (i.e. the average class accuracy) by a state-of-the-art visual classifier briefly described below.

Evaluation on image recognition. In order to evaluate the effect of co-segmentation on the recognition accuracy we apply the following training pipeline: (i) BiCoS, BiCoS-MT or one of the baseline segmentation algorithms is used to obtain foreground segmentation masks on the training data (note, in all cases the algorithm can make use of the class labels of the training data – BiCoS can be applied to each class independently, and BiCoS-MT is formulated to use this information); (ii) a 1-vs-rest SVM

image classifier is learnt for each category (see below) using the foregrounds of that category as positives, and the foregrounds of all other categories as negatives; (iii) a generic segmenter is learnt (which will be applied to the test images) as an SVM for super-pixels using the foreground regions of *all* categories as positives and the background regions of *all* categories as negatives (note, at test time we do not know the class label so cannot use a BiCoS algorithm just for that class). The test pipeline then proceeds as: (i) segmenting into foreground background using the generic segmenter SVM on superpixels followed by step 3 of BiCoS; (ii) classifying the foreground image using each of the 1-vs-rest SVM image classifiers; and (iii) returning the class with the maximum score over all the image classifiers.

Implementation Details. The features for the image recognition part are obtained by concatenating the Bag-Of-Words (BoW) histograms of *Lab* color and SIFT descriptors, which are extracted solely from the foreground area given by the segmentation. We use three different histograms for SIFT feature corresponding to dense sampling at several scales, sampling at interest points, and sampling along the foreground boundary. Similar to the superpixel descriptors, the vector-quantization uses locally-constrained linear coding by Wang et al. (2010). The vocabulary sizes are 800 for *Lab* and 8000 for each of the 3 SIFT descriptors. Before concatenating the four descriptors together, we apply the homogeneous kernel map with approximation order 1 and sampling step 0.7 (Vedaldi and Zisserman (2010)) onto it, which results in a 74400 dimensional feature vector, that is used within the linear SVM (with $C = 1$). The implementation uses VL-Feat by Vedaldi and Fulkerson (2010).

3.3.1 Oxford Flowers 17

As discussed in Sec. 2.8, the Oxford Flowers 17 dataset contains 17 different flower species with 80 images per category. The dataset provides three different data splits with each having 60 training and 20 test images. And 818 out of the 1360 images have hand-annotated ground truth segmentations. The fact that these ground truth segmentations are unequally distributed among all categories, makes it difficult to compare segmentation and recognition accuracies simultaneously.

We first compare our algorithms to the recently published co-segmentation algorithm by Joulin et al. (2010). We use their provided code and their suggested parameter settings for this dataset. The memory demands of this algorithm are too excessive to use the predefined 60/20 train/test split. Instead, a new data split having 15 training and 65 test images per category is used. This split has 15 training images with ground truth segmentations for 16 of the categories (out of the total of 17). Thus, for the new split we can compare both segmentation and recognition accuracies. We note that other published co-segmentation methods either do not scale to even this number of images or require user supervision Batra et al. (2010), thus making Joulin et al. (2010) the natural choice for the comparison.

The numerical comparison between our own methods and method by Joulin et al. (2010) is shown in Tab. 3.1, while some of the segmented images are shown in Fig. 3.3. In Tab. 3.1 and thereafter, we also provide the accuracies for the cases when no pre-segmentation of training and test images is performed, and when the training dataset is pre-segmented with GrabCut.

We also compare the recognition accuracy of our full system that uses BiCoS to co-segment training image with the state-of-art recognition results using the predefined data splits. We show our recognition accuracies in Tab. 3.2 along with the results



Figure 3.3: Flower segmentations. Every 3 rows: (top) original images, (center) segmentations by Joulin et al. (2010) and (bottom) the segmentations of BiCoS-MT. The results of BiCoS are very similar to BiCoS-MT for these particular images.

Methods	Seg. I	Seg. II	Rec.
All foreground	32.8	32.8	62.1
Joulin et al. (2010)	75.8	86.6	74.1
Rother et al. (2004)	89.3	96.3	73.3
BiCoS (this work)	94.1	98.1	79.3
BiCoS-MT (this work)	94.7	98.3	80.5

Table 3.1: Performance on Oxford Flowers 17 (with the alternative data split – 15 images per class). Our algorithms achieves the highest results in all three measures. For this number of training images, BiCoS-MT also outperforms BiCoS.

Methods	Rec. Accuracy
Gehler and Nowozin (2009)	85.5 ± 3.0
Nilsback (2009)	88.1 ± 1.9
BiCoS (this work)	91.1 ± 1.5
BiCoS-MT (this work)	90.4 ± 2.3

Table 3.2: Performance on Oxford Flowers 17 (with the original data splits – 60 training images per class). Our algorithms once again lead to the highest recognition accuracies. For 60 images per class, BiCoS-MT no longer perform better than BiCoS.

provided by Gehler and Nowozin (2009) and Nilsback (2009).

3.3.2 Oxford Flowers 102

We recall the Oxford Flowers 102 dataset has 8289 images divided into 102 categories with 40 to 250 images per category. For each category, 10 training and 10 validation images are predefined, while the rest is left for testing. There is no segmentation ground truth provided in the dataset, therefore, we use the recognition accuracy as the only measurement.

Among all recent approaches evaluated on this dataset (Ito and Kubota (2010), Kanan and Cottrell (2010)), the best performance is reported in Nilsback (2009). The authors proposed a model-based foreground segmentation approach which segments images

Segmentation	Classification	Rec. Accuracy
no segmentation	ours	64.7
Nilsback (2009)	Nilsback (2009)	76.3
Nilsback (2009)	ours	76.8
Kanan and Cottrell (2010)	Kanan and Cottrell (2010)	72.8
Ito and Kubota (2010)	Ito and Kubota (2010)	74.8
Rother et al. (2004)	ours	77.0
BiCoS	ours	79.4
BiCoS-MT	ours	80.0

Table 3.3: Recognition performance for different combination of segmentation and classification approaches on Oxford Flowers 102. For a number of methods, we vary the (co-)segmentation approach while keeping the classification approach fixed. The proposed methods once again lead to the highest recognition accuracy.

independently. In the following experiments, we can show that this baseline can be surprisingly outperformed in our experiment framework even if the images are segmented independently using GrabCut alone. We also show that the accuracy can be pushed even further with the proposed methods (Tab. 3.3). We used the results of the segmentation by Nilsback (2009) available online to evaluate the combination of their segmentation system and our image classifier. The improvement obtained by BiCoS and BiCoS-MT over Nilsback (2009) in this experiment is all the more important, as the system by Nilsback (2009) required around 800 hand-annotated segmentation masks as training data.

3.3.3 Caltech-UCSD Birds 200

The Caltech-UCSD Birds 200 by Welinder et al. (2010b) contains 200 bird categories and 6033 images in total. There are *rough* segmentation masks provided with the dataset, which allows us to compare segmentation accuracies. However, we should note that those values are only approximated because of the rough segmentation ground

Methods	Seg. I	Seg. II	Rec.
no segmentation	17.0	17.0	6.7
Rother et al. (2004)	38.8	73.5	13.6
BiCoS	41.1	78.3	15.7
BiCoS-MT	39.9	77.3	16.2
using ground truth seg.	-	-	23.3

Table 3.4: Performance on Caltech-UCSD Birds 200. BiCoS algorithms outperform GrabCut in all measures. BiCoS-MT leads to slightly higher recognition accuracy. The bottom line provides the performance of our classification approach given ground truth segmentation.

truth obtained with MTurk. The dataset is extremely challenging, and its authors report only 19% recognition accuracy *when using ground truth masks*. Tab. 3.4 gives the segmentation and classification accuracies (using the suggested 20 training images per class split) of GrabCut, BiCoS, and BiCoS-MT paired with our recognition approach. We hope that they may serve as a baseline for further research on segmentation and co-segmentation.

3.3.4 Weizmann Horses and Caltech-4

We also evaluate BiCoS on datasets with fixed viewpoint orientation and classes that have well-defined geometric shape. This scenario is quite different from our main application and it is inevitable that methods interleaving segmentation and construction of class-specific geometric models are likely to have an advantage over BiCoS or any other method not having a geometric model. Still, it is interesting to see how well BiCoS can fare against such methods, in particular against Alexe et al. (2010a) to which it is most similar. We therefore evaluated BiCoS on Weizmann Horses by Borenstein and Ullman (2002) and 3 out of 4 Caltech-4 classes by Fei-Fei et al. (2006) (the grayscale 'cars' class was left out). We keep all the parameters fixed from the Flowers and Birds

Methods	horse		airplane		face		motorbike	
	Seg I	Seg II	Seg I	Seg II	Seg I	Seg II	Seg I	Seg II
All background	0.0	71.0	0.0	83.1	0.0	80.0	0.0	72.9
Joulin et al. (2010)	-	80.1	-	-	-	-	-	-
Cao and Fei-Fei (2007)	-	81.8	-	-	-	-	-	-
Alexe et al. (2010a)	-	86.2	-	89.8	-	89.0	-	90.3
Arora et al. (2007)	-	-	-	93.1	-	**92.4**	-	83.1
Winn and Jojic (2005)	-	93.1	-	-	-	-	-	-
Liu et al. (2007)	-	**95.9**	63.9	-	-	-	**71.6**	-
Rother et al. (2004)	60.6	86.3	59.2	90.7	60.1	86.2	41.1	80.6
BiCoS (1 iter.)	65.2	87.6	**64.5**	93.0	66.2	89.3	44.7	82.8
BiCoS (5 iter.)	68.7	90.0	63.8	**93.2**	69.0	91.1	41.8	82.4

Table 3.5: (Co-)segmentation performance on Weizmann Horses and Caltech-4 datasets reported in the literature. Despite the lack of the geometric model, BiCoS is competitive with the methods that have such models (see text for more discussion).

experiments, except that GrabCut is initialized with the central foreground rectangle occupying 25% (rather than 50%) of the image area.

The comparison with state-of-the-art is in Tab. 3.5. Here we also give results for the iterated BiCoS (repeating step 2 to step 3 for 5 times). In the experiments, BiCoS performed poorly on the motorbike class, where GrabCut often fails due to unnatural framing within many photographs. Apart from that, the performance of BiCoS is surprisingly competitive, in particular outperforming Alexe et al. (2010a) on the remaining 4 categories. The top performing method of Liu et al. (2007) outperforms BiCoS on all categories except airplanes where BiCoS (one iteration) is better. may also not perform well on the 'face' category where backgrounds are repeatable (Liu et al. (2007) decided not to report the accuracy there).

3.4 Discussion

We presented BiCoS: a co-segmentation method that is arguably very simple, yet is scalable and performs remarkably well in our experimental comparisons. BiCoS outperformed GrabCut that treats images independently and the state-of-the-art co-segmentation by Joulin et al. (2010) in all comparisons. Image classification systems trained on datasets co-segmented with BiCoS achieved state-of-the-art accuracy on the Flowers-17 and Flowers-102 datasets at the time of publication. For the small number of images per class (Tab. 3.1, Tab. 3.3) the multi-task version of BiCoS leads to even higher recognition accuracy.

Many co-segmentation methods attempt to segment images and infer the foreground and background appearance models at the same time. BiCoS follows a different, perhaps less principled way, and performs within-image and across-image appearance propagation consecutively rather than jointly. Such a two-step approach, however, allows a proper feature representation to be used at each level (color-based at the level of individual images, discriminative classification of rich superpixel descriptors at the dataset level). The success of BiCoS in our experiments therefore seems to concur with the well-known fact that devising appropriate feature representations has a higher impact on the performance of computer vision systems than the choice of a principled optimization algorithm.

Chapter 4

Classification with TriCoS

The simplicity of BiCoS paved path for further investigation using segmentation to improve fine-grained visual categorization. By putting an additional term for explicitly modeling class discrimination, we introduced TriCoS, a tri-level co-segmentation method.

Just as in BiCoS, TriCoS has a complexity linear in the total number of images and does not assume any global geometric shape or foreground color distribution throughout the set. Thus, the method can be applied to rather large training sets and we are able to evaluate it through a series of experiments on the datasets introduced in Sec. 2.8. In our experiments, we observe that a modern image representation (Fisher Vectors by Perronnin et al. (2010)) trained on the datasets pre-segmented with TriCoS achieves considerably better than state-of-the-art classification accuracies at the time of publication.

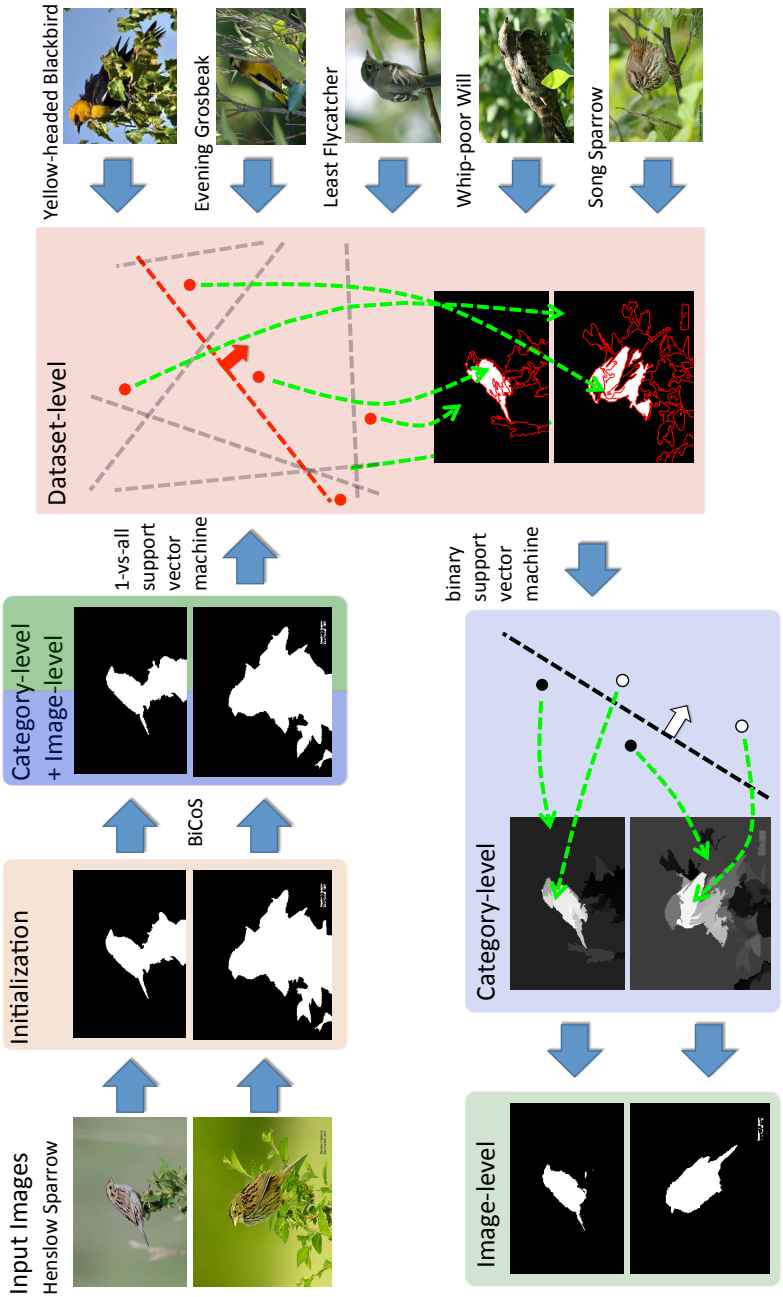


Figure 4.1: An overview of the flow of the discriminative co-segmentation pipeline TriCoS. We start with the input images and their initialization segmentation masks on the top left. They are then processed by the BiCoS. Although BiCoS often improves the segmentation quality, in this example, it preserves both the birds and the tree branches as foreground because they appear multiple times across the images from the same category. The new dataset-level step finds leaves to be less discriminative since most birds fly around trees and therefore discards the leaves in the segmentation.

4.1 Segmentation Methods

In TriCoS, we improved the initialization step, which is described in Sec. 4.1.1. The main contribution is the introduction of the new term for class discrimination in the co-segmentation objective function, which is described in Sec. 4.1.2. Finally, other improvements such as the choice of a stronger feature representation is described in Sec. 4.1.3.

Information is carried from one level to another in the form of masks, which, depending on the level, can be either hard segmentation masks (binary, either foreground or background) or soft saliency maps (real valued).

4.1.1 Objectiveness-based Initialization

BiCoS initialized the co-segmentation process using the central σ percent of pixels as a GrabCut initialization, with the hope that the category level can correct any error made by this simple assumption. While it performs well on many datasets where people tend to put the object of interest at the center of the images, we propose here a more principled way to initialize the foreground masks in each image that is based on an interest region detector. Other methods, e.g. Carreira and Sminchisescu (2010), Endres and Hoiem (2010), could equally well be used, but are more expensive than the method we propose.

The method proceeds in two stages. First, initial proposals are generated in two ways: the first approach follows van de Sande et al. (2011), which starts from superpixels by Felzenszwalb and Huttenlocher (2004), and iteratively merges the two most similar regions (based on size and gradient orientation histogram) together until the whole image is a single region. All superpixels generated in the process are included

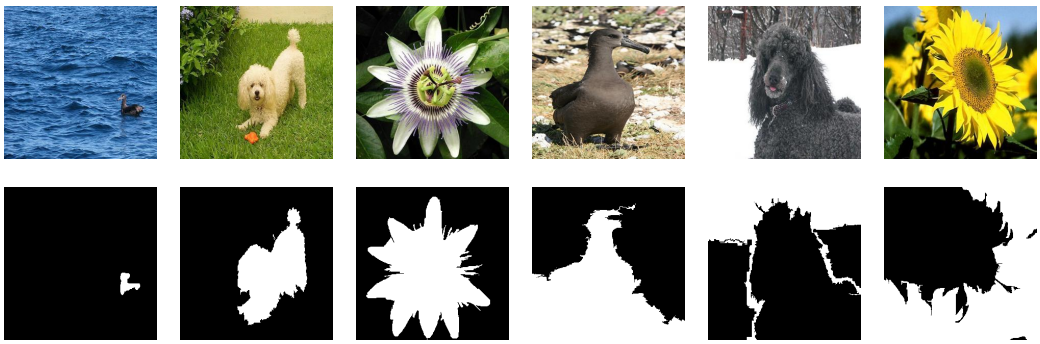


Figure 4.2: Examples of initializations. The initializations themselves can already be very good, as seen in the first 3 columns. However, if the foreground object is flat or smooth, the background can become more interesting. TriCoS is able to remedy this problem.

as separate proposals. The second approach applies a saliency map by Rahtu et al. (2010) as a cue. For each superpixel, we compute the mean saliency value as well as the 70th, 80th and 90th percentiles of saliency values. These four values are then assigned to the pixels inside the superpixel, one at a time, and the resulting image is thresholded to get a binary image. All the connected components from the resulting four binary images are considered as separate proposals.

In the second stage, we compute a simple saliency based score for each proposal in the initial set. The definition is similar to the multi-scale saliency score in Alexe et al. (2010b):

$$score = \frac{\sum_{i \in P} \delta(s(i) > 0.7)}{area(P)} \cdot \sum_{i \in P} s(i), \tag{4.1}$$

where $\delta(a) = 1$ if a is true and 0 otherwise, P refers to pixels inside the proposal, $s(i)$ is a saliency value at pixel i, and $area(P)$ is the area of the proposal. The score (4.1) favors large proposals with a small number of non salient pixels for which $s(i) < 0.7$. Finally, the best scoring proposal is selected for the foreground initialization. Examples of initializations can be found in Fig. 4.2.

4.1.2 Dataset-level Class-Discriminative Segmentation

The idea behind class-discriminative segmentation is to figure out which parts of the images actually contribute toward differentiation of the categories and which parts do not. To do so, we first train a multi-class classifier for the different categories using features summed over the foreground given by the preceding level (GrabCut). We then apply the learned classifier back to individual superpixels in each image of the class, thus obtaining a score for the discriminating power of each superpixel.

Before we introduce the class-discriminative information into the process, we apply a pre-processing data balancing step: where the process discussed in Sec. 3.1.2 is applied to the whole dataset jointly, ignoring class labels, and with slightly richer features (see Sec. 4.1.3). We found that such process balances the sizes of the foreground segments across classes, which decreases the amount of bias towards classes during the subsequent procedure. After such balancing, we introduce the class-discriminative information into the co-segmentation using the following approach.

For each image, we then extract a very high-dimensional feature vector (e.g. Fisher vector) from the given foreground area only for the multi-category classification task. As in the standard multi-class SVM, the goal is to train K separating hyperplanes in the feature space. Given a set $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ of descriptors for each of the N images and their category labels $\{y_1, y_2, \ldots, y_N\}$, we train a 1-vs-all SVM:

$$\frac{1}{2}||\mathbf{w}^k||^2 + C\sum_{i=1}^{N}\ell(y_i^k \cdot (\mathbf{w}^k)^T\mathbf{x}_i) \longrightarrow \min_{\mathbf{w}^k}, \forall k \in 1, 2, \ldots, K \tag{4.2}$$

where $\ell(t) = \max(0, 1-t)$ is the hinge loss function, and C is the regularization constant set to 10 in all our experiments. y_i^k is 1 if it belongs to the class k and -1 otherwise.

Once the classification model is trained, the question is how to use it to update the segmentation masks. To our best knowledge, there is no common solution to this type of propagation, and we propose the following solution.

Let us first consider the case of a dataset that has just two classes. As we use a linear SVM for classification, we have a single weight vector $\mathbf{w}$. Furthermore, we again extract superpixels as described in Sec. 3.1.2. For each superpixel, we extract the same feature vector $\mathbf{x}_j$ in the same way as in classification (note the change of the subscript as each feature vector $\mathbf{x}$ is composed in the same way as for classification, but is extracted from individual superpixels rather than images):

$$m_j = \text{sign}(y\mathbf{w}^T\mathbf{x}_j + \epsilon) \tag{4.3}$$

where ϵ is the bias term, y is the binary class label of the image the superpixel belongs to, and m_j is a binary foreground/background label for the superpixel $\mathbf{x}_j$.

In principle, superpixels with a positive score from $\mathbf{w}^T\mathbf{x}_j$ increase the margin of the right class and therefore should belong to foreground, while superpixels with a negative score should belong to background. However, as a result of overfitting in a high-dimensional space, the foreground area from where the classification model was trained will typically remain foreground after this process, while the background which was discarded during classification will be split into a foreground and a background part. To avoid such a uniform growth of a foreground superpixel, we introduce the bias ϵ which should have a small negative value. Intuitively, the model discards image superpixels that lie too close to the decision boundary as they can easily flip the side. As a result, the signal to noise ratio is increased. A high value of ϵ causes every superpixel to turn to foreground. And a low ϵ results in taking only a small but most discriminative part for classification, which may result in poor generalization.

Moving from the binary case to multi-class image classification, the propagation rule can be modified, in a way similar to the 1-vs-all classification rule. We thus compute the label of each superpixel as:

$$m_j = \text{sign}\left[(\mathbf{w}^{k(j)})^T \mathbf{x}_j - \max_{k'=\{1,2,\ldots,K\setminus k(j)\}} ((\mathbf{w}^{k'})^T \mathbf{x}_j) + \epsilon\right] \tag{4.4}$$

The superpixel label is now determined by the difference between the response of the true class and the maximum response of all other classes. In principle this may be suboptimal as a superpixel with positive score surely contributes to classification, an image superpixel with negative score may not be necessarily confusing, as it can contribute to discriminating from other classes. However, the category level that follows after the dataset level is designed to prefer consistency across the category and typically compensates for this sub-optimality.

By mapping all m_j onto the superpixels, we obtain a new set of binary segmentation masks which can be fed back to the category level.

4.1.3 Implementation Details

All superpixels we use in the entire pipeline are extracted using the graph-based method by Felzenszwalb and Huttenlocher (2004), with $k = 200$, σ and minsize set proportional to the image size. Typically, around 40-100 superpixels are generated for each image.

The descriptors for the superpixels in Sec. 3.1.2, are stacked from five fairly standard sub-descriptors covering information over color, shape, size and location, as described

in BiCoS. The color and SIFT histograms are of length 20 and 80, respectively. However for the global category-level step in Sec. 4.1.2, the histogram sizes are increased to 200 and 800, respectively.

For the multi-class SVM in Sec. 4.1.2, we sample SIFT features at multiple scales and Lab color features at dense grid locations within the category-independent segmentation masks for each image. Feature points outside the masks are discarded. Lab features are then encoded using LLC (Wang et al. (2010)) with 1000 dictionary entries, while PCA is applied to dense SIFT, and the 80 most principal dimensions are encoded using the Fisher vectors (Perronnin et al. (2010)) with a Gaussian mixture of 256 components. The total feature dimension is therefore 41960 after stacking the two descriptors.

4.2 Experimental Results

As we aim at class-discriminative co-segmentation, which may well differ from human-annotated foreground segmentations, we skip any segmentation accuracy measure and only measure the class-average classification accuracy. Note that our method only considers the segmentation of the training images. At test time, we apply a pipeline consisting of two steps: class-independent segmentation and image classification.

Class-independent segmenter. We propagate co-segmented masks of training images onto test images, which cannot be co-segmented as these are to be classified individually. Based on the co-segmented images, we train a class-independent segmenter in the same way as described at the beginning of Sec. 4.1.2. This class-independent segmenter can then be used to give binary segmentation masks on test image superpixels and therefore accurate foreground segmentations during testing.

Final image classifier. We train a 1-vs-all linear SVM classifier in the same way as described in Sec. 4.1.2. Unlike in BiCoS, the classifier is not trained on the output of the co-segmentation directly, but on the segmentations produced by the newly trained class-independent segmenter, with the rationale to make the data pre-processing for the final classifier training as similar to the data pre-processing at test-time as possible. During test time, we simply first apply the class-independent segmenter to each test image, and classify the image using the 1-vs-all linear SVM on the features from the foreground area.

The overall pipeline can be seen as an improved version of BiCoS. A direct comparison can be found in Sec. 4.2.1.

The BiCoS results in the second row in each comparison table use the new initialization described in Sec. 4.1.1. In our experiments with the particular datasets, the proposed initialization scheme did not give significant advantage over a simpler initialization with a central box (as in BiCoS). We, however, stick to the new initialization as it does not rely on the assumption that most objects are large and centered in images, and is, therefore, more general.

In the experiments we compare our system to previous methods from three different domains: (1) co-segmentation or, if available, ground truth segmentations/bounding boxes, (2) methods that segment each image individually at training and at test time and (3) classification without any segmentation.

During all evaluations, TriCoS starts at the category level, moves to image level and iterates one whole loop before stopping at the image level.

4.2.1 Caltech-UCSD Birds 200-2010

We show our evaluation in Tab. 4.1 while qualitative examples can be found in Fig. 4.3 and Fig. 4.4. For the experiment with GrabCut, GrabCut is directly applied on both training and test images using the initialization introduced in Sec. 4.1.1. For the experiments using ground truth bounding box or foreground/background segmentations, we do not use the hand annotated data during test time. Instead, we use ground truth data from the training images to train the class-independent segmenter and apply it to the test images for the classification.

Most importantly, TriCoS outperforms BiCoS, justifying our class-discriminative co-segmentation approach. We observe a huge decrease in accuracy if all extracted features are used for classification ("no segmentation"), which once again shows the importance of object localization in classification in general. Moreover, we observe that only bounding box is not sufficient as using accurate ground truth foreground segmentation has a clear gain over using ground truth bounding box.

The BiCoS in combination with our class-independent segmentation and the new classification method clearly shows our new pipelines being significantly superior than the one proposed in BiCoS. The majority of the gain is achieved using Fisher encoding by Perronnin et al. (2010) instead of LLC by Wang et al. (2010).

While TriCoS has a significant gain over hand annotated bounding boxes, it is still not as good as hand annotated pixel-level foreground segmentation. This is most likely caused by the insufficient amount of training data, as discriminative information can be discarded by TriCoS if it does not occur often enough in the training set.

The baseline accuracies provided by Welinder et al. (2010b) uses ground truth foreground segmentation both during training and testing, in combination with a multiple



Figure 4.3: Bird segmentations: (top) original images, (middle) BiCoS outputs, (bottom) TriCoS outputs. The three columns on the left show that TriCoS finds the yellow head of the bird to be not discriminative and cuts it off during the co-segmentation. The right columns show that several images of one bird species contain red plastics, which misleads BiCoS to classify the red plastics as foreground. TriCoS finds the red plastics to be confusing for the classification and classifies them as background.

kernel learning approach for color and shape features, whereas Khan et al. (2011) mainly focus on the way to better combine those features.

4.2.2 Stanford Dogs

As discussed in Sec. 2.8, the Stanford Dogs dataset originally consists of 120 dog species and 20580 images. There are 100 images per class for training and about 80 images for testing. In order to get a similar number of training images per class as the bird and flower datasets, we randomly selected 30 images per class for training, while keeping the test set as it is.

The experimental results on this sub-sampled dog dataset are shown in Tab. 4.2, which leads to similar conclusions as in Tab. 4.1. GrabCut alone seems to make a lot of mistakes, and therefore achieves lower accuracy than not using any segmentation.

Coseg	Class-indep. seg.	Final classifier	Acc.	GT used?
TriCoS (this work)	ours	ours	**25.5**	no
BiCoS	ours	ours	23.7	no
BiCoS-MT	ours	ours	24.2	no
-	GrabCut	ours	19.2	no
-	no segmentation	ours	14.1	no
BiCoS	BiCoS	BiCoS	15.7	no
BiCoS-MT	BiCoS	BiCoS	16.1	no
GT	ours	ours	**26.7**	seg/trn
GT	ours	ours	22.7	bb/trn
-	-	Welinder et al. (2010b)	19.0	seg/trn+tst
-	-	Yao et al. (2011)	19.2	bb/trn+tst
-	-	Khan et al. (2011)	22.4	unsure

Table 4.1: Comparison on the Caltech-UCSD Birds 200-2010 dataset (15 training images per class). The last column shows whether ground truth (GT) information is used during either training or testing. seg: accurate foreground/background segmentation, bb: bounding box. Khan et al. (2011) did not mention the usage of GT during test time, however directly compares their results to methods that make use of the GT.

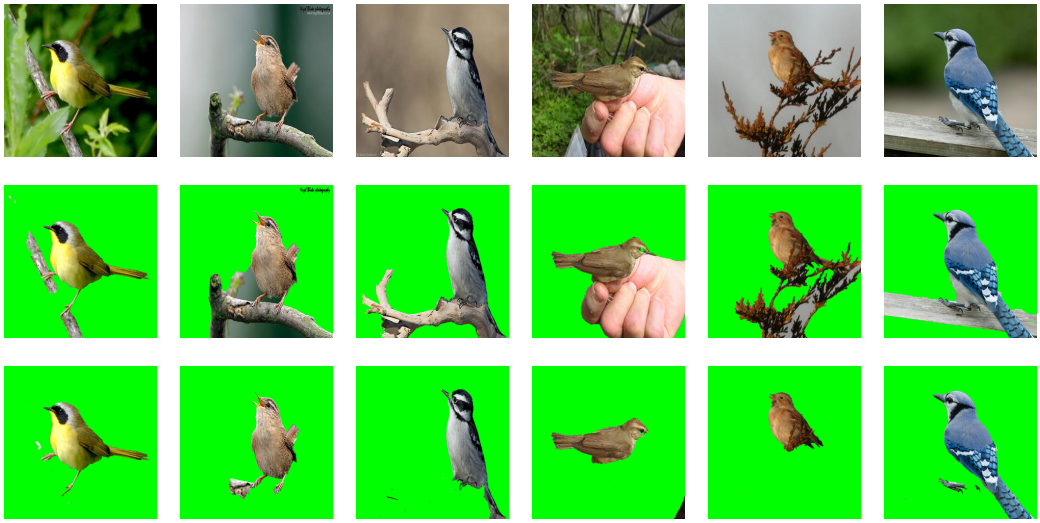


Figure 4.4: More Bird segmentations: (top) original images, (middle) BiCoS outputs, (bottom) TriCoS outputs. The most noticeable difference between BiCoS and TriCoS is the removal of tree branches of leaves. These often appear in many images in one category, so that BiCoS segments them as foreground. But they appear across categories as well and are therefore not class-discriminative and are discarded in TriCoS.

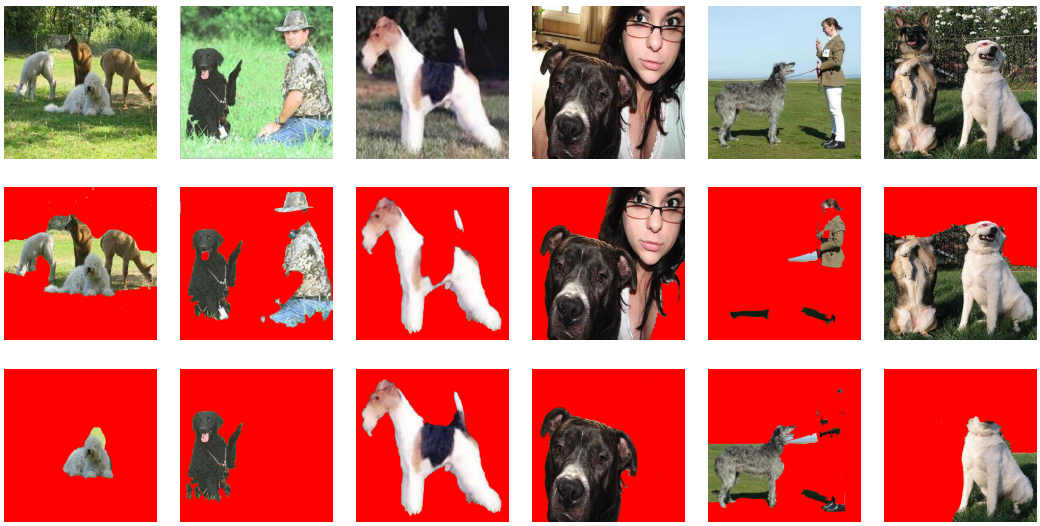


Figure 4.5: Dogs segmentations: (top) original images, (middle) BiCoS outputs, (bottom) TriCoS outputs. If an image patch only appears once in the co-segmentation set, BiCoS is unable to flip its binary label from its initialization. However, using discriminative information across the entire dataset, TriCoS is sometimes able to correct this.

Coseg	Class-indep. seg.	Final classifier	Acc.	GT used?
TriCoS	ours	ours	**26.9**	no
BiCoS	ours	ours	25.7	no
-	GrabCut	ours	17.3	no
-	no segmentation	ours	21.0	no
-	-	Khosla et al. (2011)	14.5	no
GT	ours	ours	26.0	bb/trn

Table 4.2: Comparison on the Stanford Dogs dataset. Note that TriCoS performs better than BiCoS and hand-annotated bounding boxes.

Coseg	Class-indep. seg.	Final classifier	Acc.	GT used?
TriCoS	ours	ours	85.2	no
BiCoS	ours	ours	85.5	no
-	GrabCut	ours	81.7	no
-	no segmentation	ours	81.5	no
BiCoS	BiCoS	BiCoS	79.1	no
BiCoS-MT	BiCoS	BiCoS	80.0	no
-	-	Kanan and Cottrell (2010)	71.4	no
-	-	Khan et al. (2011)	73.3	no
GT	Nilsback	ours	85.6	seg/trn
GT	Nilsback	BiCoS	76.8	seg/trn
GT	Nilsback	Nilsback and Zisserman (2008)	76.3	seg/trn

Table 4.3: Comparison on the Oxford Flowers 102 dataset. TriCoS, BiCoS and Nilsback's supervised segmentation method give more or less the same accuracy.

However, both TriCoS and BiCoS remedy these errors. Examples can be found in Fig. 4.5.

4.2.3 Oxford Flowers 102

The evaluation is shown in Tab. 4.3. Nilsback and Zisserman (2008) use hand annotated pixel-level segmentations to train a geometry-based segmentation model that is optimized for flowers and a multiple kernel learning (MKL) in conjunction with a non-linear SVM for classification.

Unfortunately, TriCoS performs worse than BiCoS and Nilsback's supervised segmentation method, but only by a small margin. We argue that the Fisher vector encoding of the densely sampled and SIFT features already saturates the performance so that no more gain can be achieved with better segmentation.

4.3 Discussion

We have demonstrated the importance of class-discriminative co-segmentation by comparing TriCoS to its predecessor BiCoS on the challenging Caltech-UCSD Birds, Stanford Dogs and Oxford Flowers 102 datasets, where TriCoS outperforms BiCoS in two of the three cases and is only worse by a very small margin on the flowers.

Moreover, we use an interest region detection-based method to initialize the system. In this way, it does not require any guess of the location of the foreground object as is done in BiCoS.

TriCoS outperforms hand-annotated bounding boxes on both birds and dogs, despite only requiring image class labels and no additional annotation. Our experimental results agree with previous findings that accurate foreground segmentation is of great importance in fine-grained classification systems.

Last but not least, using our state-of-the-art classification system in conjunction with TriCoS, we achieve classification accuracy improvement over previously published results by a margin of 3.1%, 12.4% and 5.2% on the birds, dogs and flowers at the time of publication.

Chapter 5

Classification with symbiotic model

In previous chapters, we looked at how foreground segmentation improves fine-grained categorization performance. This conclusion was also drawn by Nilsback and Zisserman (2006), Parkhi et al. (2012), who demonstrated that at the subordinate category level, the background is seldom discriminative and it is beneficial to segment out the foreground and to discard the visual information in the background.

Another line of recent works in fine-grained categorization assumed that it often happens that two similar classes can only be distinguished by the appearance of localized and very subtle details (such as the color of the beak for bird classes or the shape of the petal edges for flower classes). With generic classification approaches these fine differences often get *swamped* by the bulk of the image, whenever encoding of the image content into a visual signature of some sort is performed. Therefore, Bourdev et al. (2011), Parkhi et al. (2012), Yang et al. (2012), Yao et al. (2011), Zhang et al. (2012) focused on the localization of these discriminative image parts as a precursor to categorization. Once the discriminative parts are localized, they are encoded into separate parts of the visual signature, enabling the classifier to pick up on the fine differences

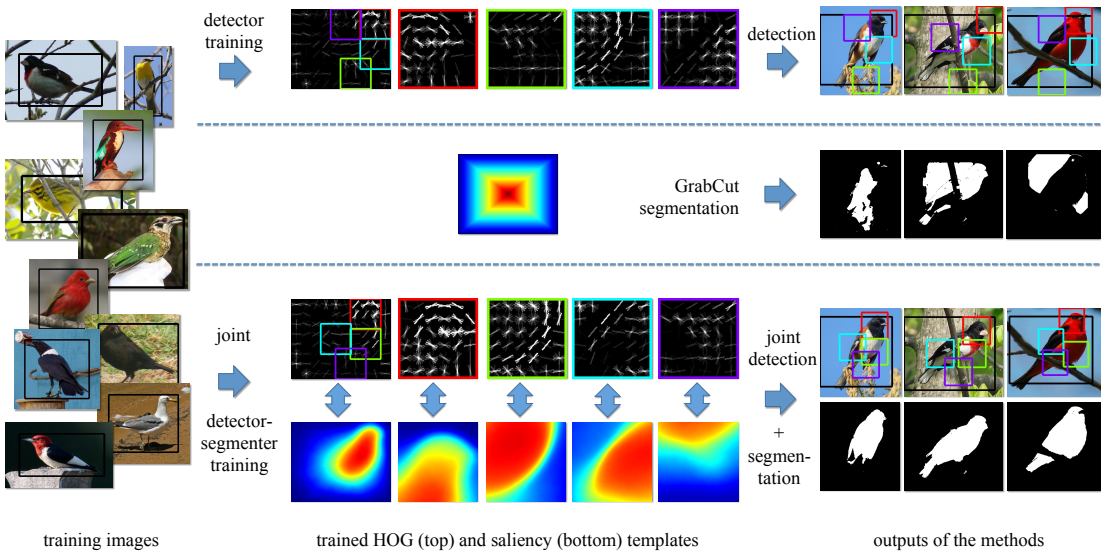


Figure 5.1: Best viewed in color. We demonstrate our system using images from the Caltech-UCSD Bird dataset. Left: examples of the training images. Black frames indicate the provided ground truth bounding box. Top: a stand alone unsupervisedly trained Deformable Part Model (DPM) with its results to the right. Center: GrabCut automatically segments the images using the outside of the given bounding box as background and a prior saliency map for the region inside the bounding box. Bottom: our approach, which trains a symbiotic set of detector templates and saliency maps and apply them jointly to images. As a result a considerable improvement in segmentation accuracy, part-localization consistency, and the ultimate fine-grained classification accuracy is achieved. (The saturation in the output images is reduced for illustration only).

in those parts.

In the light of all this evidence, it is natural to investigate the combination of part localization and foreground segmentation for fine-grained categorization, and their interaction in combination is the topic of this work. Our least surprising finding (which nevertheless translates into a very competitive categorization system) is that a simple concatenation of visual signatures, provided by a system that performs part localization and by a system that performs foreground segmentation, leads to improved categorization accuracy (as compared to classifiers operating with each of the two signatures individually).

More interestingly, we demonstrate that the accuracy of fine-grained categorization can be further boosted if part localization and foreground segmentation are performed together, so that the outcomes of both processes aid each other. As a result, better segmentation can be obtained by taking into account part localizations, and, likewise, more semantically meaningful and discriminative parts can be learned and localized if foreground masks are taken into account. We implement this feedback loop with an alternating energy minimization. The resulting *symbiotic* system results in a better categorization performance compared to the system obtained by a mere concatenation of two visual signatures (discussed above). Overall, our symbiotic system outperforms the previous state-of-the-art on all datasets considered in our experiments (both the 2010 and 2011 version of Caltech-UCSD Birds, and Stanford Dogs as discussed in Sec. 2.8) by a substantial margin. This symbiotic system is the main contribution of the chapter.

As a coda, we investigate the gains in performance by using additional annotation, and show that although training performance is near saturation, significant improvements are still possible at test time; thus confirming similar findings (e.g. a human in-the-loop by Branson et al. (2010)) in recent literature.

5.1 Symbiotic Segmentation and Localization

We start with an overview of our system. The system is built around a model of the base category (e.g. bird) which includes a deformable part model $\mathcal{W}$ and a set $\mathcal{S}$ of saliency maps associated with each part (*a saliency model*). Given a pretrained model and an input image I that needs to be encoded, the base-category model is first fitted to

this image via the (approximate) minimization of the following two-term functional:

$$E(\mathbf{p}, \mathbf{f}, \mathbf{c}|\mathcal{W}, \mathcal{S}, I) = E^{DPM+}(\mathbf{p}|\mathbf{f}, \mathcal{W}, \mathcal{S}, I) + E^{GC+}(\mathbf{f}, \mathbf{c}|\mathbf{p}, \mathcal{S}, I) \tag{5.1}$$

Here, the minimization is performed over the part localizations $\mathbf{p}$, the foreground mask $\mathbf{f}$, and the color distributions of the foreground and background $\mathbf{c}$. The recovered part localizations $\mathbf{p}$ and the foreground segmentation $\mathbf{f}$ are then used to encode the image content into a highly-discriminative visual signature as discussed in the next section. The two terms in (5.1) correspond to the popular models we build upon: E^{DPM+} denotes an augmented Deformable Part Model (DPM) by Felzenszwalb et al. (2010) energy; while E^{GC+} denotes an augmented GrabCut by Rother et al. (2004) energy. We now discuss the variables in (5.1) in more detail.

Deformable part model $\mathcal{W} = \{\mathbf{w}_t\}$: here, we use a multi-component Deformable Part Model (DPM) by Felzenszwalb et al. (2010) consisting of several mixtures of parts, where each part is described by a HOG template and a geometric location prior. We denote the number of mixture components N, and the number of parts in each component M. We omit extra indices for different mixture components and use $\mathbf{w}_0$ to describe the root HOG template for each component. $\mathbf{w}_t$ then denotes the parameters of the t-th part (the HOG template and the geometric prior).

Saliency model $\mathcal{S} = \{\mathbf{s}_t\}$: we associate with the root and each part $\mathbf{w}_t$ of the deformable part model an extra map $\mathbf{s}_t$ that corresponds to the foreground probability. Pixels of this saliency map thus have values between 0 and 1, with 1 indicating a high chance of the pixel being foreground and 0 otherwise. As both terms in (5.1) include the minimization of the disparity between the saliency maps $\mathcal{S}$ and the foreground mask $\mathbf{f}$ (as explained below), the saliency maps establish the connection between the two terms. An example of a set of saliency maps is shown in the center of the bottom

row of Fig. 5.1.

Part localizations $\mathbf{p}$: this variable denotes the location (the bounding box coordinates) of the detected parts in an image. Only one mixture component is active for a single image. The localization of a particular part template $\mathbf{w}_t$ is denoted $\mathbf{p}_t$. The part localizations are shown as colored bounding boxes in the output images of Fig. 5.1.

Color distributions $\mathbf{c} = \{\mathbf{c}_0, \mathbf{c}_1\}$: following GrabCut by Rother et al. (2004), we model the distribution of colors in the image in the foreground and the background as Gaussian mixtures in the RGB space (denoted $\mathbf{c}_1$ and $\mathbf{c}_0$ respectively).

Foreground segmentation $\mathbf{f}$: this map assigns each pixel the value 1 if it is assigned to the foreground, and 0 if it is background. Examples of the binary segmentations are shown as binary maps in Fig. 5.1.

Note that $\mathbf{p}$, $\mathbf{f}$, $\mathbf{s}$ are specific to an image I, while $\mathcal{W}$ and $\mathcal{S}$ are global parameters describing the base-level category (e.g. bird or dog). These parameters can be learned from a dataset $\mathcal{I}$ of images containing instances of this base category as discussed in Sec. 5.1.2.

5.1.1 Model fitting

We fit the model $(\mathcal{W}, \mathcal{S})$ to an image by alternating the optimization of the two terms in (5.1) (so that their sum is also approximately minimized). These terms are defined and minimized as follows.

Augmented DPM energy. The first term in (5.1) finds the best part localization $\mathbf{p}$

given the DPM $\mathcal{W}$ and the foreground segmentation $\mathbf{f}$:

$$E^{DPM+}(\mathbf{p}|\mathbf{f},\mathcal{W},\mathcal{S},I) = E^{DPM}(\mathbf{p}|\mathcal{W},I) + \sum_{t=1,2,3,\cdots} T(\mathbf{p}_t,\mathbf{s}_t,\mathbf{f}) \tag{5.2}$$

$$T(\mathbf{p}_t,\mathbf{s}_t,\mathbf{f}) = \begin{cases} 0, \text{ if } \frac{|\mathbf{f}[\mathbf{p}_t]\cap[\mathbf{s}_t>0.5]|}{|\mathbf{f}[\mathbf{p}_t]\cup[\mathbf{s}_t>0.5]|} \geq \alpha \\ +\infty, \text{ otherwise} \end{cases} \tag{5.3}$$

where $E^{DPM}(\mathbf{p}|\mathcal{W},I)$ is a standard DPM detection functional (i.e. negative of the detector score) from Felzenszwalb et al. (2010), $\mathbf{f}[\mathbf{p}_t]$ describes a subimage of the segmentation $\mathbf{f}$ given by the bounding box from $\mathbf{p}_t$ resized to match the size of $\mathbf{s}_t$; $[\mathbf{s}_t > 0.5]$ is a binary mask obtained by thresholding the part t saliency map $\mathbf{s}_t$ at 0.5; $|A|$ means the number of pixels of A; $A \cap B$ the intersection between A and B, and $A \cup B$ their union. The term (5.3) thus constrains the localizations of the parts to locations in the image where the saliency masks $\mathbf{s}_t$ placed according to the localizations $\mathbf{p}_t$ are in agreement with the subparts of the foreground mask $\mathbf{f}$ they are superimposed onto. This agreement is measured with the intersection-over-union overlap measure that has to be higher than a prespecified parameter α. We investigate the dependence of our system on α in the experimental section.

The term $E^{DPM}(\mathbf{p}|\mathcal{W},I)$ can be optimized using an off-the-shelf DPM detector by Felzenszwalb et al. (2010), which places the HOG-templates over the image to maximize the correlation with the HOG descriptor of the image, while also optimizing the pairwise geomtric consistency between the root and the parts. To perform the optimization subject to the constraint (5.3), we let the DPM detector return a long list of possible part localizations and choose the one with the highest detection score that also fits the overlap constraints for all parts[1].

[1] An exact solution would be to modify unary terms inside the DPM detector, which is certainly possible, but was not needed in practice.

Augmented GrabCut energy. The second term in (5.1) is an augmented GrabCut energy that scores segmentation maps $\mathbf{f}$ and the foreground/background color models $\mathbf{c}$ given the part locations $\mathbf{p}$ and the pre-trained saliency templates $\mathcal{S}$:

$$E^{GC+}(\mathbf{f},\mathbf{c}|\mathbf{p},\mathcal{S},I) = \sum_x U(x|\mathbf{f},\mathbf{c},\mathbf{p},\mathcal{S},I) + \\ +\lambda \sum_{(x,x')} |\mathbf{f}(x)-\mathbf{f}(x')|\ V\left(I(x)-I(x')\right) \tag{5.4}$$

$$U(x|\mathbf{f},\mathbf{c},\mathbf{p},\mathcal{S},I) = \beta\, U^{\text{GMM}}\left(I(x)|\mathbf{c}_{\mathbf{f}(x)}\right) + \\ + \frac{\sum_{t\geq 0} \mathbf{p}_t(x)\cdot(\mathbf{s}_t(x)-0.5)}{\sum_{t\geq 0}\mathbf{p}_t(x)+\epsilon} \tag{5.5}$$

where x indicates pixels in the image, $I(x)$ denotes an RGB value at pixel x, (x,x') spans all pairs of adjacent pixels, t spans all parts and the root ($t=0$). V is the binary Ising potential weighted according to the intensity gradient between the two pixels (same weights as in Rother et al. (2004)). ϵ is a small positive constant to prevent dividing by zero.

The unary potential U consists of a traditional term U^{GMM} which is equal to the log-likelihood of $I(x)$ under the Gaussian mixture $\mathbf{c}_{\mathbf{f}(x)}$, where $\mathbf{f}(x)$ is the foreground/background label of pixel x. This unary term retained from Rother et al. (2004) is augmented with the second term which is the mean of the saliency maps over all parts covering a pixel (including the root saliency map). $\mathbf{p}_t(x)$ is a binary indicator that has the value 1 if the location of the t-th part $\mathbf{p}_t$ covers the pixel x, and 0 otherwise. The effect of this term is to encourage the computed segmentation, $\mathbf{f}$, to align with the mean of the saliency maps. β controls the balance between the two unary terms and its effect is also evaluated during the experiments.

Despite the addition of the extra unary term, the augmented GrabCut energy (5.4) can be solved in the same way as in Rother et al. (2004) by alternating between graph cut

minimizations by Boykov and Kolmogorov (2004) with Gaussian Mixtures refittings. Unlike in the work by Rother et al. (2004), we initialize the alternations by setting β to zero (i.e. disabling U^{GMM}) and running graph cut.

Implementation details. To fit a model $(\mathcal{W}, \mathcal{S})$ via alternating minimizations (5.2) and (5.4), we need to find a way to initialize the process. One way to do this is to minimize the unconstrained version of (5.2) (i.e. with constraints (5.3) disabled), which gives part locations $\mathbf{p}$. However, we found that the DPM detector performs poorly without the help of the segmentation, that is why we initialize the alternations by first optimizing a reduced version of the augmented GrabCut energy. In this case, only the root saliency map $\mathbf{w}_0$ of each mixture component, resized to fit the annotated object bounding box, is taken into account (while non-root parts are ignored). We then proceed with iterating between part localization and segmentation.

Furthermore, while optimizing for (5.2), we have prior knowledge that the root filter ideally should overlap strongly with the user-provided bounding box. Therefore, given several DPM detections (corresponding to different mixture components) we pick the one with the highest detection score (lowest value of (5.4)), which also has a root filter overlap with the bounding box greater than 0.7 (intersection-over-union). If no detection can be returned in this way, we omit this constraint[2]. Finally, when performing optimization (5.2), we omit the constraints (5.3) if the segmentation $\mathbf{f}$ is empty (which may happen with GrabCut) or if α is too high and causes no detection satisfying (5.3) to be returned.

[2]This can be regarded as an extra term in (5.4) assigning a large but finite penalty to all detections that violate this overlap constraint

5.1.2 Learning the Model

The part templates $\mathcal{W}$ and $\mathcal{S}$ are trained using a set $\mathcal{I}$ of training images. We generally follow the approach of Felzenszwalb *et al.* for learning a deformable part model $\mathcal{W}$ (c.f. section 5.2 in Felzenszwalb et al. (2010)). Thus, we first learn the HOG templates for root filters in the mixture components via latent SVM training (we use a separate unrelated dataset as a source of negative examples; we constrain the root filters to overlap with user-provided boxes by at least 70%). At the same time, we run GrabCut on the positive examples (using bounding box annotations), and estimate the saliency maps $\mathbf{s}_0$ corresponding to root filters (by averaging corresponding foreground masks) before part discovery is performed.

We then use a standard DPM approach to discover repeatable parts with an important modification. In Felzenszwalb et al. (2010), "interesting" parts are discovered greedily by covering the high-energy (large gradient magnitude) parts of the root HOG-template. In our case, we modify this interestingness measure by multiplying the HOG magnitude by the saliency maps estimated for each filter. In this way, we constrain the discovery process to discover part which overlap substantially with the foreground (as estimated by a GrabCut). We found this modification to be important to make the learned parts consistent with our model (5.1), but also to discover more semantically consistent parts. We come back to the issue of unsupervised part discovery in the experiments section.

Once the parts and their HOG-templates are discovered, we can apply learned templates to the training images ((5.2) without the constraint (5.3)). The saliency maps $\mathbf{s}_t$ for each part are then once again learned by averaging the segmentation masks under the placed part boxes. We found such averaging-based aggregation to be resilient to suboptimal initialization of part placement and/or imperfect GrabCut results of some

images. Generally, the learned model can be further optimized through alternation-based optimization, however we found that this does not consistently improve categorization accuracy.

5.2 Related Work

There is a line of work stretching back over a decade on the interaction between segmentation and detection. In early works, object category detectors simply proposed foreground masks (Borenstein and Ullman (2002), Leibe and Schiele (2003)). Later methods used these masks to initialize graph-cuts based segmentation by Boykov and Kolmogorov (2004) that could take advantage of image specific color distributions, giving crisper and more accurate foreground segmentations (Kumar et al. (2005), Levin and Weiss (2006)).

The *poselet* line of research by Bourdev and Malik (2009) is a more recent addition to this line of work. It investigates the benefit of keypoint annotation when learning part detectors, and subsequently uses these poselet part detectors with foreground masks for object category detection and segmentation (Brox et al. (2011)). Poselets have also been used for the task of fine-grained visual categorization (Bourdev et al. (2011), Zhang et al. (2012)). The related *birdlets* method by Farrell et al. (2011) used volumeric primitives (two 3D ellipsoids) in place of 2D boxes to describe parts. Overall, unlike our method, the poselet works rely on additional user supervision (rather than the segmentation process) to guide the discovery of meaningful and discriminative parts.

The closest work to ours is that of Yang et al. (2012) that also accomplishes unsupervised learning of a deformable part model in order to find discriminative parts

	Birds10	Birds11	Dogs
Yao et al. (2011)	19.2	-	-
TriCoS	25.5	-	-
Yang et al. (2012)	28.2	-	38.0
Zhang et al. (2012)	-	28.2	-
Symb*	41.0	55.7	42.4
Symb	**45.1**	**59.5**	**44.6**

Table 5.1: Performance on the three fine-grained categorization datasets. Symbotic system ("Symb") consistently outperforms any published results by a large margin. "Symb*" shows the accuracy of the system, for which classifiers were trained on the sets not augmented by left-right mirroring. Zhang et al. (2012) have confirmed to have used mA rather than mAP in their paper.

for fine-grained categorization. An earlier method had used the image as a bounding box for learning parts for scene classification by Pandey and Lazebnik (2011). We demonstrate, however, that the parts discovered in the absence of supervision are less discriminative than those discovered with the help of the segmentation process as is done in our method.

In summary, the interplay between part localization and segmentation has not been investigated in the context of fine-grained categorization (to the best of our knowledge). By exploiting this interplay, the proposed approach is able to achieve a significant improvement in the categorization accuracy.

5.3 Experimental Results

The empirical evaluation is carried out on three benchmark datasets for fine-grained image classification – the Caltech-UCSD Birds 2010 and 2011, and Stanford Dogs.

In all experiments, we make use of the provided bounding boxes around the object during both training and testing, as do most of the approaches we compare to. During

preprocessing, all images are first resized such that the bounding box has the longest dimension equal to 300 pixels. Images are cropped to include the bounding box together with a maximum 50 pixel wide strip around the box. This is important for any GrabCut-related steps as the background can be better estimated using the strip. Each dataset is augmented with the left-right mirrored versions of its training images, as this typically yields a 3-5% improvement over not doing so (for reference we also give final results without such mirroring).

The symbiotic model is fitted to every train and test images using 5 alternation iterations (the convergence is observed after 3 iterations in most cases). It takes about 10 seconds to fit the model to a typical image. The parameters α and β were set to 0.1 and 0.5 respectively (we find the final accuracy to be not too sensitive to the variation of these parameters). The choice of the parameters M, N is discussed below.

Classification Process. The symbiotic model outputs one binary segmentation and a set of detected part bounding boxes for a given image. Descriptors are extracted from each of them individually, i.e., one feature vector, $\mathbf{x}^{SEG}$, for the foreground region in the segmentation, and a feature vector for each of the parts apart from the root template. A feature vector is not included for the root template as it would be too redundant with $\mathbf{x}^{SEG}$. We denote the concatenation of all part features as $\mathbf{x}^{PART}$. If the final feature dimension is D, we use $D/2$ for $\mathbf{x}^{PART}$ and the other $D/2$ for $\mathbf{x}^{SEG}$.

Each region (i.e. the foreground and the box of each part) is encoded by: (1) LLC-encoded (Wang et al. (2010)) Lab color histogram vector, and (2) Fisher vector (Perronnin et al. (2010)) aggregating SIFT features (the implementation Chatfield et al. (2011) was adopted). Both features are ℓ_2 normalized after encoding and then concatenated. Finally, after another $\ell 2$ normalization, $\mathbf{x}^{SEG}$ and $\mathbf{x}^{PART}$ are concatenated. A conventional multi-class 1-vs-rest linear support vector machine (SVM) is used for the

final fine-grained classification (the regularization strength is set by cross-validation).

To encode the foreground, we use a k-means Lab vocabulary of size 512, and a SIFT GMM with 128 components. The resulting feature vector $\mathbf{x}^{SEG}$ is has 20992 dimensions. When encoding parts, we choose the size of the vocabulary so that $\mathbf{x}^{PART}$ and $\mathbf{x}^{SEG}$ are always the same length (i.e. 20992 dims each), no matter how many parts and mixture components are used.

Performance Measures. We evaluate the categorization performance of several baselines and variations of our system, and report two performance measures for this: (1) **Mean accuracy (mA)**: for each class we measure the proportion of test images of the class that are classified correctly (as belonging to this class). The proportion is then averaged over all classes. This measure is the one used in most previous works. (2) **Mean average precision (mAP)**: For each class, we evaluate the SVM score of the class' classifier for the entire dataset. Once the dataset is ordered by decreasing score, the average precision (AP) of the returned list is computed (i.e. the area under the precision-recall curve). The AP numbers are averaged over all classes. This measure is more relevant than mean accuracy (mA) for some applications (e.g. Web image search).

5.3.1 Results and Comparisons

Overall, our complete system substantially surpasses all previously published results on all three datasets (Tab. 5.1). The models learned by the symbiotic system can be seen in Fig. 5.1 and Fig. 5.2. The relative importance of the model components, as well as the net effect of the "symbiosis" between the segmentation and part localization, are evaluated in Tab. 5.2.

ID	Model fitting	Descriptor	Birds11		Birds10		Dogs	
			mA	mAP	mA	mAP	mA	mAP
1	taking whole bounding box	$\mathbf{x}^{SEG}$	40.7	32.5	27.9	20.0	39.7	33.0
2	GrabCut segmentation	$\mathbf{x}^{SEG}$	51.1	40.4	39.3	26.7	42.2	33.9
3	Symbiotic model fitting	$\mathbf{x}^{SEG}$	54.7	42.2	40.1	26.4	43.6	35.5
4	DPM part localization	$\mathbf{x}^{PART}$	38.6	27.3	26.7	15.1	22.2	17.0
5	Symbiotic model fitting	$\mathbf{x}^{PART}$	47.7	38.5	37.4	28.6	31.8	26.7
6	GrabCut + DPM (independent)	$[\mathbf{x}^{SEG}; \mathbf{x}^{PART}]$	54.4	46.6	41.7	30.4	41.3	35.8
7	Symbiotic model fitting	$[\mathbf{x}^{SEG}; \mathbf{x}^{PART}]$	**59.5**	**51.2**	**45.1**	**35.7**	**44.6**	**39.5**

Table 5.2: A detailed comparison with baselines (no model fitting, segmentation only, part localization only). Note that segmentations produced by the symbiotic model allow for more discriminative signatures than those produced with GrabCut alone (#3 vs. #2), while parts learned and localized by the symbiotic model are more discriminative than those learned and localized by DPM (#5 vs. #4). Finally, categorization with full signatures produced by symbiotic model is better than categorization based on the concatenation of segmentation-based and part-based signatures produced by GrabCut and DPM run independently (#7 vs #6). All these improvements are due to the fact that part localization and segmentation processes assist each other within the proposed symbiotic model.

In the table, we compare the categorization accuracy of the systems resulting from applying GrabCut alone or DPM part localization alone, while keeping the rest of parameters (initialization, feature encoding, etc.) fixed. Notably, a considerable improvement over a GrabCut-based system (line 2) is observed even if we only use the segmentation-based descriptor $\mathbf{x}^{SEG}$ in our system (line 3), thus highlighting that segmentations obtained by our systems are better (at least for further categorization). Likewise, the same improvement is observed for part localization, when the segmentation process is used to aid part discovery and fitting, as opposed to using a DPM model on its own (line 5 vs line 4). Finally, and most importantly, the symbiotic system improves considerably in all measures on all three datasets when compared to the system that gets the same visual signature by running the classification and the part localization processes independently and concatenating the corresponding signatures (line 7 vs line 6).

The interaction between the segmentation and the part localization processes are further shown in Fig. 5.3 and Fig. 5.4. Note, that in the case of Fig. 5.3, we used the same deformable part model $\mathcal{W}$ (learned within the symbiotic model) but evaluated it with and without the help of the segmentation process. In Fig. 5.4, we simply compare the segmentations obtained by our system and by GrabCut. In both cases, it can be seen how symbiosis between the part localization and the segmentation improve the performance of each process.

We note that the improvement over the baselines (especially over the GrabCut baseline) is smaller for the Dogs dataset rather than for the Birds datasets. We attribute this fact to a greater pose variability for dogs that is harder to cope with for the deformable parts model. At the same time, dogs have a nice roundish shape which makes them very appropriate for GrabCut (so that the aid from the parts localization is not needed

in most cases). The performance of the DPM on dogs can be potentially improved by having more mixture components. However, as discussed below, it might hurt the generalization in the categorization step, and especially since we keep the feature dimension of $\mathbf{x}^{PART}$ the same. Post-processing as suggested in Zhang et al. (2012), may also be useful in this case.

Influence of the parameters. We have further evaluated the influence of the size of the deformable parts model on the categorization accuracy, namely N (number of mixture components) and M (the number of parts per component). As discussed in Felzenszwalb et al. (2010), in the context of detection a larger N increases the non-linearity of the model while also increasing data fragmentation. Meanwhile, an M has to strike a balance between having too many parts some of which are not detectable and having too few parts, which will make the detector less powerful.

In the context of building the base-class model for fine grained classification, M and N have some additional meaning. While large N may also increase the data fragmentation within some subordinate classes, potentially having large N may also attribute different subordinate classes to different components, thus making the categorization easier. At the same time, picking the value for M faces the usual choice between feature repeatability and discriminativity. The more parts the model has, the more discriminative information it can provide into $\mathbf{x}^{PART}$. However, it becomes more difficult to detect parts repeatedly at the same semantic "locations".

We mainly selected these 2 parameters based on visual feedback during the training stage. But we also did some quantitative evaluation using different settings for the Bird 2011 dataset, as shown in Tab. 5.3. Overall, for the bird datasets, we chose $N = 1$ and $M = 4$, while $N = 2$ and $M = 4$ seems to be more reasonable for the dogs dataset[3].

[3]We use the functionality of the code Felzenszwalb et al. (2010) which allows to apply each DPM mixture component twice (once with mirroring and once without) during training and test.

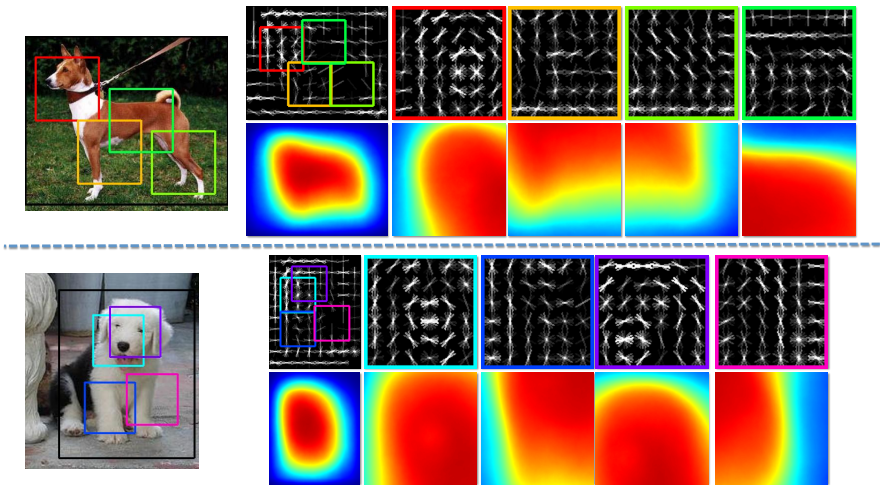


Figure 5.2: Trained $\mathcal{W}$ and $\mathcal{S}$ for the dogs dataset. After learning a symbiotic model, the two mixture components (shown side-by-side) happen to correspond to a more profile and a more frontal views.

$N \times M$	1×8	1×4	1×2	1×1	2×4	2×2	4×2
mA	58.3	**59.5**	57.5	57.8	57.1	55.3	53.4
mAP	**53.2**	51.2	48.2	44.3	51.6	47.3	45.5

Table 5.3: Effect of different choice of N and M evaluated on the Caltech-UCSD Birds 2011. The loss in accuracy with higher number of mixture components indicates that the complexity of a bird pose does not justify more than one mixture component in our model.

5.3.2 Experiments with Extra Annotation

Looking at Tab. 5.2, one can notice that generally the segmentation-based signatures outperform part-localization based signatures very considerably. In fact, the latter often lead to categorization accuracy even below the performance of a simpler model (which extracts the signature by looking at the entire bounding box). Only by combining segmentation and part localization (lines 6 and 7 in the table) can we see a consistent benefit from having part localization in the system. One natural question is whether the performance of part localization is inherently limited or is this a problem with segmentation-supervised and, particularly, unsupervised discovery?

To address this question we used the extensive annotations available for the Birds 2011. Apart from the bounding boxes, there are 15 part locations annotated per image. These parts include, e.g. beak, eyes, feet, etc. Given these annotations, we evaluated what would be achievable if we move away from unsupervised parts discovery and localization to supervised parts learning, or even using supervised parts localization during both training and testing (the latter would correspond to the scenario of asking the user to annotate some parts in the test image, thus approaching the human-in-the-loop approach investigated in Branson et al. (2010)).

For simplicity, we considered a single part (a head of a bird), which leads to a setup that is similar to Parkhi et al. (2012). Thus, we first made use of the annotated head locations and trained a head detector (which was a mixture of HOG templates). This detector was used to locate heads in bird images. The first 2 experiments in Tab. 5.4 correspond to this setup. In a second set of experiments, we used the ground truth (rather than detected) head locations at all stages. Through these batch of experiments we followed the rest of our pipeline (i.e. we learn a saliency map for the head, we use head detector to aid GrabCut, etc.).

localization	Descriptor	GT	mA	mAP
det. head	$\mathbf{x}^{PART}$	trn	52.4	31.9
GC + det. head	$\left[\mathbf{x}^{SEG}; \mathbf{x}^{PART}\right]$	trn	61.0	51.2
GT head	$\mathbf{x}^{PART}$	trn/tst	60.2	45.5
GC + GT head	$\left[\mathbf{x}^{SEG}; \mathbf{x}^{PART}\right]$	trn/tst	69.5	62.2

Table 5.4: Using extra annotation on Caltech-UCSD Birds 2011. The top two rows show the results if the head detector is trained using human annotation rather than unsupervisedly trained, while the bottom rows show the accuracies if the head position is given even during test time. Overall, the performance of the simple system that uses extra annotation is better than the performance of the symbiotic system (c.f.Tab. 5.2).

As shown in Tab. 5.4, the resulting systems were able to surpass the performance of the symbiotic system even when using trained head detector. Using ground truth head localizations, the gap in the achieved accuracy compared to the symbiotic system (and, naturally, all other systems evaluated on this task) becomes very large. Overall, our conclusion here is that part localization has a great potential for fine-grained categorization. While the segmentation-based discovery and localization that we present in this chapter is a definite step forward, compared to fully unsupervised part discovery and localization, there is still a big room for improvement to unleash the full potential of part localization for base-class modeling.

5.4 FGVC challenge 2013

The FGVC challenge (fgc) was organized in conjunction with ICCV 2013 to demonstrate new improvements in FGVC. It contained subsets extracted from 5 prominent fine-grained dataset. The details of the data are listed in Tab. 5.5. Participants were tasked with solving each of the 5 FGVC problems separately, and mean accuracies are averaged to obtain the final score of a team. Optional bounding boxes for each im-

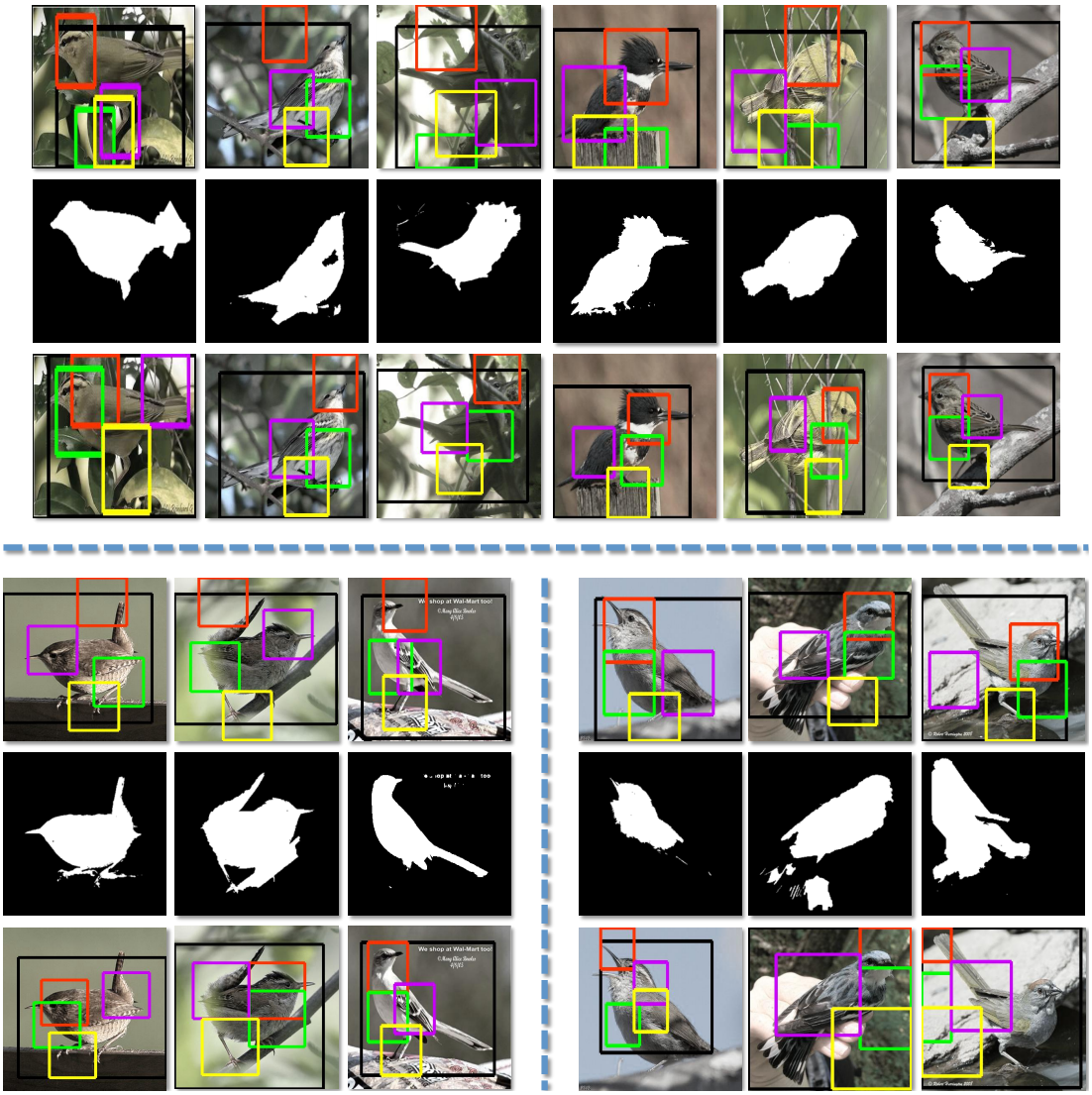


Figure 5.3: Examples taken from the Caltech Birds dataset. Every three rows: (top) part localizations using the symbiotically trained DPM, but fitted without the guidance of segmentation, (bottom) the same DPM model fitted with the help of segmentation (i.e. our full system). The segmentations are shown in the middle. In most columns, a noticeable improvement in the localization consistency is observed. The last three columns in the bottom three rows show some failure cases where segmentations hurts part localization.

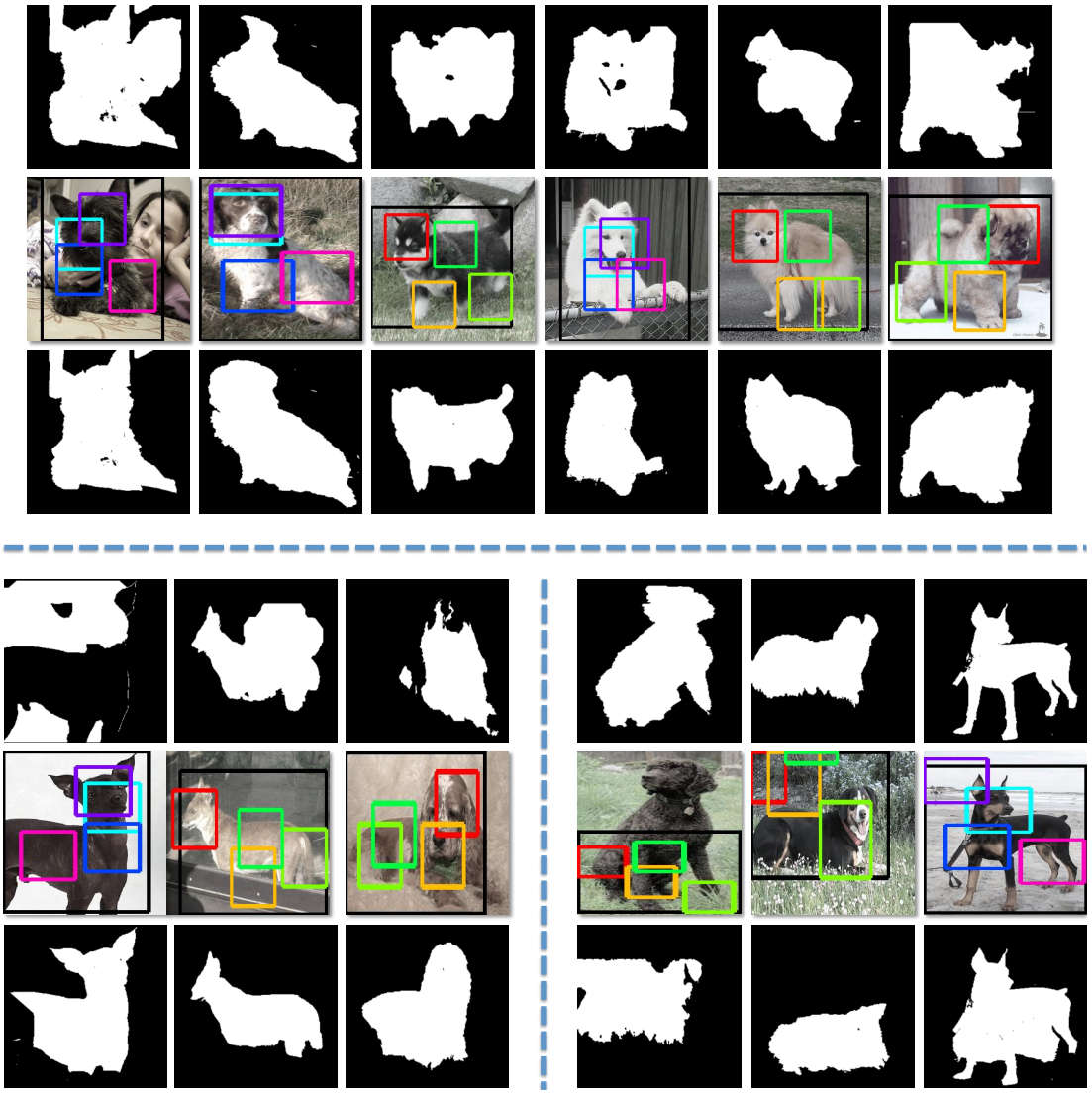


Figure 5.4: Examples taken from the Stanford Dogs dataset. Every three rows: (top) stand-alone segmentation results using GrabCut, (bottom) segmentation results with the help from the localized parts shown in the middle row (our full system). In most columns, a noticeable improvement in the segmentation accuracy is observed. The last three columns show sample failure cases.

Domain	# classes	# training images	# test images
Airplanes	100	6667	3333
Birds	83	4150	4105
Cars	196	8144	8041
Dogs	120	26500	12000
Shoes	70	3511	1002

Table 5.5: Overview of the data at FGVC challenge 2013. It contained 2 animal categories, birds and dogs, 2 vehicle categories, airplanes and cars. The shoe category turned out to be relatively small compared to the other ones and was solved quite well by almost all participants.

age is provided. And participants were able to choose between two tracks depending whether they made use of the bounding boxes during test time.

Results are listed in Tab. 5.6 (track 1, with bounding box usage) and Tab. 5.7 (track 2: without bounding boxes). Our entry, which achieved 4th in track 1 and 2nd in track 2, was a straightforward application of the symbiotic model on 4 out of 5 categories. Our entry for the shoes did not use the symbiotic model and used a simple GrabCut only approach, the same as entry 2 in Tab. 5.2.

The competition was won by a joint team of Xerox Europe and Inria. The method was similar to ours in that it also used Fisher vectors and SVM for classification. However, they used a much larger vocabulary (1024 - 4096 Gaussians compared to ours 256) and they made crucial modifications that turned out to have a great impact on the performance, e.g. RootSIFT by Arandjelović and Zisserman (2013) or filtering of low energy patches. A detailed report of the method can be found in Gosselin et al. (2014).

It is worth mentioning that the majority of team used CNN-based methods (Krizhevsky et al. (2012)), which required extra training data (usually ImageNet) to prevent overfitting.

Team	Aircraft	Birds	Cars	Dogs	Shoes	Average
Inria-Xerox	81.46	71.69	87.79	52.90	91.52	77.07
CafeNet*	78.85	73.01	79.58	57.53	90.12	75.82
Inria-Xerox	75.88	66.28	84.70	50.42	88.63	73.18
VisionMetric*	75.49	63.90	74.33	55.87	89.02	71.72
Ours	75.85	69.06	81.03	44.89	87.33	71.63
Inria-Xerox	80.59	58.54	84.67	35.62	90.92	70.07
CognitiveVision*	67.42	72.79	64.39	60.56	84.83	70.00
DPD Berkeley*	68.47	69.58	67.40	50.84	89.52	69.16
VisionMetric	73.93	51.35	69.31	38.63	87.33	64.11
CognitiveVision	58.81	51.69	52.37	47.37	78.14	57.68
MPG	9.45	54.57	69.27	42.92	88.42	52.93
MPG	9.45	56.47	63.77	0.97	88.42	43.82
Infor FG*	30.39	9.06	4.45	0.82	35.23	15.99
InterfAIce	5.79	2.56	1.12	6.96	5.99	4.48

Table 5.6: Official results in Track 1, with the use of bounding boxes during test time. * indicates the usage of external data during training. These results are not directly comparable to the ones without using external training data. Our entry ranked 4th overall and 2nd among participants not using any additional training data.

Team	Aircraft	Birds	Cars	Dogs	Shoes	Average
Inria-Xerox	80.74	49.82	82.71	45.71	88.12	69.42
Ours	72.49	46.02	77.99	37.14	89.12	64.55
Inria-Xerox	66.40	44.51	76.35	43.96	86.33	63.51
Inria-Xerox	80.74	34.45	76.89	24.40	87.33	60.76
DPD Berkeley*	45.51	42.70	43.38	41.91	59.98	46.70
Infor FG*	9.66	5.75	3.71	32.71	4.69	11.30
InterfAIce	5.43	2.58	1.17	6.94	5.29	4.28

Table 5.7: Official results in Track 2, without the use of bounding boxes during test time. Our entry 2nd, only behind Inria-Xerox' highly optimized Fisher vector-based method.

5.5 Conclusion

We have introduced and demonstrated the worth of a symbiotic part localization and segmentation model for fine-grained categorization. It successfully pulls together a number of recent research strands: the use of distinctive parts for registration when discriminating sub-ordinate categories (Bourdev et al. (2011), Parkhi et al. (2012), Yang et al. (2012), Yao et al. (2011), Zhang et al. (2012)); unsupervised discovery of mid-level discriminative patches (Pandey and Lazebnik (2011), Singh et al. (2012), Yang et al. (2012)); learning a DPM given only weak annotation (a loose bounding box compared to the tight boxes provided in PASCAL VOC) (Blaschko et al. (2010), Chum and Zisserman (2007), Nguyen et al. (2009)); and, improving segmentations using a lite spatial model (Winn and Jojic (2005)).

It also opens up new research questions: how can the model be extended from loose bounding box annotation to (even weaker) image level annotation? How should the number of components and parts be determined automatically? How should humans be used in-the-loop (Branson et al. (2010)) to provide annotation at test time (based on the results from section 5.3.2)?

Chapter 6

Developing a new flower dataset

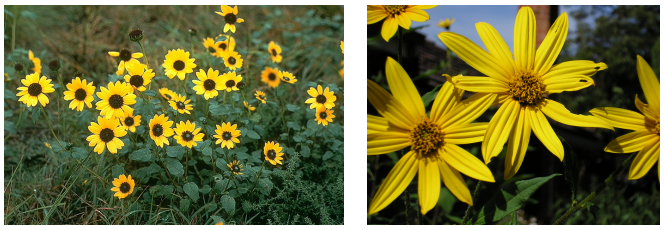
Larger and better datasets have served as enablers and catalysts for new methods. With the help of many botanists and plant enthusiasts, we collected a new dataset containing over 1000 classes of different flowers species. The size and number of classes of this dataset surpassed all public FGVC datasets to our best knowledge. In collecting this dataset, we also faced many challenges and this chapter contains insights on solving these.

6.1 The plant world

In this section, we share some basic knowledge about the plant science, its taxonomic hierarchy and nomenclature. In the biological world, all living entities are organized by a single hierarchy, the taxonomic tree. Looking from the bottom up, the leaves of the tree are the *species* and the parents of species are called *genus* (plural: *genera*). Each species has a unique binomial name which consists of a genus prefix and a species suffix. For example, the most commonly known type of sunflower is *Helianthus annuus*,



(a) Hel. annuus (b) Hel. debilis (c) Hel. occidentalis



(d) Hel. petiolaris (e) Hel. tuberosus

Figure 6.1: Helianthus (Hel.) or commonly known as sunflower is one of the larger genera in the flower world with 85 different species. This figure shows five common ones (Images from Wikipedia).

where *Helianthus* is the genus and actually covers over 85 different sunflower species. *annuus* is a species suffix and the two names together form an unique species name. Fig. 6.1 shows 5 out of 85 different Helianthus species.

The whole taxonomic tree is estimated to have 1.9 million leaf nodes. The root of the tree has three children, which are the *domains*. These include Archaea and Bacteria (both single cell organisms) and Eukarya, which covers all organisms with enclosed cells. The children of Eukarya are called *kingdoms*, which consists of Animalia, Fungi and Plantae, the animals, the fungi and the plants. There are more levels between the Plantae kingdom and the genus level, they are however irrelevant for the creation of our dataset.

Not all plants have flowers, that is why only a subset of the Plantae kingdom made to the list of the candidates for a flower dataset. This flowering subset is called the *Angiosperms*, which has an estimate of 350,000 species.

6.2 Motivations

There are various reasons for us to start with collecting a new dataset:

Larger datasets enable better algorithms. Since computers are becoming faster every day, more and more data can be used for training an intelligent system. In the recent history of object recognition, most big advances were only made possible via new datasets of unprecedented size and/or quality. For example, the deformable-parts-model (DPM) gained its fame via the PASCAL VOC datasets, or more recently, the convolutional neural networks (CNN) was first proven successful to large scale object recognition via the ImageNet challenges. The mostly cited FGVC datasets, as discussed in Sec. 2.8, have no more than 200 categories, a number which we like to increase and hopefully encourage the development of the next big advance in computer vision.

Benchmarks are getting saturated. Some of the benchmark FGVC datasets, including the Oxford Flowers 17 and 102, the UCSD-Caltech birds 2010 and 2011, have their performance slowly saturated. 90% accuracy (see Tab. 6.3) is achieved by a fine-tuned CNN on Oxford Flowers 17 and 102, and performance on the birds has increased from 6% to 85% (Branson et al. (2014)) over the course of 3 years. Visualization for both dataset has indicated that most errors were due to limited training data size or poor test image quality, e.g. cropped, blurred, significantly occluded. Another danger of keeping using older benchmark datasets is the indirect feedback from test set back to training, since looking at the test set performance is unavoidable with every related paper one reads.

A taxonomically correct dataset. Most FGVC datasets are based on common names and not taxonomic names, which means a category can be a single species, a genus

or something even higher in the taxonomic hierarchy. This leads to different degrees of intra-class variation within each category and forces computer vision researchers to solve problems that should not have existed in the first place. For instance, most categories on Oxford Flowers 102 have near 100% accuracy, while one or two perform very badly. These are the ones that are actually a genus and display more variation than competing categories. Moreover, a dataset with a hierarchy enables new research direction within the FGVC.

Better understanding of building an FGVC dataset. Building a dataset is a prerequisite for any new FGVC-based application. However, building a fine-grained dataset is somewhat more challenging than building a generic dataset, and there are plenty of design choices involved. With the collecting of a large FGVC dataset, we'd like to set a reference case for how certain decisions affects the performance of the eventual FGVC applications.

6.3 Challenges

Before going into the main challenges of collecting an FGVC dataset, let us first look at the common data acquisition pipeline, which consists of the following steps:

Category candidates list creation. First, it must be decided which objects should belong to the dataset. For example, the name list for the Caltech 101 dataset (Fei-Fei et al. (2004)) was sampled from all words with a drawing in Webster Collegiate Dictionary, and ImageNet was built using a word hierarchy called WordNet.

Data acquisition. The standard way to gather images for a new dataset is to scrape them from the internet. One usually uses the Google or Bing API, searches for the name of a category and retrieves the top hundred to thousand results. Google or Bing

search engines associate names with images via the description in the HTML page the image is linked to, and for well understood categories, such as the majority of classes in ImageNet, these associations are mostly correct.

Data curation. Images downloaded from the search engines do not have any guarantee to be correct and of good quality. Therefore, these images usually need to be checked manually by a human operator. In this process, the operator is presented with a candidate image and a label and is asked to make a binary decision whether the label corresponds to the given image. In the early days, graduate students were hired to performing this task, (Fei-Fei et al. (2004) or Everingham et al. (2006)), but it has been mostly replaced by Amazon Mechanical Turk (AMT) in the more recent years.

One property that defines an FGVC problem is the fact that non-experts do not know the correct label name given an input image. This property affects the common data acquisition pipeline in many ways.

When it comes down to fine-grained labels on the web, the chance of mislabeling increases dramatically compared to generic labels. In the plant world, mislabeling frequently occurs between species sharing a common genus. For instance, when the search keyword is *Helianthus gigantheus*, some of the top search results from Bing and Google actually belong to *Helianthus annuus*, a sister species of the former. Helianthus annuus has a gigantic flower centre and therefore gets easily mislabeled as *gigantheus*. In reality, *gigantheus* describes the actual size of the plant, not the size of the flower centre. As a result of mislabeling and missing of fine-grained labels in the description of candidate images, the yield of using search engines for building an FGVC dataset is very limited. Fig. 6.2 shows one example.

Obviously, the data curation stage is also greatly affected, since the operator lacks the necessary knowledge to verify or reject image-label-associations.

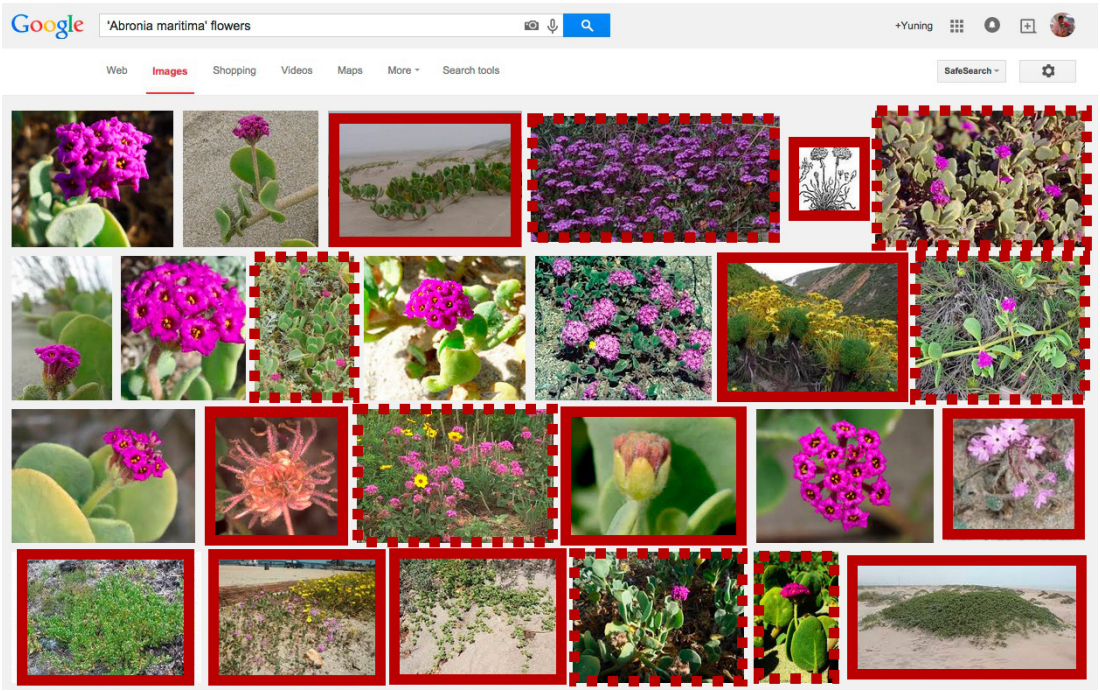


Figure 6.2: A Google search for the flower species *Abronia maritima*. The precision already drops significantly within the top 20 of the search results. A solid red border indicates an incorrect label or no flowers inside at all. A dotted red border indicates a correct label, but poor imaging quality (e.g. too small for being used for visual recognition). Rest are good images that would be accepted into the dataset.

6.4 Data sources

We collected 304,419 flowering plant species names from the Plantlist[1], which is a collaboration between various botanic institutions across the globe in order to create a unified database of plant names.

The images in the new dataset were collected via three sources: interest groups, commissioned, and extracted from existing datasets, each of which will be discussed in more detail below.

6.4.1 Interest groups

One of the best resources to gather a fine-grained dataset is via interest groups, which consists of hobbyists and enthusiasts of the domain (e.g. of flowers) the dataset is built around (Maji et al. (2013)). Therefore, interest groups were our first choice. To most benefit from these interest groups, we collaborated with the Royal Botanic Garden Kew, who had relationship to a number of interest groups on the popular photo sharing site *Flickr* [2].

Obviously, trying to collect a dataset for all 304k species at the same time seems unreasonable. We decided to start off with 1642 species that can be commonly found in the UK. Around 20k Flickr links belonging to these species were gathered.

Local flower enthusiasts volunteered to help double checking the acquired data from Flickr. Due to limited resource, each candidate image was only checked by a single volunteer. By the end, 15,539 images covering 562 species were collected via interest groups.

[1] www.theplantlist.org
[2] www.flickr.com

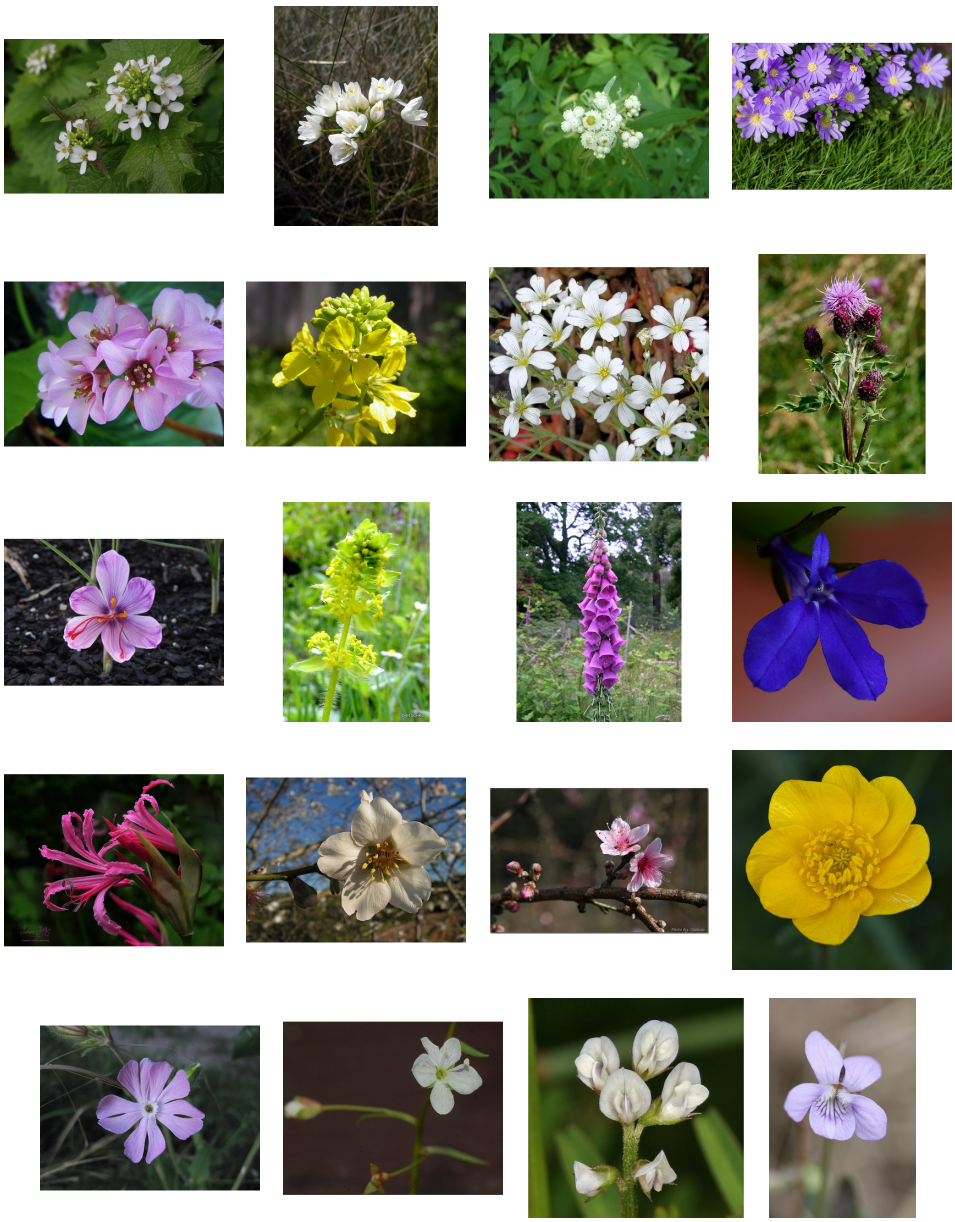


Figure 6.3: Images collected via interest groups on Flickr. These images are of high quality, usually with a big and in focus flower in the center of the image. They have been checked by local flower enthusiasts, who ensured their labels are correct.

Some examples of images gathered are shown in Fig. 6.3. Notice most photographs are of very high imaging quality, since they were taken using high-end cameras. The flowers in each image are well centered and in focus, while the background tend to be

blurred, which is ideal for many foreground segmentation methods. However, those high quality images pose a domain shift to the flower images taken by a smartphone application, which is one of the ultimate goal of FGVC. As depicted in Fig. 6.4, smartphone images tend to look very different than Flickr images, which is a result of different camera technologies and the fact that real life images differ from artistic images. Unfortunately, mining smartphone photographs is even more challenging, because these are usually private compared to Flickr images which are made for displaying and sharing. Therefore, our hope is to start with high quality, overly optimistic images and slowly correct this domain shift via users taking pictures on their phone and submitting them to our FGVC application.

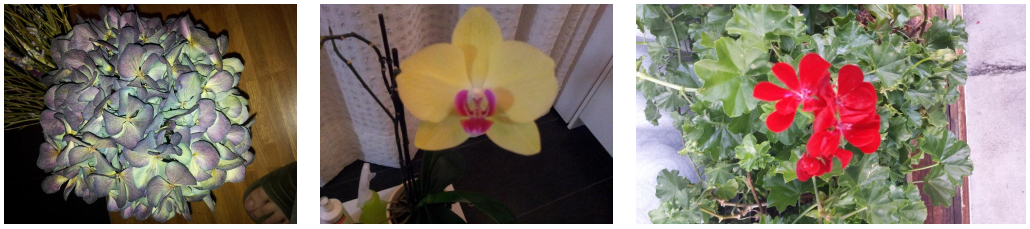


Figure 6.4: Three smartphone submissions to our demo flower classification app (oxflora.yuningchai.com). These images demonstrate a domain shift from the images acquired via both interest groups and commissioning. Left: the true color of the flowers cannot be determined due to camera lightning. Middle: the photograph was taken indoor whereas most images retrieve via interest groups are outdoor. Right: the flower is slightly blurred, while the background is perfectly in focus.

6.4.2 Commissioned

It is natural when collecting a dataset, no matter whether downloaded via Google search or via interest groups, that the distribution of number of images will be uneven, since some categories are rarer than others. While most people would just cut off the categories with fewer images, we applied a more courageous approach in dealing with the imbalance. After assembling an initial dataset with images from the interest

groups, we commissioned flower enthusiasts to go out and take pictures of flowers belonging to the species that rarely appeared in the initial dataset. The enthusiasts were paid 1 GBP per photograph and were asked to use their professional cameras and not smartphone cameras, since we valued higher quality images over correcting the domain shift as discussed in Sec. 6.4.1.

This task is made easier since the initial set only consisted of common UK species, where coordinates of appearance of each species are known. 11,259 images covering 445 species were gathered via commissioning.

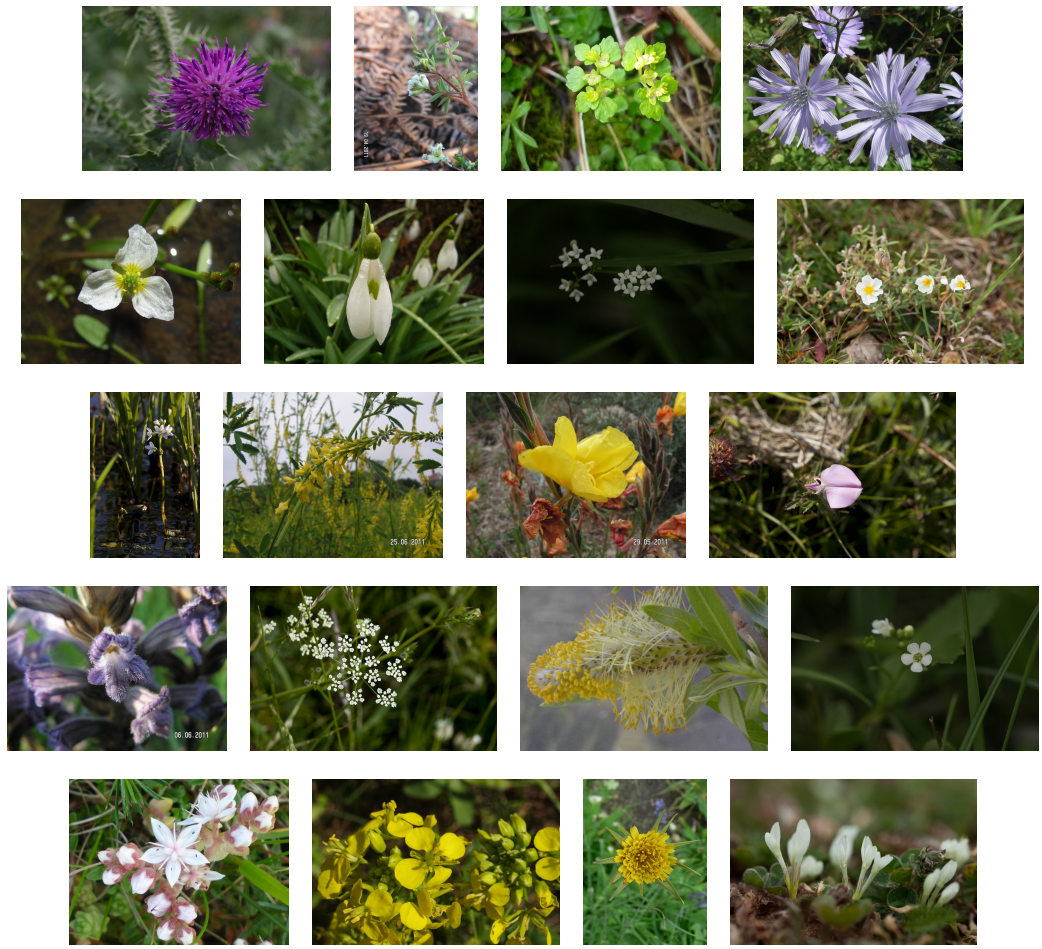


Figure 6.5: Images collected via commissioned flower enthusiasts. The image quality is similar to that of the Flickr images collected via interest groups. Commssioning people is however very expensive, and should only be used to fill categories that are poorly represented by other means.

As shown in Fig. 6.5, images taken by by commissioned enthusiasts show similar quality as the ones collected via interest groups. Theoretically, the images collected this way should represent the ideal of a computer vision scientist. However, upon visualizing the commissioned photographs, we discovered a few problems caused by miscommunication, which should be prevented in the future:

Multiple images displaying the same flower instance. Although it was mentioned to the photographers, it still happened in numerous occasions that one flower instance

appeared in multiple photographs. In the case of rare flowers, the photographers often opt to take multiple shots of the same flower rather than trying to find a second flower. Moreover, many photographers have the habit to take multiple images in quick succession, in order to pick out a best one later but forgets to do so. Although view changes represent one of the challenges in FGVC, being able to generalize over instances is also important. Therefore, images taken from multiple instances are more valuable than from a single instance. An example is shown in Fig. 6.6.



Figure 6.6: Illustration of multiple commissioned images sharing the same flower instances, which is not desired when building the flower dataset.

Not fully grown flowers. Images of flowers still in its early growing stage are not ideal for training a flower classifier, since many of its visual characteristics are not yet fully developed. The photographers should have been asked to return to the plant after a few days. Some examples of images containing not fully grown flowers are found in Fig. 6.7.

6.4.3 Mined from existing datasets (ImageNet)

Many existing datasets contain valuable images that would fit well into our flower dataset.

The Oxford Flower 102 dataset contains almost 8000 flower images across 102 categories. Unfortunately, many categories belong to a genus rather than a species and hence cannot be used. Most of the remaining categories are already well represented



Figure 6.7: Examples of flowers that are still in its early growing stage and not appropriate to be used to train a flower classifier.

via the two data sources mentioned earlier. Therefore, we decided to not include the Oxford Flower 102 dataset for now.

Several fine-grained datasets have been created by taking a subset from ImageNet, most notably the Stanford Dogs (Khosla et al. (2011)) and Stanford Cars (Krause et al. (2013)). Driven by the success of these datasets, we built the second part of the flower dataset from ImageNet.

ImageNet contains the synset *angiosperm, flowering plant* with 618 children. However, only 281 species could be unambiguously mapped to our flower species list. We then discarded categories with less than 50 images, which left 259 categories in total. 187,559 images covering these species were acquired from ImageNet.

6.5 Experiments with AMT

Although ImageNet has been filtered by Amazon Mechanical Turk (AMT), we noticed on many occasions that ImageNet images suffer from incorrect label or poor image quality, which raised the question: How good are ImageNet images for a fine-grained dataset? Is another filtering necessary? To answer these questions, we had ImageNet

flower images filtered by the AMT a second time, this time in a more fine-grained friendly environment.

6.5.1 Setup

As discussed in Sec. 6.3, the curation of FGVC labels is challenging since AMT operators won't be able to associate fine-grained species name to images. Inspired by the design of Welinder et al. (2010b), we developed a visual comparison-based annotation environment specifically for FGVC. The idea is simple: although operators cannot associate names with images, they are able to compare between images. As shown in Fig. 6.8, we presented the operator with five hand-picked images showing the desired species, and asked the operator whether the input image looks similar to them. The five reference images should span the possible appearance space of a target species as complete as possible. Obviously, mislabeling can still occur if two species can only be distinguished in small details, which will be missed by the operators. But we think it is still an improvement over label-image type of annotation and it is the best we can do within a reasonable cost limit. Along with checking for the correct label, we also asked the AMT operators to reject images which do not contain any flower or is of bad imaging quality, or the contained flowers are too small. We gave elaborative examples to explain those cases.

During the annotation, every 100 images belonging to the same category were grouped together and submitted as a single AMT task. A total of 577 tasks were submitted, containing 50,057 images across 259 categories. Each task was repeated for 5 times, yielding 250,285 binary annotations in total. A histogram of number of "accept" votes per image is shown in Tab. 6.1.



Figure 6.8: AMT interface designed for FGVC. Five sample images displaying a good variation of desired images are fixed on the top of the page. AMT operators are asked to select the candidate images that look similar to the five samples. Candidate images show blue border when selected. And each image is checked by 5 AMT operators.

# votes	0	1	2	3	4	5	total
occurrence	1,157	3,036	7,166	12,256	15,916	10,526	50,057
percentage	0.02	0.06	0.14	0.24	0.32	0.21	1

Table 6.1: A histogram of number of votes submitted by AMT operators over 50k image samples. Since every image is checked by 5 operators, the maximum of votes is 5 if all operators agree on the sample being a good image with the correct label. A 0 indicates all operators univocally rejecting an image. Overall, the table shows that ImageNet data is of good quality, with over 70% of images passing the majority voting criterion and only a mere 2% are rejected by all operators.

	# Trn. images	Accuracy
Majority	38,698	43
Union	48,900	45
No filtering	50,057	**46.6**
Welinder	33,279	42.5
Welinder with bias	42,670	44.0

Table 6.2: CNN-based FGVC experiment on the effect of various ways using the AMT annotation. Two trends can be observed here: 1) Not using the AMT and just taking all training images seems to perform the best. 2) The accuracy monotonically increases with the number of images used during training.

6.5.2 Playing with annotations

In a second AMT experiment, we would like to find out whether the acquired annotations can be used to improve the classification performance. We created subsets of the 50,057 images submitted to the AMT according to different ways to aggregate the AMT responses:

- **Majority voting**. An image is accepted if more than half of the operators vote for it.
- **Union**. An image is accepted if at least one operator votes in its favor.
- **No filtering**. All images are used for training, independent of the AMT results.
- **Welinder et al.**. An implementation of Welinder et al. (2010a). The work approximate the annotation process as a probabilistic model. See Fig. 6.9 for details.

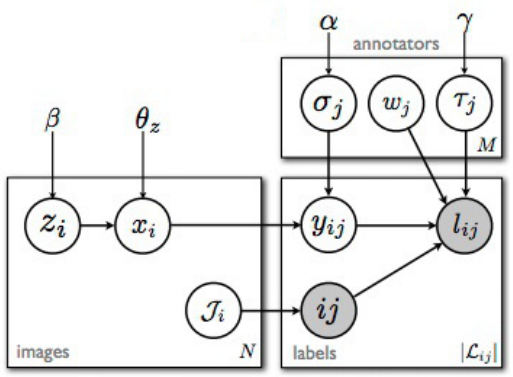


Figure 6.9: Probabilistic graphical model extracted from Welinder et al. (2010a). The index i denotes the i-th sample/image, while the index j indicates the j-th operator. The groundtruth label $z_i \in \{0, 1\}$ describes whether in reality an image should be accepted or rejected, which is the variable to estimate. z_i generates a vector-style feature representation x_i, which is in turn affected by an both operator-specific and image-specific noise, yielding y_{ij}. Finally, an operator-specific threshold τ_j and weighting of feature representation ω_j are applied and we observe l_{ij} as the output of the AMT.

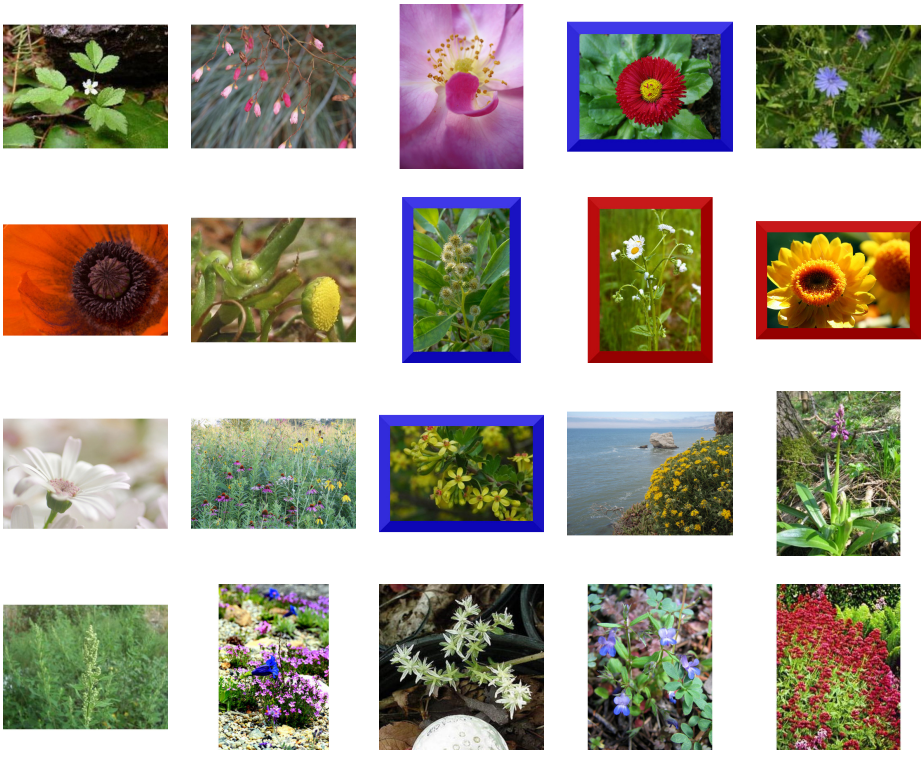


Figure 6.10: Randomly selected images that would be excluded from the dataset according to the majority decision of 5 AMT operators. AMT operators were asked to check for both image quality and label correctness. Most of the samples here are truthfully rejected in our opion, except for the two images with a red border. The three images with a blue border were correctly rejected because of wrong species label (ImageNet IDs: n11939699_1437, n11955896_1720, n12008749_2716). The rest are probably rejected because of poor image quality.

A visualization of the images rejected by the majority decision of 5 AMT responses is shown in Fig. 6.10. Only 2 out of 20 images would have been wrongly excluded from the dataset, the other 18 are truthfully rejected because they are either of poor quality or labeled to the wrong category. Qualitatively, the AMT responses cleans up the ImageNet.

Experimental setup. To numerically evaluate the AMT responses, we designed an experiment with the help of a state-of-the-art CNN by Simonyan and Zisserman (2014), which showed great improvement over the original CNN by Krizhevsky et al. (2012) during the ImageNet challenge 2014. We constructed 5 training sets according to the different ways presented at the start of this section. We then used each of them to fine-tune an ImageNet-CNN for fine-grained flower classification. Images were resized so that the shorter dimension is equal to 256 pixels. Random windows of 227×227 are then sampled as input of the network during training. We created an evaluation set from ImageNet that does not overlap with any sample submitted to the AMT. This subset was manually filtered by students for quality and label correctness, and contains 10,360 images. During test time, the center 227×227 pixels is cropped and used as the only input for classifying an image. A comparison of the 5 configurations are shown in Tab. 6.2.

It seems the more images used during training the better is the performance. It is especially surprising in the comparison between "Union" and "No filtering", where the only images discarded by the former were images where all 5 AMT operators univocally rejected. Our intuition is that although these images are very noisy with respect to the fine-grained label, they still helped training the lower layers of the network and served well as negative samples for the negative classes.

In conclusion, ImageNet data proved to be very good to build a new fine-grained dataset. If one is only interested in building a classification system with highest accuracy, no further filtering is needed. However, if the interest lies in building a dataset as clean as possible, another round of fine-grained style annotation would still improve the quality of the dataset.

6.6 Statistics and benchmark

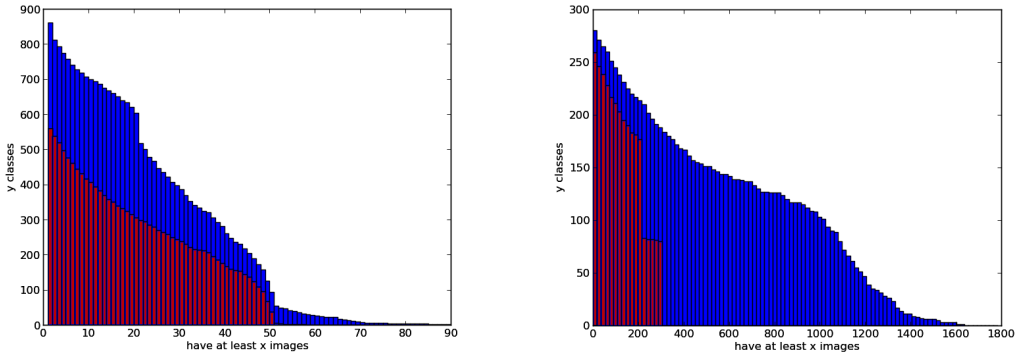
In conclusion, the new Oxford Flower dataset consists of:

- 15,539 images covering 562 categories retrieved. Acquired via flower interest groups on Flickr.
- 11,259 images filling the less represented of the 842 species. Acquired via commissioning flower enthusiasts to take new photographs in the wild.
- 187,559 images extracted from ImageNet, covering 259 species. 50,057 from them undergone another AMT filtering, with each of them having 5 binary labels from 5 different AMT operators indicating the correctness of the sample.

The number of images per class distributions are shown in Fig. 6.11.

From the data we collected, 4 subsets were made for comparison purposes for future work:

Oxford Flowers 261 (OF261). From the 26,798 images collected via Flickr interest groups and commissioning, we selected 261 species each of which has 40 or more images. 30 images per class are randomly selected for training, and the rest are used for evaluation. Image samples belonging to this dataset are shown in Appx. A.



(a) Interest group and commissioned (b) From ImageNet

Figure 6.11: Histograms showing how many classes (x axis) have at least how many images (y axis). Note the scales are different for both graphs. a) Red shows the histogram of images taken from Flickr interest groups, and blue shows those plus photographs we commissioned. It is clear to see that the commissioned photographers tried to fill the gap so every species has at least 50 images. Additional species, which did not appear from Flickr interest groups but are common in the UK, are included as well. b) Blue shows all flower images extracted from ImageNet, and red shows the subset of data which we collected AMT annotation for.

Oxford Flowers ImageNet 234 (OFI234). From the 50'057 ImageNet flower images we acquired AMT annotations for, 30 training and 30 test images are randomly selected for each of the 234 present categories. In the *Imbalanced* variation, the same 30 test images are used for testing, while *all* other images from the 50'057 are used for training. Image samples belonging to this dataset are shown in Appx. B.

Oxford Flowers 471 (OF471). Here, the training parts of the two previous datasets are merged into one single larger training set, while the test sets are merged into a new test set. There are 24 categories shared between the two sources. In those cases, the joined training set consists of 30 images randomly sampled from the training sets of both sources to maintain a healthy class-balanced dataset.

Using the same CNN architecture and training procedures as discussed in Sec. 6.5.2, we computed baselines for the 4 datasets, which are presented in Tab. 6.3. It was

Name	# Trn. Images	# Tst. Images	Baseline accuracy (%)
OF102	2040	6149	90.0
OF261	7830	5176	83.5
OFI234	7020	6911	37.4
OFI234 Imbalanced	34,454	6911	47.2
OF471	14,130	12,087	54.7

Table 6.3: Overview of the new flower datasets. OF261 consists of images only from Flickr interest groups and commissioning. Both OFI234 are from the flower subset of ImageNet. And OF471 is the union of OF261 and OFI234. As a reference, OF102 is the old Oxford Flower 102 dataset by Nilsback and Zisserman (2008).

surprising that the performance on OFI234 set is significantly lower than on OF261, while the former has even fewer categories to classify. This problem is traced back to the limited imaging quality and label correctness in ImageNet compared to images gathered by other means. OFI234 Imbalanced shows better performance because it has a much larger training set.

Chapter 7

Conclusion

In this chapter, we summarize the contributions in this thesis and discuss future research directions.

7.1 Achievements

We proposed several methods to improve performance on FGVC. Most parts of the work focused on improving foreground segmentation and introducing co-segmentation to FGVC. Towards the end, we included parts alignment into our segmentation framework. Apart from technical novelties, each of the works outperformed previous works by a good margin at the time of publication. They also explored new developments in the context of FGVC, e.g. the usage of the Fisher vectors by Perronnin et al. (2010). Here we summarize the main achievements:

Co-segmentation for image classification. We presented BiCoS, a co-segmentation method that is arguably very simple, yet is scalable and performs remarkably well in our experimental comparisons. BiCoS was significantly faster than most other co-

segmentation method, having a linear complexity with respect to the number of pixels (superpixels) compared to quadratic. BiCoS outperformed GrabCut that treats images independently and the state-of-the-art co-segmentation (e.g. Joulin et al. (2010)) in all comparisons. And an image classification system trained on datasets co-segmented with BiCoS improved over the state-of-the-art by 5% on Oxford Flowers 102 and 10% on UCSD-Caltech Birds at the time of publication. These numbers were only topped by our own TriCoS work a year later. Despite being mainly developed for FGVC, BiCoS also inspired other works in the segmentation community (Kim et al. (2012), Kuettel et al. (2012)).

Class-discriminative co-segmentation. TriCoS was one of the first to directly incorporate multi-way classification terms into a segmentation objective. By exploring two new developments in the community, interest region proposals and Fisher vectors, TriCoS achieved new state-of-the-art for the UCSD-Caltech Birds and the Stanford Dogs by improving the former by 6% and the latter by 12% in accuracy at the time of publication.

Unify foreground segmentation and parts localization In Symbiotic, we formulated the two most important aspects of FGVC research, foreground segmentation and parts localization into a unified convex optimization framework, which could be easily solved via block coordinate descent. At the time of publication, the method more than doubled the state-of-the-art accuracy on UCSD-Caltech Birds from 28.2% to 59.5%, and also improved the performance on Stanford Dogs by 6%. Moreover, it showed that combining two tools in a principled and convex way significantly outperforms more heuristic approaches. Symbiotic achieved the 2nd place in the Fine-grained Challenge 2013 in the category *No Boundingbox*. It also fared well in the *With Boundingbox* category, achieving the 4th place overall, and the 2nd place among all candidates who

did not use additional data.

Creation of a new benchmark dataset. We collected a new flower dataset consisting of 15,539 images covering 562 species from Flickr interest groups, 11,259 images covering 445 species from commissioning flower enthusiasts, and 187,559 images covering 259 species extracted from ImageNet. Alongside the images, we conducted experiments using the Amazon Mechanical Turk (AMT) service to filter ImageNet flowers, and concluded that although additional filtering using the AMT qualitatively purifies the data, but the sheer size of the data plays a more important role than its correctness when reflected in fine-grained classification accuracy. We also proposed subsets to be used as benchmark and provided baselines for these.

Development of web/phone flower recognition applications. We developed several demo system to show off our expertise in flowers recognition. The newest version [1] runs on a subset from the new flower dataset mentioned above, and can classify between 271 with 93% top 10 accuracy. The processing time is kept under 2 seconds. Printscreens of the demo is shown in Fig. 1.1.

7.2 Future work

7.2.1 Models and training

Transferring knowledge to CNN. CNN has been proven to be very successful in many applications in computer vision. Therefore, we believe that one of the priorities in FGVC should be to implement existing techniques in deep learning. At the moment, most works still use older techniques such as the DPM (Felzenszwalb et al. (2010)) or

[1] oxflora.yuningchai.com

the Poselets (Bourdev and Malik (2009)) for parts alignment and then apply a CNN on top of the activations. Zhang et al. (2014) replaced parts alignment with a RCNN-based (Girshick et al. (2013)) framework and observed a clear improvement. However, pixel-wise foreground segmentation and unsupervised parts learning, which are also shown to be important to FGVC (Chap. 5) have yet to be implemented in deep learning. We believe this would increase the performance as significantly as in Zhang et al. (2014).

Unified deep learning framework. Conventionally, foreground segmentation or parts alignment had to be done separately to the final multi-class classification, since features used in the former differs from the ones used in the latter. However, it is only a matter of time when both steps are replaced by deep learning. We believe that it should be worth investigating whether merging both CNNs into a joint optimization objective would be beneficial.

Benefitting from the hierarchy. Our flower dataset as discussed in Chap. 6 is one of the first to have a taxonomically correct hierarchy. Various works have shown improved classification accuracy (most recently by Deng et al. (2014) on ImageNet) using a category hierarchy. An interesting direction would be to investigate hierarchies in a fine-grained scenario, e.g. on the new flower dataset. Another possible direction of using the hierarchy is how deep the algorithm should go along the hierarchy to provide the user with the most confident and useful information, as initialized by Ordonez et al. (2013).

Multiple query images. Although FGVC has made huge progress in the recent years, its performance is still not good enough for commercial applications. One way to solve confusions between closely looking species is to perform the categorization on multiple query images belonging to the target species. Classification of a set of images in the fine-grained context has not been investigated yet, to our best knowledge.

7.2.2 Data is still the king

New ways for data acquisition. As shown in Chap. 6, collecting a fine-grained dataset is very challenging. Extracting images from the web and using the AMT is very scalable and provides a huge amount of data, but the image quality is not guaranteed. Collecting and curating data via contacts and interest groups provide good quality, but is not scalable at all. We believe the solution lies somewhere in between the two cases. One idea is to start large and low precision and slowly tighten the threshold for image quality using annotators with increasing expertise.

Better annotation tools. Images are not the only means to enrich the training set. Another source of data is the annotation on existing images, e.g. parts locations, foreground bounding boxes, attributes. However, obtaining those extra annotations can be expensive. We believe the cheaper way that provides more acurate annotations is to aid the annotators using computer vision systems, as done by Branson et al. (2010).

Appendices

Appendix A

Oxford Flowers 261


Aceras anthropophorum


Achillea ptarmica


Adoxa moschatellina


Aethusa cynapium


Agrimonia procera


Alcea rosea


Alliaria petiolata


Allium roseum


Allium triquetrum


Anacamptis pyramidalis


Anagallis tenella


Anchusa officinalis


Anemone ranunculoides


Angelica sylvestris


Anthyllis vulneraria


Aquilegia vulgaris


Arenaria serpyllifolia


Arum maculatum


Achillea millefolium


Aconitum napellus


Aegopodium podagraria


Agrimonia eupatoria


Ajuga reptans


Alisma plantago-aquatica


Allium giganteum


Allium schoenoprasum


Allium ursinum


Anagallis arvensis


Anchusa arvensis


Anemone blanda


Angelica archangelica


Anthriscus sylvestris


Antirrhinum majus


Arctium minus


Artemisia vulgaris


Asperula cynanchica


Aster tripolium


Astrantia major


Barbarea verna


Bellis perennis


Brassica napus


Cakile maritima


Calluna vulgaris


Calystegia soldanella


Campanula rotundifolia


Cardamine flexuosa


Cardamine pratensis


Carlina vulgaris


Centaurea jacea


Centaurea nigra


Centaurium erythraea


Cephalanthera damasonium


Ceratocapnos claviculata


Chelidonium majus


Astragalus glycyphyllos


Ballota nigra


Barbarea vulgaris


Blackstonia perfoliata


Buddleja davidii


Calendula officinalis


Caltha palustris


Campanula patula


Cardamine amara


Cardamine hirsuta


Carduus nutans


Centaurea cyanus


Centaurea montana


Centaurea scabiosa


Centranthus ruber


Cephalanthera rubra


Chamerion angustifolium


Cichorium intybus


Cirsium arvense


Cirsium heterophyllum


Claytonia perfoliata


Clinopodium vulgare


Convallaria majalis


Cornus mas


Cosmos bipinnatus


Cruciata laevipes


Cytisus scoparius


Dicentra spectabilis


Diplotaxis muralis


Epilobium hirsutum


Erigeron acer


Erophila glabrescens


Erythronium dens-canis


Filago vulgaris


Fritillaria meleagris


Galium palustre


Cirsium dissectum


Cirsium vulgare


Clematis vitalba


Colchicum autumnale


Convolvulus arvensis


Corylus avellana


Crataegus monogyna


Cymbalaria muralis


Daucus carota


Digitalis purpurea


Dracunculus vulgaris


Epipactis helleborine


Erodium cicutarium


Erophila verna


Euphorbia peplus


Filipendula ulmaria


Galium odoratum


Galium saxatile


Gentianella anglica


Geranium pratense


Geranium sanguineum


Geum urbanum


Glaux maritima


Gymnadenia conopsea


Helleborus orientalis


Hesperis matronalis


Hydrocharis morsus-ranae


Iberis amara


Iris pseudacorus


Iris versicolor


Kniphofia uvaria


Lamium amplexicaule


Lapsana communis


Lathyrus pratensis


Lepidium heterophyllum


Leucanthemum vulgare


Geranium dissectum


Geranium robertianum


Geranium versicolor


Glaucium flavum


Glechoma hederacea


Hedera helix


Hemerocallis fulva


Hyacinthoides non-scripta


Hypericum pulchrum


Impatiens glandulifera


Iris sibirica


Jasminum nudiflorum


Lamium album


Lamium purpureum


Lathyrus palustris


Lemna gibba


Lepidium ruderale


Linaria purpurea


Linaria vulgaris


Listera ovata


Lotus corniculatus


Lunaria annua


Malus domestica


Matricaria discoidea


Melampyrum arvense


Melilotus officinalis


Mentha arvensis


Mimulus guttatus


Neottia nidus-avis


Oenanthe aquatica


Oenothera biennis


Origanum vulgare


Oxalis corniculata


Papaver rhoeas


Pedicularis palustris


Pentaglottis sempervirens


Linum catharticum


Lonicera periclymenum


Lotus pedunculatus


Lythrum salicaria


Malva moschata


Meconopsis cambrica


Melampyrum pratense


Mentha aquatica


Menyanthes trifoliata


Myosurus minimus


Nymphaea alba


Oenanthe lachenalii


Ononis repens


Orobanche hederae


Papaver dubium


Papaver somniferum


Pedicularis sylvatica


Phlox paniculata


Picris echioides


Pilosella officinarum


Plantago coronopus


Platanthera chlorantha


Potentilla erecta


Primula veris


Prunus cerasifera


Prunus spinosa


Pulmonaria longifolia


Ranunculus ficaria


Ranunculus lingua


Ranunculus sceleratus


Rhododendron ponticum


Rubus fruticosus


Sambucus nigra


Saxifraga granulata


Scrophularia auriculata


Sedum album


Pilosella aurantiaca


Pimpinella saxifraga


Plantago lanceolata


Potentilla anserina


Potentilla sterilis


Primula vulgaris


Prunus dulcis


Pseudofumaria lutea


Ranunculus acris


Ranunculus flammula


Ranunculus repens


Reseda lutea


Rosa canina


Rudbeckia fulgida


Saponaria officinalis


Scleranthus annuus


Sedum acre


Sedum telephium


Senecio aquaticus


Senecio jacobaea


Silene dioica


Sisymbrium orientale


Stachys arvensis


Stellaria graminea


Thesium humifusum


Torilis nodosa


Trifolium campestre


Trifolium fragiferum


Trifolium hybridum


Trifolium repens


Tripleurospermum inodorum


Ulex europaeus


Valeriana officinalis


Verbascum thapsus


Veronica arvensis


Veronica chamaedrys


Senecio erucifolius


Senecio squalidus


Sinapis arvensis


Sorbus aucuparia


Stachys officinalis


Stellaria media


Thymus polytrichus


Tradescantia virginiana


Trifolium dubium


Trifolium glomeratum


Trifolium medium


Trifolium subterraneum


Tussilago farfara


Valeriana dioica


Valerianella carinata


Verbena officinalis


Veronica beccabunga


Veronica filiformis


Veronica hederifolia


Veronica serpyllifolia


Vicia cracca


Viola odorata


Viola tricolor


Veronica montana


Viburnum opulus


Viola arvensis


Viola palustris

Appendix B

Oxford Flowers ImageNet 234


Abronia latifolia


Abronia villosa


Ageratum houstonianum


Amberboa moschata


Anemone cylindrica


Anemone quinquefolia


Anemone tetonensis


Anemonella thalictroides


Anthemis cotula


Aquilegia caerulea


Aquilegia vulgaris


Arethusa bulbosa

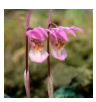

Arnica montana


Begonia rex


Bletilla striata


Buphthalmum salicifolium


Calendula officinalis


Calypso bulbosa


Abronia umbellata


Adonis annua


Allionia incarnata


Anaphalis margaritacea


Anemone nemorosa


Anemone sylvestris


Anemone virginiana


Anthemis arvensis


Antirrhinum majus


Aquilegia canadensis


Arenaria serpyllifolia


Arnica cordifolia


Aster dumosus


Bellis perennis


Brassia verrucosa


Calandrinia ciliata


Callistephus chinensis


Castilleja miniata


Catananche caerulea


Cephalanthera rubra


Chrysanthemum morifolium


Claytonia virginica


Collinsia verna


Corallorhiza maculata


Corallorhiza trifida


Cyclamen hederifolium


Cypripedium acaule


Cypripedium californicum


Cypripedium montanum


Dactylorhiza fuchsii


Daucus carota


Dianthus caryophyllus


Dianthus deltoides


Encelia farinosa


Epipactis gigantea


Erigeron annuus


Centranthus ruber


Chelidonium majus


Claytonia caroliniana


Collinsia parviflora


Conoclinium coelestinum


Corallorhiza striata


Cotula coronopifolia


Cyclamen purpurascens


Cypripedium arietinum


Cypripedium fasciculatum


Cypripedium reginae


Dahlia pinnata


Dianthus barbatus


Dianthus chinensis


Dianthus plumarius


Encyclia tampensis


Epipactis helleborine


Erigeron aurantiacus


Erigeron divergens


Erigeron speciosus


Eriophyllum wallacei


Erysimum cheiri


Felicia amelloides


Gentiana acaulis


Gentiana calycosa


Gentiana lutea


Gentiana saponaria


Gentianella quinquefolia


Gentianopsis holopetala


Glaucium flavum


Gymnadenia conopsea


Gypsophila paniculata


Helianthus annuus


Helianthus laetiflorus


Helianthus tuberosus


Hesperis matronalis


Erigeron philadelphicus


Eriophyllum lanatum


Erysimum asperum


Eustoma grandiflorum


Gazania rigens


Gentiana andrewsii


Gentiana clausa


Gentiana pneumonanthe


Gentiana villosa


Gentianopsis crinita


Gerbera jamesonii


Gomphrena globosa


Gymnadenia odoratissima


Helianthus angustifolius


Helianthus giganteus


Helianthus petiolaris


Helichrysum bracteatum


Heterotheca villosa


Hexalectris spicata


Himantoglossum hircinum


Lactuca serriola


Leucanthemum superbum


Lewisia cotyledon


Liatris punctata


Linaria canadensis


Liparis loeselii


Lithophragma parviflorum


Lonas annua


Malpighia glabra


Meconopsis betonicifolia


Mentzelia lindleyi


Mirabilis jalapa


Mirabilis multiflora


Moehringia lateriflora


Nigella damascena


Nigella sativa


Hexalectris warnockii


Hulsea algida


Layia platyglossa


Leucanthemum vulgare


Lewisia rediviva


Liatris pycnostachya


Linaria vulgaris


Lithophragma affine


Lobularia maritima


Malcolmia maritima


Matthiola incana


Mentzelia laevicaulis


Mesembryanthemum crystallinum


Mirabilis longiflora


Mirabilis oblongifolia


Monotropa uniflora


Nigella hispanica


Nyctaginia capitata


Ophrys apifera


Ophrys sphegodes


Papaver alpinum


Papaver californicum


Papaver nudicaule


Papaver rhoeas


Penstemon barbatus


Penstemon fruticosus


Penstemon palmeri


Penstemon serrulatus


Pericallis hybrida


Petunia hybrida


Phacelia campanularia


Phacelia tanacetifolia


Platanthera bifolia


Platystemon californicus


Portulaca grandiflora


Psychopsis papilio


Ophrys insectifera


Orchis mascula


Papaver argemone


Papaver heterophyllum


Papaver orientale


Papaver somniferum


Penstemon centranthifolius


Penstemon linarioides


Penstemon rupicola


Pericallis cruenta


Petunia axillaris


Petunia integrifolia


Phacelia minor


Phalaenopsis amabilis


Platanthera chlorantha


Polianthes tuberosa


Psychopsis krameriana


Ratibida columnifera


Ratibida tagetes


Rudbeckia laciniata


Saponaria officinalis


Scabiosa atropurpurea


Shortia galacifolia


Silene caroliniana


Silene latifolia


Silphium laciniatum


Solidago missouriensis


Solidago nemoralis


Solidago rugosa


Solidago spathulata


Spiranthes cernua


Spiranthes romanzoffiana


Stokesia laevis


Tagetes erecta


Tanacetum coccineum


Tanacetum vulgare


Rudbeckia hirta


Saintpaulia ionantha


Sarcochilus falcatus


Senecio bigelovii


Silene acaulis


Silene dioica


Silene virginica


Solidago canadensis


Solidago multiradiata


Solidago odora


Solidago sempervirens


Sparaxis tricolor


Spiranthes porrifolia


Spiranthes spiralis


Stylophorum diphyllum


Talinum paniculatum


Tanacetum ptarmiciflorum


Tellima grandiflora


Tragopogon dubius


Uvularia grandiflora


Valeriana officinalis


Verbesina alternifolia


Verbesina virginica


Veronica arvensis


Veronica officinalis


Wyethia amplexicaulis


Zantedeschia aethiopica


Tragopogon pratensis


Vaccaria hispanica


Vanilla planifolia


Verbesina encelioides


Veronica agrestis


Veronica chamaedrys


Veronica serpyllifolia


Xeranthemum annuum


Zinnia acerosa

Bibliography

Fgcomp 2013. https://sites.google.com/site/fgcomp2013. 92

Bogdan Alexe, Thomas Deselaers, and Vittorio Ferrari. Classcut for unsupervised class segmentation. In *ECCV*, 2010a. 39, 43, 49, 56, 57

Bogdan Alexe, Thomas Deselaers, and Vittorio Ferrari. What is an object? In *CVPR*, 2010b. 62

R. Arandjelović and A. Zisserman. Three things everyone should know to improve object retrieval. In *Proc. CVPR*, 2012. 23

R. Arandjelović and A. Zisserman. All about VLAD. In *Proc. CVPR*, 2013. 95

Himanshu Arora, Nicolas Loeff, David A. Forsyth, and Narendra Ahuja. Unsupervised segmentation of objects using efficient learning. In *CVPR*, 2007. 39, 57

A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky. Neural codes for image retrieval. In *Proc. ECCV*, 2014. 25

Dhruv Batra, Adarsh Kowdle, Devi Parikh, Jiebo Luo, and Tsuhan Chen. icoseg: Interactive co-segmentation with intelligent scribble guidance. In *CVPR*, 2010. 38, 49, 52

H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded up robust features. In *Proc. ECCV*, May 2006. 21

Thomas Berg, Jiongxin Liu, Seung Woo Lee, Michelle L. Alexander, David W. Jacobs, and Peter N. Belhumeur. Birdsnap: Large-scale fine-grained visual categorization of birds. In *Proc. CVPR*, 2014. 30

Andrew Blake, Carsten Rother, M. Brown, Patrick Pérez, and Philip H. S. Torr. Interactive image segmentation using an adaptive GMMRF model. In *ECCV*, 2004. 40

M. B. Blaschko, A. Vedaldi, and A. Zisserman. Simultaneous object detection and ranking with weak supervision. In *NIPS*, 2010. 97

Liefeng Bo, Xiaofeng Ren, and Dieter Fox. Kernel descriptors for visual recognition. In *NIPS*, 2010. 21

E. Borenstein and S. Ullman. Class-specific, top-down segmentation. In *ECCV*, 2002. 39, 56, 83

Lubomir D. Bourdev and Jitendra Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *ICCV*, 2009. 16, 17, 83, 122

Lubomir D. Bourdev, Subhransu Maji, and Jitendra Malik. Describing people: A poselet-based approach to attribute classification. In *ICCV*, 2011. 74, 83, 97

Yuri Boykov and Marie-Pierre Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. In *ICCV*, 2001. 19, 40, 43

Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *PAMI*, 2004. 81, 83

S. Branson, G. Van Horn, S. Belongie, and P. Perona. Bird species categorization using pose normalized deep convolutional nets. 2014. 26, 28, 34, 100

Steve Branson, Catherine Wah, Florian Schroff, Boris Babenko, Peter Welinder, Pietro Perona, and Serge Belongie. Visual recognition with humans in the loop. In *ECCV*, 2010. 27, 28, 29, 76, 91, 97, 123

Thomas Brox, Lubomir D. Bourdev, Subhransu Maji, and Jitendra Malik. Object segmentation by alignment of poselet activations to image contours. In *CVPR*, 2011. 83

L. L. Cao and L. Fei-Fei. Spatially coherent latent topic model for concurrent segmentation and classification of objects and scenes. In *ICCV*, 2007. 43, 57

João Carreira and Cristian Sminchisescu. Constrained parametric min-cuts for automatic object segmentation. In *CVPR*, 2010. 61

Rich Caruana. Multitask learning. *Machine Learning*, 28(1), 1997. 47

K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman. The devil is in the details: an evaluation of recent feature encoding methods. In *BMVC*, 2011. 21, 85

K. Chatfield, K. Simonyan, and A. Zisserman. Efficient on-the-fly category retrieval using convnets and gpus. In *Proc. ACCV*, 2014. 25

O. Chum and A. Zisserman. An exemplar model for learning object classes. In *CVPR*, 2007. 97

Gabriella Csurka, Christopher R. Dance, Lixin Fan, Jutta Willamowski, and CÃl'dric Bray. Visual categorization with bags of keypoints. In *ECCV*, 2004. 11

Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR*, pages 886–893, 2005. 11, 15, 21

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proc. CVPR*, 2009. 34

Jia Deng, Nan Ding, Yangqing Jia, Andrea Frome, Kevin Murphy, Samy Bengio, Yuan Li, Hartmut Neven, and Hartwig Adam. Large-scale object classification using label relation graphs. In *Proc. ECCV*. 2014. 122

Ian Endres and Derek Hoiem. Category independent object proposals. In *ECCV*, 2010. 61

M. Everingham, L. Van Gool, C. K. I. Williams, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2006 (VOC2006), 2006. URL `http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2006/`. 102

Theodoros Evgeniou and Massimiliano Pontil. Regularized multi–task learning. In *KDD*, 2004. 47, 48

Ryan Farrell, Om Oza, Ning Zhang, Vlad I. Morariu, Trevor Darrell, and Larry S. Davis. Birdlets: Subordinate categorization using volumetric primitives and pose-normalized appearance. In *ICCV*, 2011. 17, 18, 83

L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. *CVPR 2004, Workshop on Generative-Model Based Vision*, 2004. 101, 102

L. Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 28(4), 2006. 39, 56

P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *PAMI*, 2010. 15, 20, 77, 79, 82, 89, 121

Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient graph-based image segmentation. *IJCV*, 59(2), 2004. 41, 43, 61, 65

Brian Fulkerson, Andrea Vedaldi, and Stefano Soatto. Class segmentation and object localization with superpixel neighborhoods. In *ICCV*, 2009. 49

Peter V. Gehler and Sebastian Nowozin. On feature combination for multiclass object classification. In *ICCV*, pages 221–228, 2009. 54

Felix Gers. Long short-term memory in recurrent neural networks, 2001. 23

R. B. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *arXiv preprint arXiv:1311.2524*, 2013. 25, 26, 122

Philippe-Henri Gosselin, Naila Murray, Herve Jegou, and Florent Perronnin. Revisiting the fisher vector for fine-grained classification. *Pattern Recognition Letters*, 2014. 95

Stephen Gould, Richard Fulton, and Daphne Koller. Decomposing a scene into geometric and semantically consistent regions. In *ICCV*, 2009. 49

G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology, 2007. URL `http://authors.library.caltech.edu/7694`. 10

Dorit S. Hochbaum and Vikas Singh. An efficient algorithm for co-segmentation. In *ICCV*, 2009. 36, 41

Derek Hoiem, Alexei A. Efros, and Martial Hebert. Geometric context from a single image. In *ICCV*, 2005. 49

Satoshi Ito and Susumu Kubota. Object classification using heterogeneous co-occurrence features. In *ECCV*, 2010. 54, 55

T. Jaakkola and D. Haussler. Exploiting generative models in discriminative classifiers. In *NIPS*, pages 487–493, 1998. 22

Armand Joulin, Francis R. Bach, and Jean Ponce. Discriminative clustering for image co-segmentation. In *CVPR*, 2010. 36, 38, 39, 49, 52, 53, 54, 57, 58, 120

Christopher Kanan and Garrison W. Cottrell. Robust classification of objects, faces, and flowers using natural image statistics. In *CVPR*, 2010. 54, 55, 72

Fahad Shahbaz Khan, Joost van de Weijer, Andrew D.Badganov, and Maria Vanrell. Portmanteau vocabularies for multi-cue image representation. In *NIPS*, 2011. 69, 70, 72

Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Li Fei-Fei. Novel dataset for fine-grained image categorization. In *First Workshop on Fine-Grained Visual Categorization, CVPR*, 2011. 34, 35, 71, 110

Edward Kim, Hongsheng Li, and Xiaolei Huang. A hierarchical image clustering cosegmentation framework. 2012. 120

J. Krause, J. Deng, M. Stark, and L. Feifei. Collecting a large-scale dataset of fine-grained cars. In *fine-grained visual categorization workshop*, 2013. 110

A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *NIPS*, pages 1106–1114, 2012. 24, 25, 35, 95, 115

D. Kuettel, M. Guillaumin, and V. Ferrari. Segmentation propagation in imagenet. In *Proc. ECCV*, 2012. 120

M. P. Kumar, P. H. S. Torr, and A. Zisserman. OBJ CUT. In *CVPR*, 2005. 83

Neeraj Kumar, Peter N. Belhumeur, Arijit Biswas, David W. Jacobs, W. John Kress, Ida Lopez, and JoÃčo V. B. Soares. Leafsnap: A computer vision system for automatic plant species identification. In *The 12th European Conference on Computer Vision (ECCV)*, October 2012. 30, 31

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 23, 24

B. Leibe and B. Schiele. Interleaved object categorization and segmentation. In *BMVC*, 2003. 83

A. Levin and Y. Weiss. Learning to combine bottom-up and top-down segmentation. In *ECCV*, 2006. 83

Guangcan Liu, Zhouchen Lin, Xiaoou Tang, and Yong Yu. A hybrid graph model for unsupervised object segmentation. In *ICCV*, pages 1–8, 2007. 39, 57

D. G. Lowe. Object recognition from local scale-invariant features. In *ICCV*, 1999. 10, 11, 21, 41

S. Maji, J. Kannala, E. Rahtu, M. Blaschko, and A. Vedaldi. Fine-grained visual classification of aircraft. Technical report, 2013. 104

Tomasz Malisiewicz and Alexei A. Efros. Improving spatial support for objects via multiple segmentations. In *BMVC*, 2007. 36

C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008. 10

Lopamudra Mukherjee, Vikas Singh, and Chuck R. Dyer. Half-integrality based algorithms for cosegmentation of images. In *CVPR*, 2009. 36, 41

Minh Hoai Nguyen, Lorenzo Torresani, Fernando de la Torre, and Carsten Rother. Weakly supervised discriminative localization and classification: a joint learning process. In *ICCV*, 2009. 97

M-E. Nilsback. *An Automatic Visual Flora – Segmentation and Classification of Flowers Images*. PhD thesis, University of Oxford, 2009. 12, 32, 54, 55

M. E. Nilsback and A. Zisserman. A visual vocabulary for flower classification. In *CVPR*, 2006. 9, 10, 11, 32, 36, 74

M-E. Nilsback and A. Zisserman. Delving into the whorl of flower segmentation. In *BMVC*, 2007. 32, 39

M. E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *ICVGIP*, 2008. 19, 32, 33, 72, 118

M-E. Nilsback and A. Zisserman. Delving deeper into the whorl of flower segmentation. *Image and Vision Computing*, 2009. 32

M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Learning and Transferring Mid-Level Image Representations using Convolutional Neural Networks. In *Proc. CVPR*, 2014. 26

Vicente Ordonez, Jia Deng, Yejin Choi, Alexander C Berg, and Tamara L Berg. From large scale image categorization to entry-level categories. 2013. 122

Megha Pandey and Svetlana Lazebnik. Scene recognition and weakly supervised object localization with deformable part-based models. In *ICCV*, 2011. 84, 97

O. M. Parkhi, A. Vedaldi, C. V. Jawahar, and A. Zisserman. The truth about cats and dogs. In *Proc. ICCV*, 2011. 20

Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *CVPR*, 2012. 74, 91, 97

F. Perronnin and D. Dance. Fisher kernels on visual vocabularies for image categorization. In *Proc. CVPR*, 2007. 22

Florent Perronnin, Jorge Sánchez, and Thomas Mensink. Improving the fisher kernel for large-scale image classification. In *ECCV*, 2010. 23, 59, 66, 68, 85, 119

Esa Rahtu, Juho Kannala, Mikko Salo, and Janne Heikkilä. Segmenting salient objects from images and videos. In *ECCV*, 2010. 62

Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. "grabcut": interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*, 23(3), 2004. 38, 39, 40, 41, 54, 55, 56, 57, 77, 78, 80, 81

Carsten Rother, Thomas P. Minka, Andrew Blake, and Vladimir Kolmogorov. Cosegmentation of image pairs by histogram matching - incorporating a global constraint into mrfs. In *CVPR*, 2006. 36, 41

Bryan C. Russell, William T. Freeman, Alexei A. Efros, Josef Sivic, and Andrew Zisserman. Using multiple segmentations to discover objects and their extent in image collections. In *CVPR*, 2006. 43

Shai Shalev-Shwartz, Yoram Singer, and Nathan Srebro. Pegasos: Primal estimated sub-gradient SOlver for SVM. In *ICML*, volume 227, 2007. 43, 48

K Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *arxiv:1409.1556*, 2014. 115

Saurabh Singh, Abhinav Gupta, and Alexei A. Efros. Unsupervised discovery of mid-level discriminative patches. In *ECCV*, 2012. 97

J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proc. ICCV*, volume 2, pages 1470–1477, 2003. 10, 11

Sinisa Todorovic and Narendra Ahuja. Extracting subimages of an unknown category from a set of images. In *CVPR*, 2006. 39

Jasper R. R. Uijlings, Arnold W. M. Smeulders, and Remko J. H. Scha. What is the spatial extent of an object? In *CVPR*, pages 770–777, 2009. 26

Koen van de Sande, Jasper Uijlings, Theo Gevers, and Arnold Smeulders. Segmentation as selective search for object recognition. In *ICCV*, 2011. 61

A. Vedaldi and B. Fulkerson. Vlfeat – an open and portable library of computer vision algorithms. In *ACM int. conf. on Multimedia*, 2010. 51

A. Vedaldi and A. Zisserman. Efficient additive kernels via explicit feature maps. In *CVPR*, 2010. 51

Sara Vicente, Vladimir Kolmogorov, and Carsten Rother. Cosegmentation revisited: Models and optimization. In *ECCV*, 2010. 36, 41

Sara Vicente, Carsten Rother, and Vladimir Kolmogorov. Object cosegmentation. In *CVPR*, 2011. 36

C. Wah, S. Branson, P. Perona, and S. Belongie. Multiclass recognition and part localization with humans in the loop. In *Proc. ICCV*, 2011. 28, 29

Catherine Wah, GV Horn, Steve Branson, Subhransu Maji, Pietro Perona, and Serge Belongie. Similarity comparisons for interactive fine-grained categorization. *CVPR*, 2014. 29, 30

Jinjun Wang, Jianchao Yang, Kai Yu, Fengjun Lv, Thomas S. Huang, and Yihong Gong. Locality-constrained linear coding for image classification. In *CVPR*, 2010. 21, 44, 51, 66, 68, 85

P. Welinder, S. Branson, S. Belongie, and P. Perona. The multidimensional wisdom of crowds. In *NIPS*, 2010a. 113, 114

P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010b. 33, 34, 55, 68, 70, 111

John M. Winn and Nebojsa Jojic. Locus: Learning object classes with unsupervised segmentation. In *ICCV*, 2005. 39, 57, 97

Shulin Yang, Liefeng Bo, Jue Wang, and Linda G. Shapiro. Unsupervised template learning for fine-grained object recognition. In *NIPS*, 2012. 13, 14, 74, 83, 84, 97

Bangpeng Yao, Aditya Khosla, and Fei-Fei Li. Combining randomization and discrimination for fine-grained image categorization. In *CVPR*, 2011. 15, 70, 74, 84, 97

Ning Zhang, Ryan Farrell, and Trevor Darrell. Pose pooling kernels for sub-category recognition. In *CVPR*, 2012. 17, 19, 74, 83, 84, 89, 97

Ning Zhang, Ryan Farrell, Forrest Iandola, and Trevor Darrell. Deformable part descriptors for fine-grained recognition and attribute prediction. In *ICCV*, 2013. 15, 16

Ning Zhang, Jeff Donahue, Ross Girshick, and Trevor Darrell. Part-based r-cnns for fine-grained category detection. In *ECCV*, 2014. 26, 27, 122