

High-Level, Part-Based Features for Fine-Grained Visual Categorization

Thomas Berg

Submitted in partial fulfillment of the
requirements for the degree
of Doctor of Philosophy
in the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2017

©2017
Thomas Berg
All Rights Reserved

ABSTRACT

High-Level, Part-Based Features for Fine-Grained Visual Categorization

Thomas Berg

Object recognition—“What is in this image?”—is one of the basic problems of computer vision. Most work in this area has been on finding basic-level object categories such as *plant*, *car*, and *bird*, but recently there has been an increasing amount of work in “fine-grained” visual categorization, in which the task is to recognize subcategories of a basic-level category, such as *blue jay* and *bluebird*.

Experimental psychology has found that while basic-level categories are distinguished by the presence or absence of parts (a bird has a beak but car does not), subcategories are more often distinguished by the characteristics of their parts (a starling has a narrow, yellow beak while a cardinal has a wide, red beak). In this thesis we tackle fine-grained visual categorization, guided by this observation. We develop alignment procedures that let us compare corresponding parts, build classifiers tailored to finding the interclass differences at each part, and then combine the per-part classifiers to build subcategory classifiers.

Using this approach, we outperform previous work in several fine-grained categorization settings: bird species identification, face recognition, and face attribute classification. In addition, the construction of subcategory classifiers from part classifiers allows us to automatically determine which parts are most relevant when distinguishing between any two subcategories. We can use this to generate illustrations of the differences between subcategories. To demonstrate this, we have built a digital field guide to North American birds which includes automatically generated images highlighting the key differences between visually similar species. This guide, “Birdsnap,” also identifies bird species in users’ up-

loaded photos using our subcategory classifiers. We have released Birdsnap as a web site and iPhone application.

Table of Contents

List of Figures	iv
List of Tables	x
1 Introduction	1
2 Prior Work	6
2.1 Face Recognition	6
2.1.1 Alignment	6
2.1.2 Hierarchical Classifiers	8
2.2 Fine-Grained Visual Categorization	10
2.2.1 Part-based Features	10
3 Tom-vs-Pete Classifiers and Identity-preserving Alignment	12
3.1 Reference Dataset	15
3.2 Identity-preserving Alignment	16
3.3 Tom-vs-Pete Classifiers and Verification	20
3.4 Results	22
4 POOF: Part-Based One-vs-One Features	26
4.1 Part-Based One-vs-One Features	29
4.1.1 Implementation details	32
4.2 Experiments	33

4.2.1	Bird Species Identification	33
4.2.2	Face Verification	36
4.2.3	Attribute Classification	37
5	How Do You Tell a Blackbird from a Crow?	41
5.1	Related Work	44
5.2	Visual Similarity	45
5.2.1	Finding Similar Classes	46
5.2.2	Choosing Discriminative Features	47
5.2.3	Visualizing the Features	47
5.3	A Visual Field Guide to Birds	49
5.3.1	A Tree of Visual Similarity	52
6	Birdsnap	56
6.1	The Birdsnap Dataset	59
6.2	One-vs-Most Classifiers	61
6.3	A spatio-temporal prior for bird species	63
6.3.1	Adaptive kernel density estimation of the spatio-temporal prior	65
6.4	Experiments on the Birdsnap Dataset	69
6.5	Visualizing species frequency and migration	72
6.6	Illustrating field marks	74
6.7	A Tour of Birdsnap	77
6.7.1	The Birdsnap Web Site	77
6.7.2	The Birdsnap Mobile App	82
7	Conclusions	85
7.1	Recent Developments	86
	Bibliography	87

Appendix: The Birdsnap Dataset	100
A.1 Motivation Behind the Dataset	100
A.2 Building the Dataset	101
A.3 Comparisons with Other Datasets	108

List of Figures

2.1	Comparison of face alignments. Top row from left to right shows the original detected face, alignment by funnelling, and alignment by similarity. The bottom row shows affine alignment, piecewise affine alignment, and our identity-preserving warp, discussed in Chapter 3.	7
3.1	The verification system. A reference set of images is used to train a parts detector and a large number of “Tom-vs-Pete” classifiers. Then given a pair of test images, we detect the parts and used them to perform an “identity-preserving” alignment. The Tom-vs-Pete classifiers are run on the aligned images, with the results passed to a same-or-different classifier to produce a decision.	13
3.2	Labeled face parts. (a) There are fifty-five “inner” points at well-defined landmarks and (b) forty “outer” points that are less well-defined but give the general shape of the face. (c) The triangulation of the parts used to perform a piecewise affine warp.	15

3.3 Warping images to frontal. (a) Original images. (b) Aligning by an affine transformation based on the locations of the eyes, tip of the nose, and corners of the mouth does not achieve tight correspondence between the images. (c) Warping to put all 95 parts at their canonical positions gives tight correspondence, but de-identifies the face by altering its shape. (d) Warping based on genericized part locations gives tight correspondence without obscuring identity. In all methods, we ensure that the side of the face presented to the camera is on the right side of the image by performing a left-right reflection when necessary. This restricts the worst distortions to the left side of the image (shown with a gray wash here), which the classifiers can learn to weight less important than the right. 17

3.4 Finding generic parts. (a) The fiducial detector gives the inner part locations (yellow triangles) of the probe image. (b) For each reference subject, we find the image with inner parts closest, under similarity, to the detected probe parts. (c) Averaging the (inner *and* outer) part locations over this set of reference images gives the “generic” inner (blue circle) and outer (pink square) parts. (d) A close up of the eye shows that this subject’s eye is slightly longer (left-to-right) with less distance from eye to brow than the average eye. For clarity, only a subset of the full 95 parts are shown in this figure. 19

3.5 The top left image is produced by the alignment procedure. Each of the remaining images shows the region from which one low-level feature is extracted. SIFT descriptors are extracted from each square and concatenated. Concentric squares indicate SIFT descriptors at the same point but different scales. 20

3.6	(a) A comparison with the best published results on the LFW image-restricted benchmark, including the Associate-predict method [Yin <i>et al.</i> , 2011], Brain-inspired features [Pinto and Cox, 2011], and Cosine Similarity Metric Learning (CSML) [Nguyen and Bai, 2011], (b) The log scale highlights the performance of our method at the low-false-positive rates desired by many security applications.	23
3.7	LFW benchmark results. (a) The contribution of Tom-vs-Pete classifiers, compared to random projection or low-level features. (b) The contribution of the alignment method, compared with a piecewise affine warp using non-generic part locations or a global affine transformation.	24
4.1	Learning a Part-based One-vs-One Feature (POOF) for bird species identification. Given (a) a reference dataset of images labeled with class (species) and part locations, a POOF is defined by specifying two classes, one part for feature extraction, another part for alignment, and a low-level “base feature.” (b) Samples of the two chosen classes are taken from the dataset and (c) aligned to put the two chosen parts in fixed locations. (d) The aligned images are divided into cells at multiple scales, from which the base feature is extracted. A linear classifier is trained to distinguish the two classes, giving (e) a weight to each cell. We threshold the weights and find the maximal connected component contiguous to the chosen feature part, setting this as (f) the support region for the POOF. Finally, a classifier is trained on the base feature values from just the support region. The output of this classifier is our one-vs-one feature.	27
4.2	Bird species classification accuracy on (a) the full 200-species CUB benchmark and (b) the “birdlets” subset of 14 woodpeckers and vireos defined in [Farrell <i>et al.</i> , 2011].	31
4.3	Face parts from the detector of [Belhumeur <i>et al.</i> , 2011].	34

4.4	Results on the LFW benchmark. (a) POOFs and the top four previous published results. (b) Comparison of POOFs with low-level features.	36
5.1	(a) For any bird species (here the red-winged blackbird, at center), we display the other species with most similar appearance. More similar species are shown with wider spokes. (b) For each similar species (here the American crow), we generate a “visual field guide” page highlighting differences between the species.	42
5.2	A similarity tree of bird species, built from our visual similarity matrix. Species similar to the red-winged blackbird are in blue, and species similar to the Kentucky warbler are in red.	43
5.3	Visual field guide pages for the Kentucky warbler.	50
5.4	The phylogenetic “tree of life” representing evolutionary history. Species visually similar to the red-winged blackbird are in blue, and those similar to the Kentucky warbler are in red. Although the American crow and common raven are visually similar to blackbirds, they are not close in terms of evolution.	51
5.5	Similarity matrices. (a) Visual similarity. (b) Phylogenetic similarity. In both, rows/columns are in order of a depth-first traversal of the evolutionary tree, ensuring a clear structure in (b). The large dashed black box corresponds to the passerine birds (“perching birds,” mostly songbirds), while the small solid black box holds similarities between crows and ravens on the y-axis and blackbirds and cowbirds on the x-axis.	52
5.6	The top three visually similar, phylogenetically dissimilar species pairs from Table 5.1. First row: Gadwall and Pacific Loon. Second row: Hooded Merganser and Pigeon Guillemot. Third row: Red-breasted Merganser and Eared Grebe. Example images are chosen for similar pose.	54

6.1	The main, species-browsing page of the Birdsnap web site. Species can be arranged by the phylogenetic “Tree of Life” (shown), by visual similarity (as described in Section 5.3.1), by sighting frequency at the currently selected place and date (based on the spatio-temporal prior described in Section 6.3), or alphabetically.	57
6.2	The main screen of the Birdsnap iPhone app, a simpler version of the browsing wheel on the web site.	58
6.3	Sample images from the Birdsnap dataset, with bounding boxes and part annotations. The species of these samples, from left to right, are Northern Cardinal, Broad-tailed Hummingbird, Great Egret, Black-headed Grosbeak, and Nuttall’s Woodpecker.	59
6.4	One-vs-most classifiers (top) improve both overall accuracy and the consistency and “reasonableness” of classification results.	62
6.5	Fixed-time slices of our spatio-temporal prior show the Barn Swallow arriving from South America during its spring migration (left) and established in its summer grounds (right). Brighter regions indicate higher likelihood of a sighting.	64
6.6	One-vs-most accuracy omitting the k most similar classes from training. As we increase k , accuracy of the one-vs-most classifiers initially increases at all ranks. Results for additional values of k , shown in Table 6.1, are omitted for clarity.	68
6.7	Mean visual distance between query species and returned species. One-vs-most classifiers return species that are more similar to the query species. . .	70
6.8	The one-vs-most classifiers and spatio-temporal prior each contributes significantly to overall performance. The dashed line, using labeled part locations, shows hypothetical performance with human-level part localization. .	70

6.9	Species density over time in a fixed location. The “raw density” is the estimate from Section 6.3.1. Applying a median filter and adaptive threshold lets us recognize the Wild Turkey as present year round, despite the low frequency.	73
6.10	Field marks differentiating the Great Egret and the Snowy Egret. By filtering based on Tanimoto similarity, we ensure that we find three <i>different</i> features: beak color, the extension of the mouth beneath the eye, and the long, slender neck. In contrast, the top three features found by the method of Chapter 5 without filtering all appear to relate to beak color.	75
6.11	List view of species on the Birdsnap web site, here sorted by sighting frequency at the specified date and location.	78
6.12	Detail view for the Golden-winged Warbler on the web site (where it appears as a single, scrollable column).	79
6.13	Fields marks view from the web site, showing the differences between the Golden-winged Warbler and the Chestnut-sided Warbler with illustrations generated by the process described in Section 6.6.	81
6.14	The recognition submission window on the web site, after the user has clicked on the eye and tail.	82
6.15	Screens from the Birdsnap iPhone application.	83
A.1	The Amazon Mechanical Turk interface for bounding box labeling.	102
A.2	The Amazon Mechanical Turk interface for part labeling.	103
A.3	The Amazon Mechanical Turk interface for species and sub-species class labeling.	106

List of Tables

1.1	Winning mean average precision on the Pascal VOC Classification Challenge (Competition 1) [Everingham <i>et al.</i> , 2005 2013]	2
1.2	Winning top-5 accuracy on the ImageNET classification challenge. For 2015 and 2016 we show the best classification accuracy on the classification and localization challenge, as the separate classification challenge was discontinued after 2014.	3
4.1	Attribute classification accuracy. For each attribute, the top row is baseline accuracy using the low-level base features (color and gradient direction histograms) directly, and the bottom row is accuracy using POOFs. The more accurate is bold. The last column gives accuracy of [Kumar <i>et al.</i> , 2011] on the same test images, in bold when better than the POOF 600-sample classifier. The last row shows the average improvement using POOFs over the low-level features or [Kumar <i>et al.</i> , 2011]. As these are binary attributes, chance gives 50% accuracy.	38
5.1	Species pairs with high visual and low phylogenetic similarity.	53
6.1	Accuracy of the one-vs-most classifiers increases at all ranks as k increases to 15. Beyond $k = 15$, high-rank accuracy continues to increase, but rank-1 accuracy decreases.	67
A.1	Species of the Birdsnap dataset, with image and category counts. Part 1 of 3.	110

Acknowledgments

After spending a few years programming front-office systems in Tokyo, facing yet another project where the great challenge would be getting our clients to agree on the optimal tab-ordering of fields on the data entry screens, I began to wonder if learning a little computer science might put me in the way of more interesting work. So I convinced my girlfriend (now wife) to “visit” the States and headed back home to go to school. Thank you, Aya, for coming with me. I couldn’t have got through this on my own.

With no training in computer science, I didn’t interest the graduate schools that interested me, so I began taking classes at Harvard’s (open admissions) extension school. Two instructors in particular, Jamie Frankel and David Albert, fed my interest and had enough faith to write the recommendation letters I needed to get in to Columbia. Thank you, Jamie and David, for pushing me along.

At Columbia, I planned on a quick master’s degree, then back to work at a higher pay- and interest- grade. But in my first semester, I took some *very interesting classes*. Shree Nayar’s and Peter Belhumeur’s classes got me interested in computer vision and pattern recognition. Then a summer project with Peter Allen and Corey Goldfeder gave me a taste of research. Thank you, Shree and Peter and Peter and Corey, for showing me the good stuff. And especially thank you Peter, for letting me stick around when I decided I wanted to stay for a PhD.

I couldn’t seem to get anything to work my first couple years, but the advice, friendship, and examples of my fellow students kept me going and even made it fun. Thank you Neeraj Kumar for your enthusiasm and patience, Matei Ciocarlie for your welcome, Ollie Cossairt for your quiet excellence and hidden hippie outlook, Austin Reiter for your matter-of-fact

skills and speed, Jiongxin Liu for your discipline, persistence, and singing voice, Hao Dang for your cheer.

With an inexhaustible well of ideas and intuition from Peter, and his unerring nose for implementation bugs, our work began to work. Thank you Peter, for teaching me how to think clearly, how to write, and how to argue with you.

After a few years, with my graduation “imminent”, I left New York for California, remaining a Columbia student until I put the final touches on my thesis. These final touches have now taken over two years, during which Daniel Miao has leapt into the breach whenever a server went down or a disk failed. Thank you, Daniel, for being my eyes and hands on campus.

During all my time at Columbia, Anne Fleming has made the administrative bumps as smooth as they could possibly be. Daisy Nguyen has done the same for all my hardware troubles. Thank you Anne and Daisy.

This thesis is by all of us.

For Aya, as everything

Chapter 1

Introduction

Object detection and classification are among the most-studied problems in computer vision. Where are the objects in this image, and what are they? A lot of research effort has gone toward this problem, and a lot of progress has been made.

One measure of this progress is the PASCAL Visual Object Classes (VOC) challenges [Everingham *et al.*, 2014]. In the classification challenge, as administered each year from 2007 to 2012, we are presented with an image and asked to classify it according to the most prominent object it contains: an airplane, bicycle, boat, bus, car, motorbike, train, bottle, chair, dining table, potted plant, sofa, TV / monitor, bird, cat, cow, dog, horse, sheep, or person. The training set consists of 11,540 images across these classes. As shown in Table 1.1, mean average precision of the top-performing method rose from 59.4% in 2007 to 82.2% in 2012 [Everingham *et al.*, 2005–2013]. Although the challenge officially ended in 2012, later work using the 2012 dataset has achieved mean average precision as high as 85.4%, or 94.3% when using additional training data [Everingham *et al.*, 2016].

The heir to the PASCAL VOC challenges is the ImageNet Large Scale Visual Recognition Challenge (ILSVRC), run annually since 2010. In comparison with the PASCAL VOC, ILSVRC has one thousand categories instead of twenty and 1.2 million training images instead of twelve thousand. To account for the possibility of additional, unlabeled objects in the test images, classification is considered correct if any of five class labels pre-

Year	Winning mAP
2007	59.4%
2008	58.6%
2009	66.5%
2010	73.8%
2011	78.6%
2012	82.2%

Table 1.1: Winning mean average precision on the Pascal VOC Classification Challenge (Competition 1) [Everingham *et al.*, 2005–2013]

dicted by the algorithm match the test label. As shown in Table 1.2, this top-5 accuracy has improved each year, from 71.8% in 2010 to 97.0% in 2016 [Russakovsky *et al.*, 2015; Liu *et al.*, 2015; Liu *et al.*, 2016].

As object detection and classification improve, and further gains become harder to come by, it has become more important to explore the errors made by state-of-the-art methods and how these errors can inform further research. In an analysis of PASCAL VOC object detection results from two top methods, Hoiem *et al.* [Hoiem *et al.*, 2012] find the primary cause of false positives, to be “similar category” errors, in which a detector for one class fires on an instance of a similar class, for example, when the horse detector fires on a cow. These are the most common errors on average across all twenty classes, even though some classes (bottle, potted plant, and TV / monitor) have *no* similar classes and therefore *no* errors of this type at all. If we increase the number of categories, we expect this type of error to become more common, as each category will have more similar categories and the difference between similar categories will decrease. An analysis of the best results on the ILSVRC [Russakovsky *et al.*, 2013] confirms this, noting for example that the best method can distinguish dogs from non-dogs with 99% accuracy, but is much less reliable in distinguishing the 120 different dog breeds in the challenge from each other.

This problem of distinguishing very similar categories from each other is the problem

Year	Winning Top-5 Accuracy
2010	71.8%
2011	74.2%
2012	83.6%
2013	88.3%
2014	93.3%
2015	96.4%
2016	97.0%

Table 1.2: Winning top-5 accuracy on the ImageNET classification challenge. For 2015 and 2016 we show the best classification accuracy on the classification and localization challenge, as the separate classification challenge was discontinued after 2014.

of *fine-grained visual categorization* (FGVC) and is the topic of this thesis. Examples of fine-grained categorization include recognizing breeds of dog, species of bird, or models of automobile. This thesis focuses on the use of *parts* of the object for fine-grained recognition, for two reasons.

First, we are guided by results from the study of human perception. Psychologists use the terms “superordinate level,” “basic level,” and “subordinate level” to describe levels in the taxonomic hierarchy with which we label the objects in our environment. While not always perfectly well-defined, in general the basic level is the level at which we most readily recognize and label objects. For example, barring a reason to be more or less specific, we are more likely to say we saw a “car” (basic level) than a “vehicle” (superordinate level) or a “Toyota Camry” (subordinate level). Similarly we say “bird” rather than “animal” or “starling,” and “hammer” rather than “tool” or “ball-peen hammer.” In these terms, fine-grained categorization is the problem of distinguishing subordinate-level categories of the same basic-level category from each other.

Work in experimental psychology suggests that the basic level is often defined by the presence or absence of parts [Tversky and Hemenway, 1984]. A car has four wheels, a

bird has a beak and feathers, and a hammer has a head and a handle. Subordinate-level categories, in contrast, are often distinguished by characteristics of their parts: a starling has an orange beak and spotted feathers. If humans perform fine-grained categorization by considering characteristics of the parts, it's appealing to have our algorithms look to the parts as well.

Second, although FGVC is relatively young as a named area of research in computer vision, it has much in common with a very well-studied *instance*-level classification problem: face recognition. Instance recognition is often considered a different problem from FGVC, at one end of a granularity spectrum with basic-level recognition at the other end and FGVC in the middle. But the difficulty in fine-grained and instance-level recognition is essentially the same: small inter-class differences often swamped by intra-class differences, so we consider face recognition (and instance recognition in general) as an example of fine-grained categorization, where the basic-level category is *face* and the subordinate-level categories are individuals. The best methods of face recognition all include finding parts of the face (eyes, nose, etc.) so that corresponding parts of faces can be compared with each other, so it's natural to investigate whether the use of parts is important for fine-grained recognition in other domains as well.

In this thesis, we develop a set of part-based features for fine-grained visual categorization and demonstrate their application to several problems.

After discussing relevant prior work in Chapter 2, in Chapter 3 we consider the problem of face verification, matching faces by identity. Using a set of 95 parts on the face, we design a method for alignment that brings the faces into correspondence while preserving interclass differences, and a method for learning a set of stacked classifiers that distinguish one face from another. In Chapter 4, we generalize this to learn a set of part-based features we call "POOFs" to distinguish subcategories in any domain, and demonstrate the generalized method's effectiveness at classification of human faces and birds. Subcategorization using POOFs closely follows the intuition from experimental psychology, as each feature is built to measure a characteristic of a particular part, and these part-specific features are

combined to build the subcategory classifiers.

In Chapter 5, we consider applications of part-based features beyond automatic recognition, in particular how we can use these features to develop an understanding of the visual domain defined by a basic-level category. Again taking birds as our example, we automatically determine which subcategories (species) are most similar to each other and annotate images to show the key differences that distinguish similar species from each other. Finally, in Chapter 6, we describe the application of these ideas to build Birdsnap, a publicly-available digital field guide to birds, implemented as a web site and an iPhone App. We use the methods of Chapter 5 to build the guide, illustrating the similarities and differences between species, and use the methods of Chapter 4 to perform automatic identification of the birds in users' uploaded photos. This section describes the challenges we encountered when applying our methods to build a real, useful system, and the modifications to our methods that these challenges necessitated.

Chapter 2

Prior Work

2.1 Face Recognition

The “finest” fine-grained categorization is instance recognition, where we must identify individual instances of a class, and the best-studied example of instance recognition is face recognition. So we look to prior work on faces. The full body of work on face recognition is too large to survey here, so we focus on two aspects relevant to our work, alignment and hierarchical classifiers.

2.1.1 Alignment

It is well established that alignment is critical for good performance in face recognition with uncontrolled images ([Gu and Kanade, 2008; Wang *et al.*, 2006; Wolf *et al.*, 2009]). One method often applied is [Huang *et al.*, 2007a]’s “funneling,” which extends the congealing method of [Learned-Miller, 2006] to handle noisy, real-world images. These methods find transformations that minimize differences in images that are initially only roughly aligned. Another common technique is to apply a similarity or affine transformation to the images based on the locations of detected fiducial points such as the corners of the eyes and mouth. Due to both their effectiveness and the fact that pre-aligned images for the standard “Labeled Faces in the Wild” (LFW) face verification dataset are publicly avail-



Figure 2.1: Comparison of face alignments. Top row from left to right shows the original detected face, alignment by funnelling, and alignment by similarity. The bottom row shows affine alignment, piecewise affine alignment, and our identity-preserving warp, discussed in Chapter 3.

able, these alignments have become a standard part of the face recognition pipeline, with fiducial-based alignment in particular becoming popular after their initial use on LFW by [Kumar *et al.*, 2009] and [Wolf *et al.*, 2009]. We find that for our methods, which build many classifiers based on often quite small parts of the face, a global similarity or affine transformation does not give good enough correspondence.

We can achieve a closer correspondence with a more flexible transformation. We construct such a transformation by building a triangulation of the detected fiducial points and a triangulation of the same fiducial points in a target pose (we use an average over many frontal faces of different identities), and performing a piecewise affine warp mapping each

triangle of the former to the corresponding triangle in the latter. This approach has been used with fiducial detections based on active appearance models ([Cootes *et al.*, 2000; Edwards *et al.*, 1998; Asthana *et al.*, 2011a]) or 3D models ([Banz and Vetter, 2003; Asthana *et al.*, 2011b]), but neither of these methods has been reliably demonstrated on images captured in the wild and displaying simultaneous variation in pose, lighting, expression, occlusion, and image quality. We use the more robust detection of [Belhumeur *et al.*, 2011], but we find that by mapping fiducials of all faces to the same, frontal pose locations, we lose information relevant to identifying the subject. Some people have larger than average noses, and warping all noses to the same size just makes recognition more difficult. In Section 3.2, we develop a new, triangulation-based alignment method to account for this, warping to normalize pose without normalizing identity. Figure 2.1 shows a comparison between some of the earlier alignment methods with ours. Note that only the last two, triangulation-based methods are able to normalize for expression (close the mouth).

Some recent work [Schroff *et al.*, 2015] has shown that with very large datasets (millions of identities, hundreds of millions of images), simpler alignment (just translation and scale) can be sufficient. The methods in this thesis do not require such huge datasets.

2.1.2 Hierarchical Classifiers

[Wolpert, 1992] introduced the term “stacked generalizer” to refer to a classifier (or regressor) constructed by first training a collection of first-level classifiers on a problem, then training a second-level classifier that takes the outputs from the first level classifiers as input to produce a final classification. This general approach is widely used as a fusion method for combining the results from multiple classifiers, but the more interesting uses in face recognition involve training first-level classifiers on a problem that is different from, but related to, the final problem solved by the second-level classifiers. For example [Wolf *et al.*, 2008] apply a two-level classifier to the problem of face verification, in which two faces of subjects not seen at training time must be categorized as being the same or different iden-

tities. For each test pair, they train a small number of “one-shot” classifiers using one of the test images as the single positive sample and an additional fixed set of reference images as negative samples. The results from these classifiers, which operate on single faces, are used as features into the second-level “same-or-different” classifier operating on the pair of faces. [Yin *et al.*, 2011] take a similar approach but augment the positive training set with images from the reference set of subjects similar to the test images.

In both of these cases, specialized classifiers must be trained for each test sample, which limits the sophistication of the classifiers that can efficiently be used. [Kumar *et al.*, 2011] also adopt a two-level classifier structure for face verification, but use a set of attribute (gender, race, hair color, etc.) classifiers as the first stage. By using a fixed set of classifiers, they avoid the need to retrain for each test sample. However this approach requires a great deal of manual effort in choosing and labeling attributes.

Other well-performing verification methods that do not follow this precise two-level structure still follow the pattern of first learning how to extract features, then learning the same-vs-different classifier. For example, [Pinto and Cox, 2011] use a validation set of face pairs to experiment with a large number of features and choose those most effective for verification, then feed these to an SVM for verification of the test data. [Nguyen and Bai, 2011] split their training data and use one part to learn metrics in the feature space and the other to learn a verification classifier whose input is the learned distances between the image pairs. And [Wolf *et al.*, 2009] augment the one-shot classifiers above with “two-shot” classifiers that use both test images as positive samples, then use the margin width of the classifier as a feature of the pair for the verification classifier.

The features we develop in this thesis are classifier outputs, so when the features are used for classification we have, in effect, a hierarchical classifier. The construction of this hierarchical classifier is most similar to the “simile classifier”-based face verification of [Kumar *et al.*, 2011]. In their method the features are the outputs of a set of one-vs-all classifiers on classes in a reference set. By instead using one-vs-one classifiers we are able to build a much larger feature space, quadratic instead of linear in the number of subjects.

The large number of classifiers and the relative simplicity of the one-vs-one classification they must model also allows us to use fast linear SVMs, while the attribute and simile classifiers use an RBF kernel.

2.2 Fine-Grained Visual Categorization

Moving up a step from instance-level recognition, “fine-grained categorization” is usually taken to mean classification of subordinate categories within a single basic-level category. This area has been explored mostly in the context of species or breed recognition. Something almost all this work has in common is the use of some notion of “parts.”

2.2.1 Part-based Features

As discussed in the introduction, there is evidence from experimental psychology that humans use part-based features for fine-grained categorization. Work from computer vision also shows the success of part-based methods.

Perhaps due to the easy availability of the data, many fine-grained categorization papers perform their experiments in the domain of bird species identification, using the Caltech-UCSD Birds (CUB-200-2011) Dataset [Wah *et al.*, 2011b]. This dataset contains about twelve thousand photographs of birds, covering 200 species, with labels for species and the positions of fifteen parts (beak, wing, etc). Some work, for example [Duan *et al.*, 2012; Yao *et al.*, 2012; Yao *et al.*, 2011; Zhang *et al.*, 2012], attempts to find discriminative parts in the images without using these explicit part labels, but these methods have not been able to achieve the accuracy of a supervised, part-based approach where specific parts are explicitly localized at test time. Other work [Branson *et al.*, 2010; Wah *et al.*, 2011a] proposes interactive “human in the loop” approaches in which the system requests the location of the most discriminative parts from the user. [Farrell *et al.*, 2011] defines a set of just two coarse parts (the head and body) used to align the images, avoiding the use fine-scale part locations to define their features. These methods all avoid explicitly building

detectors for the fifteen, fine-scale points in the dataset, but we find that a recently developed part detection method ([Belhumeur *et al.*, 2011], the same method we use to detect fiducial points in faces) is sufficiently accurate that we can build detectors for these parts directly, and achieve substantially better accuracy by basing our features on those parts.

Additional work on species or breed identification has been demonstrated on trees [Kumar *et al.*, 2012], flowers [Nilsback and Zisserman, 2008], butterflies [Wang *et al.*, 2009; Duan *et al.*, 2012], dogs [Liu *et al.*, 2012; Parkhi *et al.*, 2012; Prasong and Chamnongthai, 2012], and stoneflies [Martinez-Munoz *et al.*, 2009]. All of this work in one manner or another builds classifiers on features extracted from particular parts of the objects to be recognized using part-specific fiducial detectors (as we do), general interest points detectors such as Harris or SIFT, or even by simply starting with an image of a single part – Kumar *et al.*'s Leafsnap system identifies trees from a photo of a leaf.

Chapter 3

Tom-vs-Pete Classifiers and Identity-preserving Alignment

We first consider fine-grained categorization in the context of faces, with an investigation into the *face verification* problem. In face verification, we are given two face images and must determine whether they are the same person or different people. The images may vary in pose, expression, lighting, occlusions, image quality, etc. The difficulty lies in teasing out features in the image that indicate identity and ignoring features that vary with differences in environmental conditions.

It should be easy to find features that correspond with identity. To distinguish Lucille Ball from Orlando Bloom, consider hair color. “Red hair” is a simple feature that consistently indicates Lucille Ball. To distinguish between Stephen Fry and Brad Pitt, the best feature might be “crooked nose.” With a sufficiently large and diverse set of these features, we should have a discriminating feature for almost any pair of subjects. Kumar *et al.* [Kumar *et al.*, 2009] explored this approach, calling these features “describable visual attributes,” implementing them as classifiers, and using them for verification. A limitation of the approach is that the set of reliable features can only be as big as the relevant vocabulary one can come up with and get training data labelers to consistently label.

In this chapter, we *automatically* find features that can distinguish between two people,

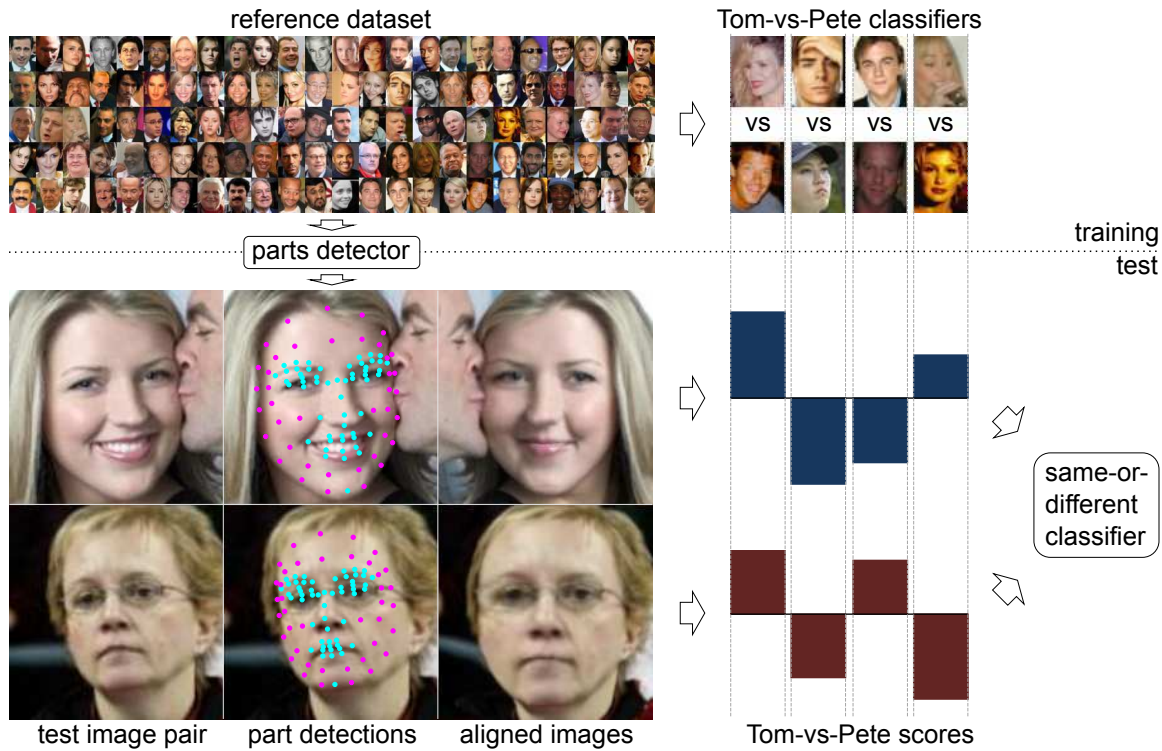


Figure 3.1: The verification system. A reference set of images is used to train a parts detector and a large number of “Tom-vs-Pete” classifiers. Then given a pair of test images, we detect the parts and used them to perform an “identity-preserving” alignment. The Tom-vs-Pete classifiers are run on the aligned images, with the results passed to a same-or-different classifier to produce a decision.

without requiring the features to be describable in words or requiring workers to label images with the feature. A simple way to find such a feature is to train a linear classifier to distinguish between two people. If the training data includes many images of each person under varied conditions, the projection found by the classifier will be insensitive to the conditions and consistently correspond to identity. We call classifiers trained in this way “Tom-vs-Pete” classifiers to emphasize that each is trained on just two individuals. We will show that they can be applied to *any* individual and used for face verification.

To demonstrate the Tom-vs-Pete classifiers, we collect a “reference set” of face images, labeled by identity and with many images of each subject. We build a library of Tom-vs-Pete classifiers by considering all possible pairs of subjects in the reference set. We then assemble a subset of these classifiers such that, for any pair of subjects, it is highly likely that we have at least a few classifiers able to distinguish them from each other. When presented with a pair of faces (of subjects *not* in the reference set) for verification, we apply these classifiers to each face and use the classifier outputs as features for a second-stage classifier that makes the “same-or-different” verification decision. Figure 3.1 shows an overview of the method.

To allow us to build a large and diverse collection of Tom-vs-Pete classifiers, and to make it more likely that each classifier will generalize beyond the two subjects it is trained on, each classifier looks at just a small portion of the face. These small regions must correspond to each other across images and identities for the classifiers to be effective, so alignment of the faces becomes particularly important. With this in mind, we adopt an alignment procedure, based on the detection of a set of face parts, that enforces a fairly strict correspondence across images. Our alignment procedure also includes a novel use of the reference dataset to distinguish geometric differences due to pose and expression, which should be normalized out by the alignment, from those that pertain to identity (such as thicker lips or a wider nose) and should be preserved. We call this an “identity-preserving alignment.”

We evaluate our method on the Labeled Faces in the Wild (LFW) [Huang *et al.*, 2007b],

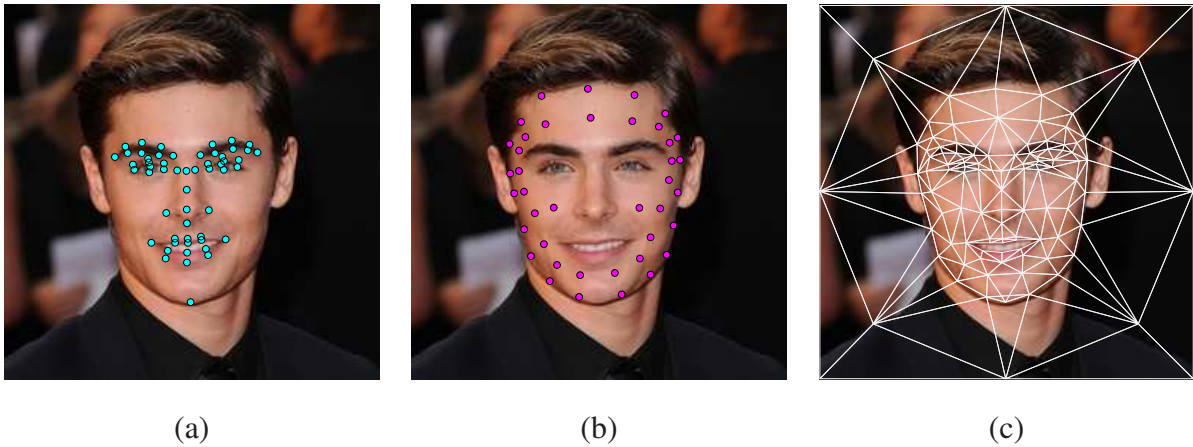


Figure 3.2: Labeled face parts. (a) There are fifty-five “inner” points at well-defined landmarks and (b) forty “outer” points that are less well-defined but give the general shape of the face. (c) The triangulation of the parts used to perform a piecewise affine warp.

a face verification benchmark using uncontrolled images collected from Yahoo News. We achieve an accuracy of 93.10%, reducing the error rate of the previous state of the art by 26.86%.

3.1 Reference Dataset

The identity-preserving alignment and the Tom-vs-Pete classifiers both rely on a dataset of reference face images labeled with identities and face part locations. We describe this dataset here for clarity of explanation in the following sections.

The dataset consists of 20,639 images of 120 people. So that we can train on our dataset and evaluate our methods on LFW, we ensure that none of the people in LFW are represented in our dataset. The images were collected by searching for the names of public figures on web sites such as Flickr and Google Images. We then filtered the resulting images by running a commercial face detector ([Omron,]) to discard images without faces and using human labelers via Amazon Mechanical Turk [Amazon, 2013] to discard images that were not of the target person, following the procedure outlined in [Kumar *et al.*, 2009].

In addition we removed the majority of “near-duplicate” images – images derived from the same original, but with different crops, compression, or other processing – following the method of [Pinto *et al.*, 2011], which is based on a simple image similarity measure. Images for 60 of the 120 people are from the “development” part of PubFig [Kumar *et al.*, 2009] dataset (which was collected as described above), while the remainder are new.

For all 20,639 images, we have obtained the human-labeled locations of 95 face parts, again using Mechanical Turk. Each point was marked by five labelers, with the mean of the three-label subset having the smallest variance taken as the final location. We divide the parts into two categories: a set of 55 “inner” parts that occur at edges and corners of relatively well-defined points on the face, such as the corners of the eyes and the tip of the nose, and a set of 40 “outer” parts that show the overall shape of the face but are less well-defined and so harder to precisely localize. The part locations are shown on a sample image in Figure 3.2.

3.2 Identity-preserving Alignment

We have constructed our alignment procedure with three criteria in mind. First, although our classification problem concerns pairs of faces, the Tom-vs-Pete classifiers used in the first stage operate on single faces. To accommodate this, all images must be aligned to a standard pose and expression. A “pairwise” alignment in which the images in each pair are brought into correspondence only with each other, which can produce less distortion than a single “all images” alignment, is not sufficient. We design our alignment to bring all images to a frontal pose with neutral expression.

Second, for the Tom-vs-Pete classifiers to be effective, the regions on which they are trained must have very good correspondence. This is because each classifier uses only a small part of the face, so the regions will have little or no overlap if the correspondence is not good, and because the linear nature of the classifiers makes it difficult to learn the more complex concepts that would be required to deal with poor alignment. Global similarity or affine transformations, for example, are not ideal, because they can bring only two or three

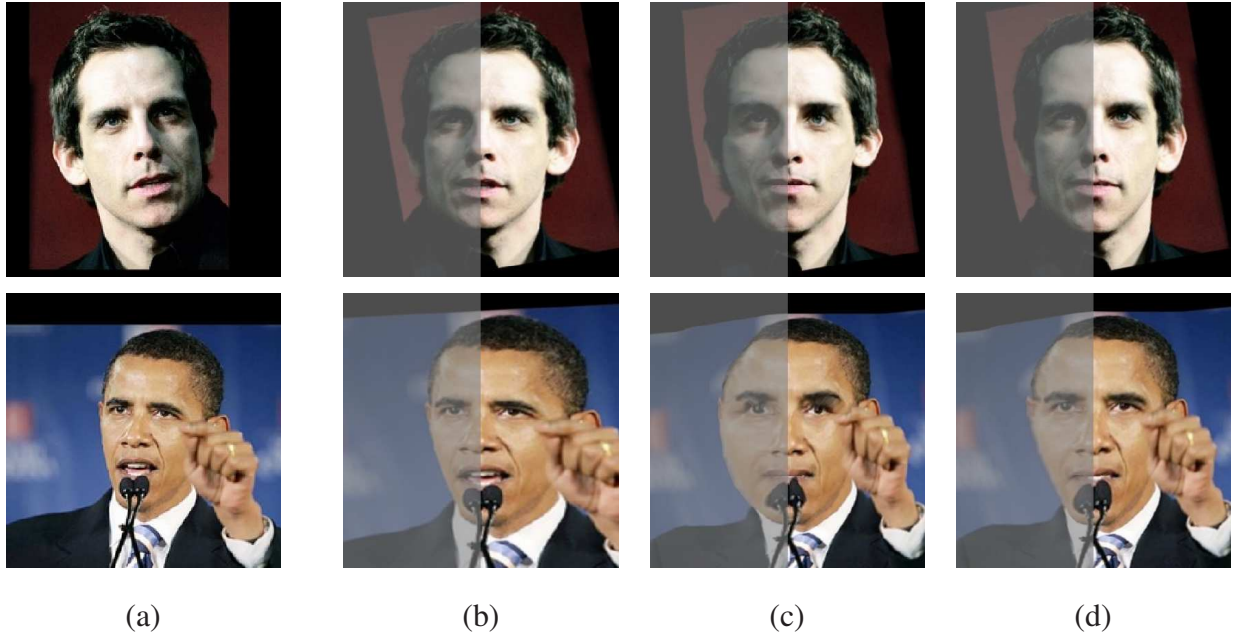


Figure 3.3: Warping images to frontal. (a) Original images. (b) Aligning by an affine transformation based on the locations of the eyes, tip of the nose, and corners of the mouth does not achieve tight correspondence between the images. (c) Warping to put all 95 parts at their canonical positions gives tight correspondence, but de-identifies the face by altering its shape. (d) Warping based on genericized part locations gives tight correspondence without obscuring identity. In all methods, we ensure that the side of the face presented to the camera is on the right side of the image by performing a left-right reflection when necessary. This restricts the worst distortions to the left side of the image (shown with a gray wash here), which the classifiers can learn to weight less important than the right.

points into perfect alignment, respectively.

Third, we must be careful not to *over-align*. The perfect alignment procedure for face recognition removes differences due to pose and expression but not those due to identity. Our alignment should turn faces to a frontal pose and close open mouths, but it should not warp a prominent jaw to a receding chin.

The alignment procedure we have designed to satisfy these criteria requires a set of part locations on each face. We use the ninety-five parts defined in the reference dataset. To find them automatically in a test image, we first use the detector of [Belhumeur *et al.*, 2011], which combines the results of an independent detector for each part with global models of the parts' relative positions, to detect the fifty-five inner parts. Then we find the image in the reference dataset whose inner parts, under similarity transformation, are closest in an L_2 sense to the detected inner parts, and “inherit” the outer part positions from that image. The parts detector is trained on a subset of the images in the reference set.

Each part also has a canonical location, where it occurs in an average, frontal face with neutral expression. To align the image, we adopt the piecewise affine warp often used with parts detected using active appearance models [Cootes *et al.*, 2000; Edwards *et al.*, 1998]. We take a Delaunay triangulation of the canonical part positions and the corresponding triangulation on the part positions in the image, then map each triangle in the image to the corresponding canonical triangle by the affine transformation determined by the three vertices. The three correspondences at the vertices of each triangle produce a unique, exact solution for the affine transformation, so all the parts are mapped perfectly to their canonical locations. Provided we have a sufficiently dense set of parts, this ensures the tight correspondence we require.

This system of alignment produces very tight correspondences and effectively compensates for pose and expression. However the warping is so strict, moving the ninety-five parts to *exactly* the same locations in every image, that features indicating identity are lost; the third criterion for our alignment is not satisfied. This can be seen in Figure 3.3 (c), whose images are somewhat anonymized compared with (a), (b), and (d). To understand why this

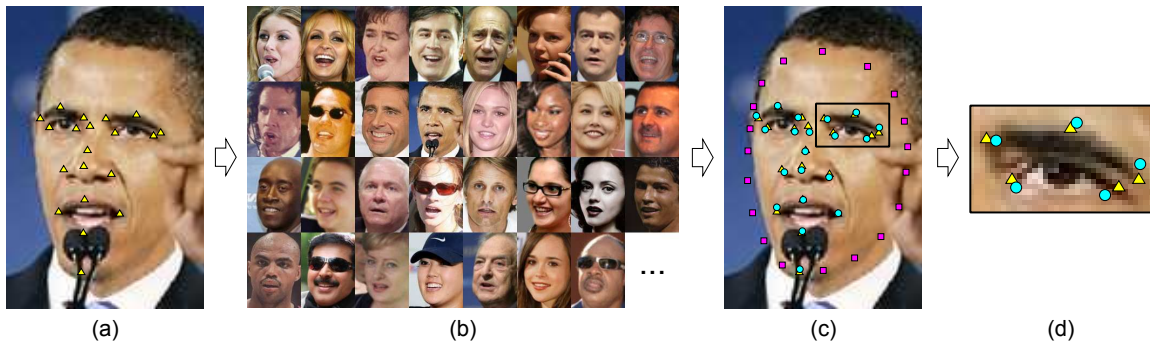


Figure 3.4: Finding generic parts. (a) The fiducial detector gives the inner part locations (yellow triangles) of the probe image. (b) For each reference subject, we find the image with inner parts closest, under similarity, to the detected probe parts. (c) Averaging the (inner *and* outer) part locations over this set of reference images gives the “generic” inner (blue circle) and outer (pink square) parts. (d) A close up of the eye shows that this subject’s eye is slightly longer (left-to-right) with less distance from eye to brow than the average eye. For clarity, only a subset of the full 95 parts are shown in this figure.

happens, note that since there are parts at both sides of the base of the nose, aligned images of all subjects will have noses of the same width. To avoid this over-alignment, we will perform the alignment based not on the part locations in the image itself, but on “generic” parts – where the parts would be for an average person with the pose and expression in the image. For a wide-nosed person, these points will be not on the edge of the nose but slightly inside, and the above-average width of the nose will be preserved by the piecewise affine warp.

To find the generic parts, we modify the procedure for locating the parts in a test image as illustrated in Figure 3.4. We run the detector of [Belhumeur *et al.*, 2011] to get the fifty-five inner part locations as before. Then we find the image with the most similar configuration of parts for each of the 120 subjects in the reference dataset. We include the additional forty outer parts of these images to get a full set of ninety-five parts for each of 120 reference faces. These represent the part locations of 120 different individuals with nearly the same pose and expression as the test image. We take the mean of the 120 sets

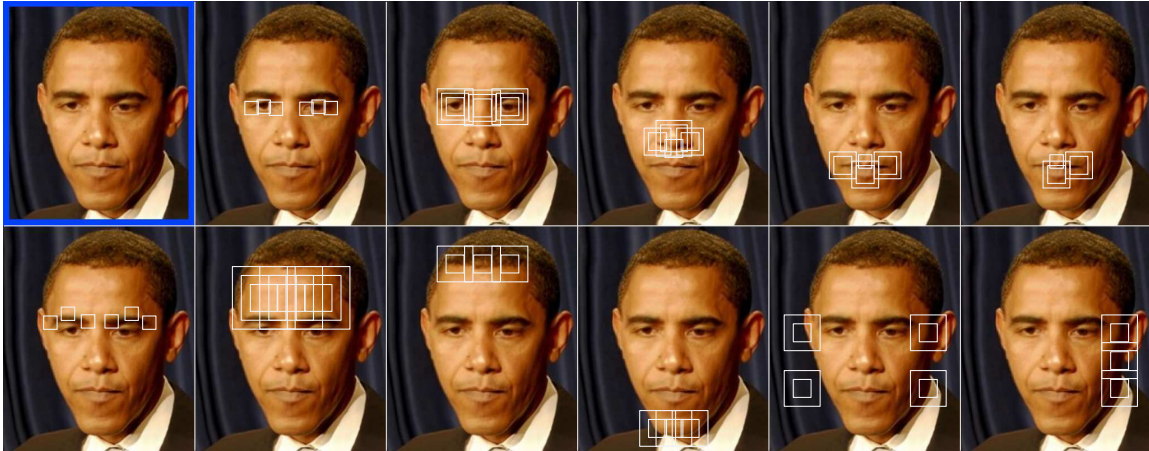


Figure 3.5: The top left image is produced by the alignment procedure. Each of the remaining images shows the region from which one low-level feature is extracted. SIFT descriptors are extracted from each square and concatenated. Concentric squares indicate SIFT descriptors at the same point but different scales.

of part locations to get the generic part locations for the test image. We use these generic part locations in place of the originally detected locations to produce an identity-preserving aligned image with a piecewise affine warp as described above.

For large yaw angles, we cannot produce a warp to frontal that looks good on the side of the face originally turned away from the camera. To reduce the difficulty this presents to the classifiers, we make a very simple guess at the yaw direction of the face (we use the detected parts to find the shorter eyebrow and assume the subject is facing that direction), then reflect the image if necessary so that all faces are facing the left side of the image. In this way, our classifiers can learn to assign more importance to the reliable, right side of the image.

3.3 Tom-vs-Pete Classifiers and Verification

Each Tom-vs-Pete classifier is a binary classifier trained using a low-level feature on images of two people from the reference dataset. If there are N subjects in the reference set and k

low-level features, we can train $k \cdot \binom{N}{2}$ Tom-vs-Pete classifiers.

In our experiments, each low-level feature is a concatenation of SIFT descriptors [Lowe, 1999] extracted at several points and scales in one region of the face. By limiting each classifier to a small region of the face, we hope to learn a concept that will generalize to individuals other than the two people used for training. The regions, shown in Figure 3.5, cover the distinctive features inside the face, such as the nose and eyes, as well as the boundary of the face. The classifiers are linear support vector machines trained using the LIBLINEAR package [Fan *et al.*, 2008].

For each face in a verification pair, we evaluate a set of Tom-vs-Pete classifiers and construct a vector of the signed distances from the decision boundaries. This vector serves as a descriptor of the face. Following the example of [Kumar *et al.*, 2009], we then concatenate the absolute difference and element-wise product of the two face descriptors and pass the result to an RBF SVM to make the same-or-different decision.

We use 5,000 Tom-vs-Pete classifiers to build the face descriptors. Experiments suggest that additional classifiers beyond this number are of little benefit. With 120 subjects in the reference dataset and 11 low-level features, we can train tens of thousands of classifiers, so we have to choose a subset. There are many reasonable ways to do so, but we design our procedure motivated by the desire for a subset of classifiers that (a) can handle a wide variety of subject pairs and (b) complement each other. To achieve the first, we will choose evenly from classifiers that excel at each reference subject pair. To achieve the second, we will use Adaboost [Freund and Schapire, 1997]. We begin by constructing a ranked list of classifiers for each subject pair (S_i, S_j) , as follows:

1. Let H_{ij} be the set $\{h_1, \dots, h_n\}$ of Tom-vs-Pete classifiers that are *not* trained on S_i or S_j .
2. Consider each h_k in H_{ij} as an S_i -vs- S_j classifier. Do this by fixing the subject with the greater median output of h_k as the positive class, then finding the threshold that produces equal false positive and false negative rates.
3. Treating H_{ij} as a set of weak S_i -vs- S_j classifiers, run the Adaboost algorithm. This

assigns a weight to each h_k .

4. Sort H_{ij} by descending Adaboost weights to get a list of classifiers L_{ij} . An initial subsequence of L_{ij} will be a set of classifiers, not trained on S_i or S_j , that combine effectively to distinguish S_i from S_j .

We construct an overall ordered list of classifiers, L , by taking the first classifier in each of the L_{ij} , then the second in each, then the third, etc. Within each group we randomly order the classifiers, and each classifier is included in L only the first time it occurs. To choose a subset of classifiers of any size, we take an initial subsequence of L .

3.4 Results

We evaluate our system on Labeled Faces in the Wild (LFW) [Huang *et al.*, 2007b], a face verification benchmark using images collected from Yahoo News. The LFW benchmark consists of 6,000 pairs of faces, half of them “same” pairs and half “different,” divided into ten folds for cross-validation. In our method the parts detection, alignment, and Tom-vs-Pete classifiers are based on the reference dataset, so the LFW training folds are used only to train the final same-vs-different classifier. Note that none of the subjects in our reference set appear in LFW, so neither the Tom-vs-Pete classifiers nor the parts detector have seen these individuals in training. We follow the “image-restricted” protocol, in which the training face pairs are marked only with a same or different label and not with the identities of each face (which would allow the generation of additional training pairs).

We obtain a mean accuracy of $93.10\% \pm 1.35\%$. LFW is widely reported on, with accuracies of twenty-five published methods listed on the maintainers’ web site [University of Massachusetts,] at time of writing. Figure 3.6 compares our performance with the top three previously published results. We achieve a 26.86% reduction in the error rate of the previous best results reported by Yin *et al.* [Yin *et al.*, 2011]. Figure 3.6 (b) demonstrates our performance at the low false positive rates required by many security applications. At 10^{-3} false positive rate we achieve a true positive rate of 55.07%, where the previous best

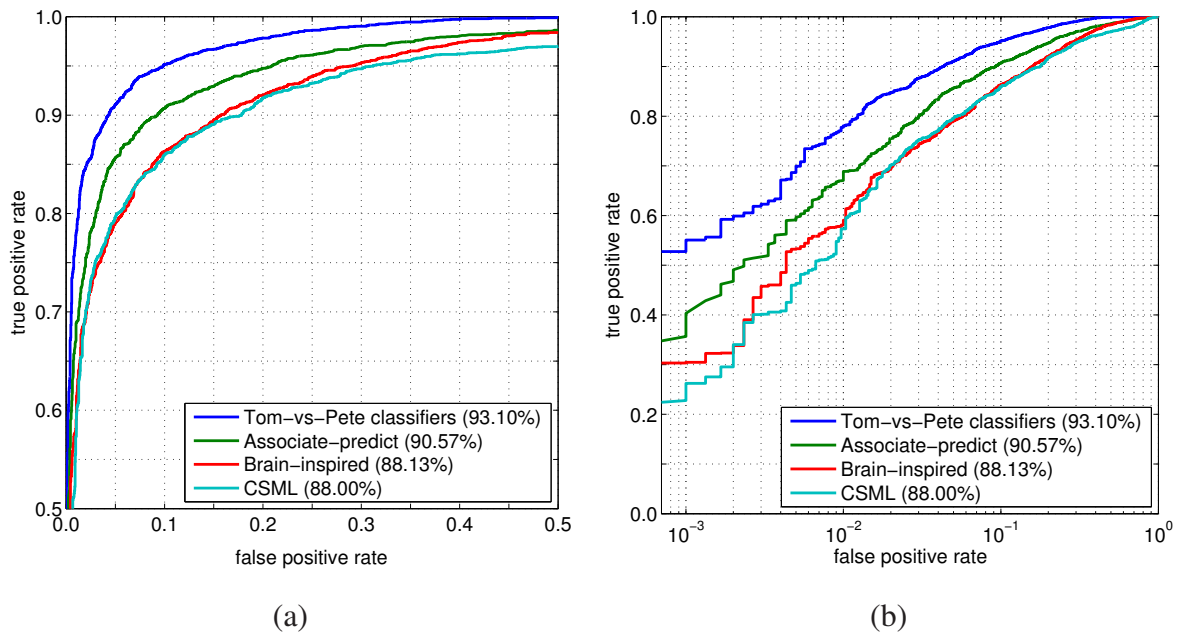


Figure 3.6: (a) A comparison with the best published results on the LFW image-restricted benchmark, including the Associate-predict method [Yin *et al.*, 2011], Brain-inspired features [Pinto and Cox, 2011], and Cosine Similarity Metric Learning (CSML) [Nguyen and Bai, 2011], (b) The log scale highlights the performance of our method at the low-false-positive rates desired by many security applications.

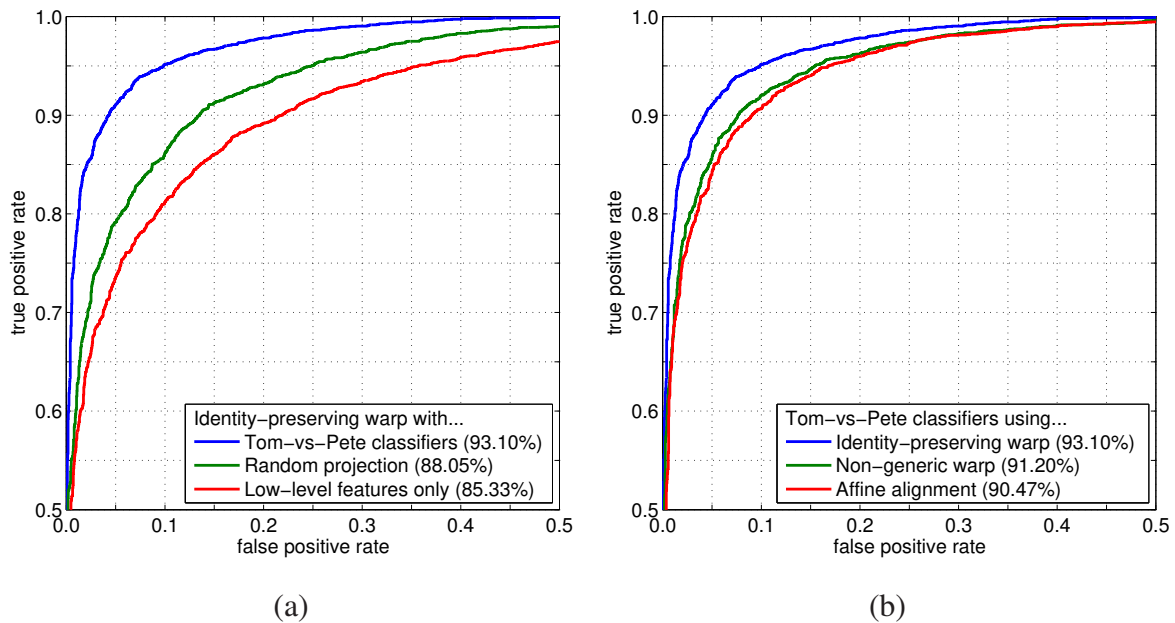


Figure 3.7: LFW benchmark results. (a) The contribution of Tom-vs-Pete classifiers, compared to random projection or low-level features. (b) The contribution of the alignment method, compared with a piecewise affine warp using non-generic part locations or a global affine transformation.

is 40.33% [Yin *et al.*, 2011].

Kumar *et al.* [Kumar *et al.*, 2009] have made the output of their attribute classifiers on the LFW images available on the LFW web site [University of Massachusetts,]. These classifiers are similar to our Tom-vs-Pete classifiers, but are trained on images hand labeled with attributes such as gender and age. Appending the attributes classifier outputs to our vector of Tom-vs-Pete outputs boosts our accuracy to $93.30\% \pm 1.28\%$.

Our method is efficient. Training and selection of the Tom-vs-Pete classifiers is done offline. The eleven low-level features are constructed from SIFT descriptors at a total of just 34 points on the face, at one to three scales each. These features are shared by all of the Tom-vs-Pete classifiers. Evaluation of each Tom-vs-Pete classifier requires evaluation of a single dot product. Finally, the RBF SVM verification classifier must be evaluated on a single feature vector.

To demonstrate the relative importance of each part of our system, we run the benchmark with several stripped-down variants of the algorithm:

- **Random projection:** Replace each Tom-vs-Pete classifier with a random projection of the low-level feature it uses. Shown in Figure 3.7 (a).
- **Low-level features only:** Discarding the Tom-vs-Pete classifiers, concatenate the low-level features to produce a descriptor of each face, and use the absolute difference of these descriptors as the feature vector for the same-or-different classifier. Shown in Figure 3.7 (a).
- **Non-generic warp:** Train and use Tom-vs-Pete classifiers, but use the detected part locations directly in a piecewise affine warp, rather than the genericized locations that produce the identity-preserving warp. Shown in Figure 3.7 (b).
- **Affine alignment:** Train and use Tom-vs-Pete classifiers, but align all images with global affine transformations based on the detected locations of the eyes and mouth instead of our identity-preserving warp. Shown in Figure 3.7 (b).

Figure 3.7 includes ROC curves from these experiments and from the full system, showing that each part of the method contributes to the high accuracy.

Chapter 4

POOF: Part-Based One-vs-One Features

From instance recognition, and face recognition in particular, we now turn to more general fine-grained visual categorization.

Some of the most accurate prior approaches to fine-grained categorization are based on detecting and extracting features from particular parts of the objects. For example, in dog breed classification one may extract features from the nose and base of the ears [Liu *et al.*, 2012; Prasong and Chamnongthai, 2012]. Intuitively, fine-grained categorization calls for part-based approaches because the differences between subcategories are small and localized. Fine-grained visual categorization also *enables* part-based approaches, because objects within the same basic-level category usually have the same parts [Tversky and Hemenway, 1984], allowing for easier comparison. For example in dog breed recognition, since all dogs have noses, we can compare features extracted from the nose. In basic-level categorization this approach is difficult, as there is no natural corresponding part among instances of dogs, motorboats, and staplers.

Computer vision has produced a wide array of standard features, including SIFT [Lowe, 1999], SURF [Bay *et al.*, 2006], HOG [Dalal and Triggs, 2005], LBP [Ojala *et al.*, 2002], etc. A straightforward approach to part-based recognition is to extract such features at the part locations and build a classifier. In general, however, these standard features are unlikely to be optimal for any particular problem; what is best may vary both by domain

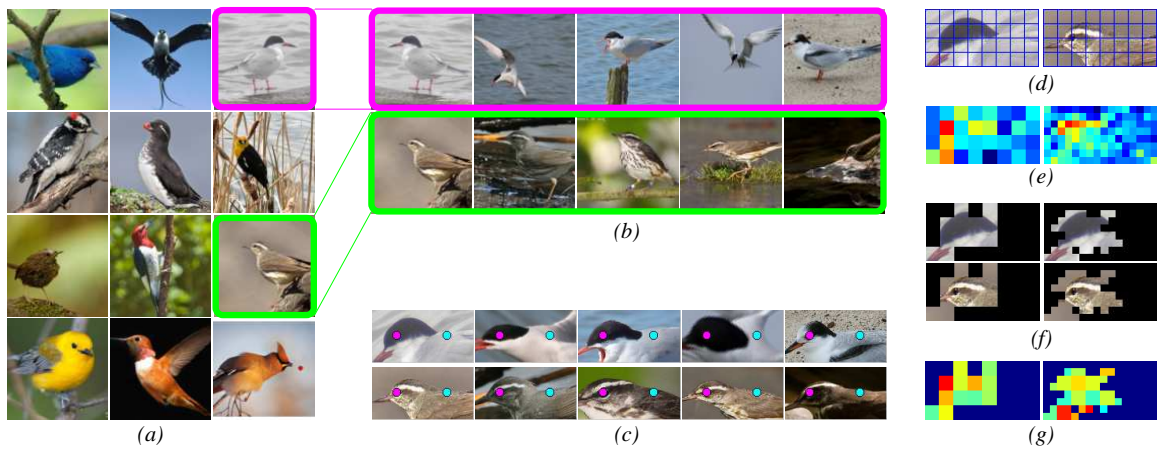


Figure 4.1: Learning a Part-based One-vs-One Feature (POOF) for bird species identification. Given (a) a reference dataset of images labeled with class (species) and part locations, a POOF is defined by specifying two classes, one part for feature extraction, another part for alignment, and a low-level “base feature.” (b) Samples of the two chosen classes are taken from the dataset and (c) aligned to put the two chosen parts in fixed locations. (d) The aligned images are divided into cells at multiple scales, from which the base feature is extracted. A linear classifier is trained to distinguish the two classes, giving (e) a weight to each cell. We threshold the weights and find the maximal connected component contiguous to the chosen feature part, setting this as (f) the support region for the POOF. Finally, a classifier is trained on the base feature values from just the support region. The output of this classifier is our one-vs-one feature.

(the best features for dogs are different from the best features for birds) and by task (the best features for face recognition are different from the best features for gender classification).

In this chapter, we build a framework for learning a large set of discriminative, intermediate-level features, which we call *Part-based One-vs-One Features (POOFs)*, specialized for a particular domain and set of parts. POOFs are a generalization of the Tom-vs-Pete classifiers from the previous chapter. The process of learning these features is illustrated in Figure 4.1. We start with a dataset of images in the domain, with class and part location labels. For any pair of classes, for any pair of parts, we extract some low-level features in a grid of cells that covers the two parts, and train a linear classifier to distinguish the two classes from each other. The weights assigned by this classifier to different cells of the grid indicate the most discriminative region around these parts for this pair of classes. We fix the support region based on these weights, then retrain the classifier to find a discriminative projection. The combination of the two parts, the low-level feature, the learned support region, and the final projection form a POOF, which can produce a scalar score (the decision value from the classifier) for any test image with locations for the two parts. This score is our intermediate-level feature.

In this chapter we make the following contributions:

- We present a fully automatic method for constructing a library of *Part-based One-vs-One Features (POOFs)* – discriminatively trained intermediate-level features – from a set of images with class and part location labels
- We demonstrate that POOFs significantly advance the state of the art on the Caltech-UCSD Birds dataset, obtaining a classification accuracy of 73.3% on the localized species categorization benchmark, quadrupling the accuracy reported in [Wah *et al.*, 2011b].
- We demonstrate that POOFs reduce the need for large training sets, showing that in the face domain they can be used as extremely effective intermediate features for tasks such as attribute labeling.

While each POOF is only trained to be discriminative for the two classes used in its definition, we find that collections of POOFs are useful not only for classification into the classes in the reference dataset, but for other tasks in the same domain. We show examples in two domains, bird species and faces.

4.1 Part-Based One-vs-One Features

Our method requires as input a reference dataset of images belonging to the domain under study, annotated with class labels and part locations. Parts are represented simply as points (x, y) in the image, and it is not necessary that all parts be present in all images. The output of our method is a set of discriminative features we call *Part-based One-vs-One Features*, suitable for many tasks in this domain. If the task at hand is supervised classification, the reference dataset may simply be the training set, but it need not be. It can also be a separate dataset labeled with classes different from those in the classification task. We show examples of this in Sections 4.2.2 and 4.2.3.

Given the reference set, the process of POOF learning is fully automatic. The method is illustrated in Figure 4.1, and is motivated overall by the goal of building a *discriminative* and *diverse* set of features. Let the reference set consist of images in N classes $\{1, \dots, N\}$, each image labeled with P parts. Each POOF we will learn is defined by

- the selection of two distinct reference classes, $i, j \in \{1, \dots, N\}$ with $i \neq j$,
- one part for feature extraction, $f \in \{1, \dots, P\}$,
- one other part for alignment, $a \in \{1, \dots, P\}$, with $a \neq f$, and
- a low-level *base feature*, b , which can be extracted from windows in the image. We term this a “base” feature to distinguish it from the higher-level feature we are learning. In the current implementation we use two base features: gradient direction histograms and color histograms.

We write $T_{f,a,b}^{i,j}$ for the POOF built based on these parameters; it is a function that extracts a single, scalar score from any image in the domain, and in combination the $T_{f,a,b}^{i,j}$ form a powerful feature space. We learn how to extract $T_{f,a,b}^{i,j}$ by the following procedure.

1. The POOF will be learned based on the reference images of classes i and j . We first take all these images, exclude those in which either part f or part a is not visible, and perform a similarity transform to bring points f and a to fixed positions. The transformed image is then cropped to a rectangular region enclosing points f and a . Depending on whether points f and a are close to or far from each other on images in this domain, $T_{f,a,b}^{i,j}$ will learn a fine-scale or coarse-scale feature.
2. We tile the cropped images with a grid of *feature cells*, and extract the base feature from each cell. We do multiple tilings, each using grid cells of a different size, and so extracting features at a different scale.
3. For the tiling at each scale, we train a linear support vector machine to distinguish class i from class j , based on the concatenation of the base feature values over the grid.
4. The trained SVM weight vector gives weights to every dimension of the base feature in every grid cell. We assign to each grid cell in each tiling the maximum absolute SVM weight over the dimensions in the feature vector that correspond to that cell. By thresholding these weights, we obtain a mask on the aligned images that defines the grid cells that are most discriminative between class i and j .
5. Starting with the grid cell containing part f as a seed, we find the maximum connected component of grid cells above the threshold in each tiling. This will act as a mask on the aligned image, defining at each scale a discriminative region around part f . By restricting the region to a connected component of f , we force POOFs with different feature parts to use different regions, encouraging diversity across the set of POOFs.

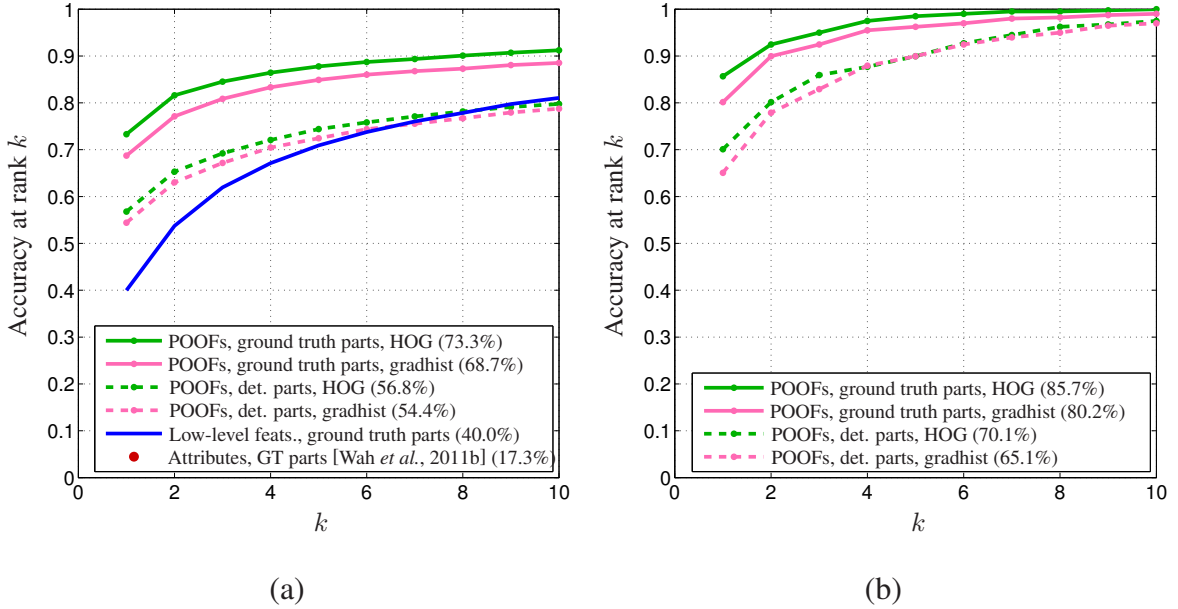


Figure 4.2: Bird species classification accuracy on (a) the full 200-species CUB benchmark and (b) the “birdlets” subset of 14 woodpeckers and vireos defined in [Farrell *et al.*, 2011].

- The low-level feature associated with $T_{f,a,b}^{i,j}$ is the concatenation of the base feature at the masked cells in all the tilings. Using this feature and all aligned images of classes i and j , we train another linear SVM. This SVM learns a projection of the masked, multiscale, local feature to a single dimension. This projection is $T_{f,a,b}^{i,j}$.

To extract feature $T_{f,a,b}^{i,j}$ from a new image with part locations, we proceed through the steps above again. The new image is aligned by similarity to put parts f and a in standard locations, then the base-level feature is extracted from just the masked cells of the tilings at each scale. The resulting vector is evaluated by the SVM to get a scalar projection value, which is the POOF score.

Note that switching i and j simply reverses the sign of the feature (i is taken as the “positive” class when training the SVMs). To avoid redundancy, we restrict ourselves to $i < j$. In contrast, parts f and a play different roles in constructing the POOF, so it may be useful to have both $T_{1,2,b}^{i,j}$ and $T_{2,1,b}^{i,j}$.

4.1.1 Implementation details

In our implementation, we use the following settings.

- In the alignment, the two parts are placed in a horizontal line 64 pixels apart. The crop is centered at the midpoint of the two parts, and is 64 pixels tall and 128 pixels wide.
- We use two scales of grid for the base feature extraction, with 8 x 8 and 16 x 16-pixel cells.
- We use two base features. The first is a gradient direction histogram. This feature comes in two variants. For the “gradhist” variant, we extract an 8-bin gradient direction histogram from each grid cell, then concatenate the histograms over all cells (or in the final $T_{f,a,b}^{i,j}$, over just the masked cells). For the “HOG” variant, we use Dalal and Triggs’ histogram of oriented gradients [Dalal and Triggs, 2005] feature, as modified by Felzenszwalb *et al.* [Felzenszwalb *et al.*, 2010] to include a dimensionality reduction step and the concatenation of histograms of signed and unsigned gradient. This gives us a nine-bin unsigned gradient direction histogram, an 18-bin signed gradient orientation histogram, and 4 normalization constants, for, in total, a 31-dimensional feature for each grid cell. These are concatenated as in the gradhist variant.

The second base feature is a color histogram. We use the same grids as for the gradient direction histograms, assigning each pixel to one of 32 color centers to form a histogram of length 32. The color histograms are then concatenated as with the gradient orientation histograms. The color centers are obtained by running k-means in RGB space on the pixels in the aligned and cropped region for all the images in the reference set, so the color centers are a function of f and a .

- For the SVM weight threshold we use the median absolute weight. This has the effect of masking out half of the region in Step 4 (which is further reduced when we restrict the region to a connected component contiguous with part f).

4.2 Experiments

To demonstrate the value and applicability of POOFs, we apply them to three problems. In Section 4.2.1, we consider bird species identification, building a set of POOFs from the training set, and applying them to recognition. In Section 4.2.2 we apply our method to face verification on unseen face pairs, building POOFs on a set of faces of different people than the test faces, demonstrating that our features learn to discriminate over the domain of images in general and not just over the particular classes from which they are built. In Section 4.2.3, we apply the POOFs built in Section 4.2.2 to attribute classification, and find that they are useful even when the classification task is on a different type of classes (attributes) than the classes on which they were learned (subject identities).

4.2.1 Bird Species Identification

The Caltech-UCSD Birds 200-2011 dataset [Wah *et al.*, 2011b], or CUB-200-2011 contains 11,788 photographs of birds spanning 200 species. Each image is labeled with its species, a bounding box for the bird, and the image coordinates of fifteen parts: the back, beak, belly, breast, crown, forehead, left eye, left leg, left wing, nape, right eye, right leg, right wing, tail, and throat. The images are split into training and test sets, with about 30 images per species in the training set, and the remainder in the test set. The authors propose several benchmarks for species recognition and part detection. Here, we evaluate on the “localized species categorization” benchmark, in which the part locations for all images are provided to the algorithm, and the task is, given the species labels on the training images, to determine the species of the test images. We also include results using an automatic parts detector in place of the ground truth positions.

There are very few images in the dataset with all fifteen parts visible. In particular, most birds have only one eye and one wing visible. When a part is not visible, it is labeled as such, with no position given. To better be able to make correspondences between parts, we preprocess the images, performing a left-right reflection on any image in which the right eye is visible but the left is not. This gives us a dataset in which almost all of the images

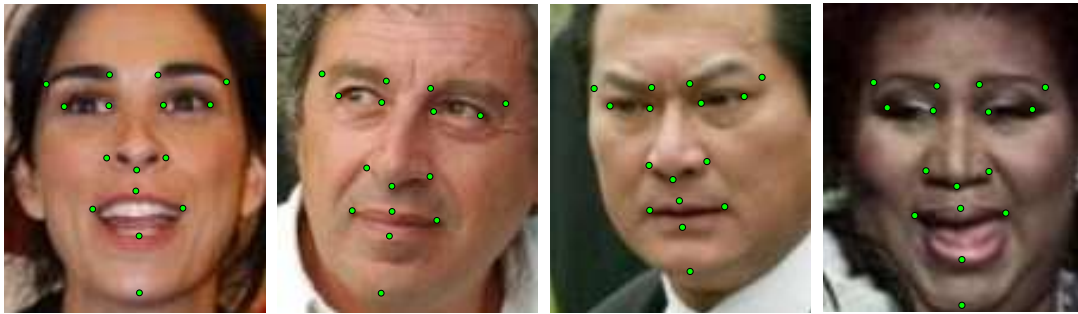


Figure 4.3: Face parts from the detector of [Belhumeur *et al.*, 2011].

have the left eye labeled (a few images have neither eye visible). We then disregard the (usually missing) right eye, right wing, and right leg parts.

To apply POOFs to this problem, we take the training set as our reference set. There are 200 classes, twelve parts, and two base features, yielding $\binom{200}{2} \cdot 12 \cdot 11 \cdot 2 = 5,253,600$ possibilities if we exhaustively learn features for all (i, j, f, a, b) . Instead, we randomly choose 5000 sets of parameters and learn just those features. We then extract the POOF scores from the training and test images, obtaining a feature vector of length 5000 for each image. Using this feature, we train a set of 200 one-vs-all linear SVMs to classify species. For each image, we rank the 200 species from highest to lowest classifier response. Taking the top ranked species for each image, we achieve a classification accuracy of 68.7% using the gradhist variant of the gradient feature, or 73.3% using the HOG variant.

While the localized species categorization protocol defined in [Wah *et al.*, 2011b] uses the ground truth part locations, this does not give *automatic* classification performance. To evaluate automatic classification, we rerun the experiment using automatically detected part locations on the test data in place of the ground truth locations. We use part locations from the part detector of [Belhumeur *et al.*, 2011] on images cropped to the bounding boxes of the birds, allowing us to compare with previous work that uses the bounding boxes but not the part labels. Using these detected part locations, we obtain a classification accuracy of 54.4% with the gradhist variant or 56.8% with HOG. The rate at which the correct species is in the top k ranked species is shown in Figure 4.2. For comparison with prior work,

we also show our results when restricted to the 14-species “birdlets” subset of the dataset defined in [Farrell *et al.*, 2011]. Our rank-1 classification accuracy on this subset using the gradhist variant is 80.2% using the ground truth parts and 65.1% using the detected parts, or 85.7% and 70.1% using HOG.

To show the benefit of the POOFs, we contrast our one-vs-all species classifiers with classifiers trained in a similar way, but without the POOFs. The POOFs are built using histograms of gradient direction and color over spatial grids covering the parts as the base features. For comparison, we build species classifiers that operate directly on the concatenation of these base features over all twelve parts. As with the POOF-based species classifiers, these classifiers are linear SVMs. These classifiers achieve a rank-1 accuracy of 40.0%.

Baseline accuracy on the localized species categorization benchmark reported in [Wah *et al.*, 2011b] is 17.31%, barely a quarter of our accuracy. To our knowledge, ours is the first subsequent work strictly following this protocol. However there are several pieces of work on this dataset reporting results of different experiments with which we can make comparisons.

Our result of 56.8% based on automatically detected parts uses only the ground truth bounding boxes, as does all the previous work cited here, and is far higher than any previous results on the full 200-species dataset, although there are differences in the experiments that make some of the comparisons imperfect. [Branson *et al.*, 2010] and [Yao *et al.*, 2011] report rank-1 accuracies, of 19% and 19.2% using multiple kernel learning and random forests respectively. However they use an earlier version of the dataset [Welinder *et al.*, 2010] with less training data. [Yao *et al.*, 2012] reports 44.73% mean average precision on the birdlets subset using the earlier version of the dataset (our mAP with HOG on the birdlets subset is 85.6% using ground truth parts or 70.2% using detected parts). Only [Duan *et al.*, 2012] and [Zhang *et al.*, 2012] report on the later version of the dataset. The former does not include results on the full 200-species set or the known birdlets subset, however the highest accuracy they report is 55%, on a five-species subset, very close to our

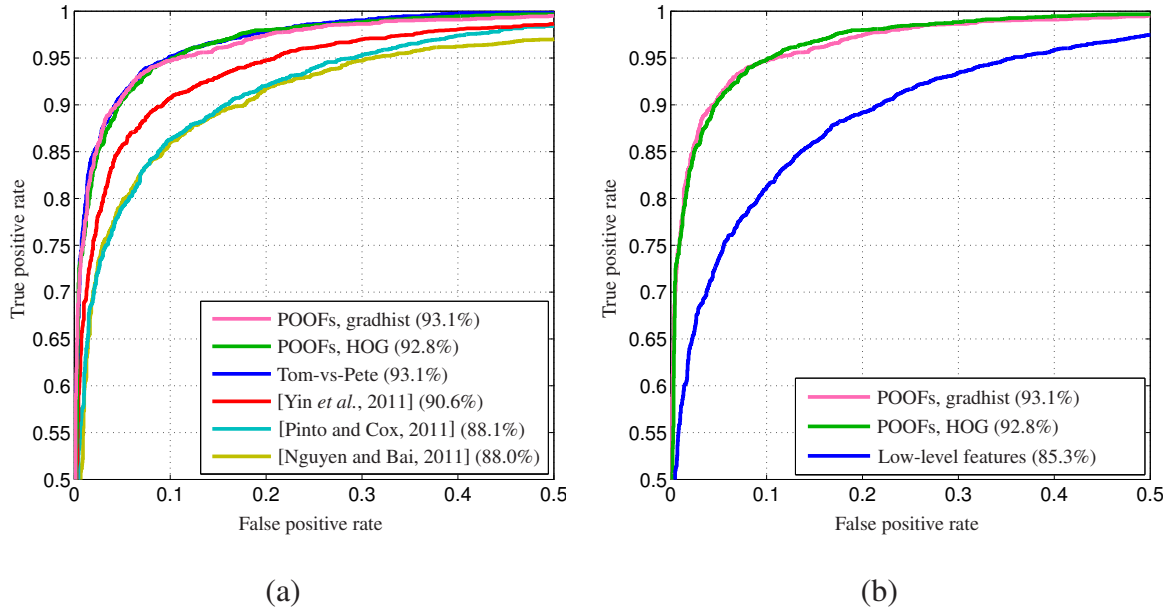


Figure 4.4: Results on the LFW benchmark. (a) POOFs and the top four previous published results. (b) Comparison of POOFs with low-level features.

automatic result on the much more difficult 200-species set. The latter is the most directly comparable to our work, reporting mean average precision of 28.18% on the 200-species benchmark and 57.44% on the birdlets subset. Our comparable mean average precisions with HOG are 56.9% and 70.2% respectively.

4.2.2 Face Verification

We now return to the face verification problem to determine whether the more general POOFs are competitive with the Tom-vs-Pete classifiers of Chapter 3, which were designed particularly for faces. Recall that in face verification, we are given two face images, of people not encountered at any training stage, and must determine whether they are two images of the same person or images of two different people. Because we must deal with previously unseen faces, there is no training set of images belonging to the classes we will be faced with at test time, as there was in the bird species identification experiments, where

we could learn our features based on the training set. Here, as in Chapter 3, we learn the features from a set of face images (the reference set described in Section 3.1), entirely separate from the evaluation dataset, in the belief that the features we discover are generally applicable to the face domain. As before, we evaluate on the Labeled Faces in the Wild (LFW) benchmark, using images that have been processed with the identity-preserving alignment of Section 3.2.

Except for using POOFs in place of the Tom-vs-Pete classifiers, we follow the procedure of Chapter 3. For each verification pair of images (I, J) , we get 10,000-dimensional POOF score vectors $f(I)$, $f(J)$. We then represent the pair by the concatenation of $|f(I) - f(J)|$ and $f(I) \cdot f(J)$ (where the subtraction and multiplication are performed elementwise) to get a 20,000-dimensional pair feature vector. This image pair feature is extracted from the training folds to train a same-vs-different classifier that makes the verification decision.

We obtain an accuracy of 93.1%, with a standard deviation of 0.40% across the ten folds using the gradhist variant, or $92.8\% \pm 0.47\%$ using HOG. Our method shares a great deal with the method of Chapter 3, and obtains very similar results. The most important difference is that POOFs can be applied generally to any domain, while the Tom-vs-Pete classifiers are based on support regions carefully chosen based on our experience with face recognition. POOFs are also more efficient at test time, using a linear rather than an RBF kernel. Our ROC curve is shown in Figure 4.4a, with the four best previously published results on this benchmark. Figure 4.4b compares the result from the POOFs with a result using the base features alone, showing, as in Figure 4.2 for bird species recognition, a substantial boost due to the POOFs.

4.2.3 Attribute Classification

Our third experiment is attribute classification on human faces. For their work on attributes, [Kumar *et al.*, 2011] downloaded face images from the Internet, labeled them with attributes such as gender, race, age, and hair color, and used these labels to train attribute

Attribute	Method	Number of training samples					Kumar <i>et al.</i>	Attribute	Method	Number of training samples					Kumar <i>et al.</i>					
		6	20	60	200	600				6	20	60	200	600						
Gender	low-level	50.7	61.0	66.9	81.4	87.8	90.5	No	low-level	51.2	56.6	58.8	75.6	79.5	83.9					
	POOFs	86.2	89.9	89.7	91.3	91.7		Eyewear	POOFs	65.9	76.9	75.9	85.6	87.0						
Asian	low-level	53.9	53.9	68.4	78.2	83.2	86.5	Eyeglasses	low-level	51.7	53.9	61.5	71.4	79.4	86.4					
	POOFs	75.2	75.8	84.3	87.6	89.8			POOFs	74.5	79.3	77.2	85.6	89.5						
White	low-level	57.0	57.4	68.3	76.7	77.7	85.5	Mustache	low-level	53.3	61.1	69.0	75.2	81.9	83.1					
	POOFs	66.3	74.9	82.6	81.7	80.5			POOFs	70.0	82.0	73.7	81.7	85.8						
Black	low-level	60.9	68.3	76.7	84.1	87.3	75.4	Receding	low-level	55.0	56.3	67.0	70.0	73.6	75.7					
	POOFs	74.0	84.2	87.4	88.9	90.4		Hairline	POOFs	63.7	66.4	69.3	70.5	71.8						
Youth	low-level	53.6	56.0	59.8	62.5	66.2	66.1	Bushy	low-level	49.9	55.8	63.5	67.4	72.1	71.7					
	POOFs	71.0	62.0	67.6	67.7	70.8		Eyebrows	POOFs	60.0	61.8	66.0	67.7	73.5						
Middle Aged	low-level	49.5	51.0	49.6	53.2	56.0	54.2	Arched Eyebrows	low-level	53.2	51.1	54.6	63.3	65.9	66.4					
	POOFs	47.1	50.9	51.4	57.5	59.6			POOFs	64.5	66.9	63.5	69.1	70.9						
Senior	low-level	54.6	60.6	63.7	72.1	74.3	69.5	Big Nose	low-level	52.5	52.5	59.0	63.3	66.6	65.4					
	POOFs	70.7	75.9	73.6	80.0	79.5			POOFs	55.2	63.6	61.5	64.9	68.3						
Black Hair	low-level	50.3	53.6	62.3	67.9	68.9	66.0	No Beard	low-level	57.1	51.2	62.8	71.2	75.9	80.6					
	POOFs	54.6	59.3	62.9	67.9	66.7		POOFs	71.1	68.0	68.8	68.7	76.7							
Blond Hair	low-level	53.7	60.7	69.0	72.3	74.6	67.6	Round Jaw	low-level	50.8	49.5	50.0	53.2	55.7	50.5					
	POOFs	70.5	68.8	72.6	71.4	75.2			POOFs	51.5	53.7	54.4	55.6	54.8						
Bald	low-level	54.4	57.3	65.4	68.7	70.9	71.8	Average improvement							12.3	13.4	8.0	4.3	2.7	2.8
	POOFs	55.4	62.2	64.9	66.3	66.9														

Table 4.1: Attribute classification accuracy. For each attribute, the top row is baseline accuracy using the low-level base features (color and gradient direction histograms) directly, and the bottom row is accuracy using POOFs. The more accurate is bold. The last column gives accuracy of [Kumar *et al.*, 2011] on the same test images, in bold when better than the POOF 600-sample classifier. The last row shows the average improvement using POOFs over the low-level features or [Kumar *et al.*, 2011]. As these are binary attributes, chance gives 50% accuracy.

classifiers based on low-level features such as raw pixel color and gradients. We use this same dataset to train a set of attribute classifiers based on POOFs. Kumar *et al.* have made available both human labels and the results of their attribute classifiers for 19 binary attributes on the 7701 images in View 2 of LFW. Restricting ourselves to these 19 attributes, we use these images as our test set.

Although the classes in this task (attributes) are of a different type from those in the previous experiment (identities), we remain in the face domain, and so expect the POOFs we learned there to be useful here. We use the POOFs learned in Section 4.2.2 without modification. (This means they are trained using our reference set, not the attributes-labeled images.) To build attribute classifiers, we simply extract our 10,000 POOF scores from the attribute training images, and use these feature vectors to train a linear SVM for each attribute. One of the benefits of POOFs is that by incorporating knowledge of the domain learned from the reference set, which is not labeled with attributes, they reduce the need for a large attribute-labeled training set. To demonstrate this, we restrict the number of images we use from the training set.

The results on the test set are shown in Table 4.1, using the gradhist variant of the gradient orientation base feature. As before, we also show the performance of classifiers built directly on the low-level base features. In almost every case our POOFs outperform the classifier operating directly on the low-level features. The difference is especially large when the amount of training data is small. At six training samples, many of the direct classifiers are at chance accuracy (*e.g.* gender) or even worse; it is easy for the classifier to attach significance to a random peculiarity of the six images it sees. Our POOFs, based on what they have learned is discriminative in a different set of classes (identities) in the same domain (faces), avoid this noise. The table also shows the results of the classifiers of [Kumar *et al.*, 2011] on this dataset. These classifiers are trained on between 1500 and 5600 samples each. To account for biases in the dataset, the accuracies we report are the means of the accuracies on positive and negative test images. (For example, the test set is 6% Asian, so a direct calculation of accuracy would give a “never-Asian” classifier 94%

accuracy, but our calculation would give it 50%.)

Chapter 5

How Do You Tell a Blackbird from a Crow?

How do you tell a blackbird from a crow? To answer this question, we may consult a guidebook (*e.g.*, [Sibley, 2000; Svensson *et al.*, 2011]). The best of these guides, products of great expertise and effort, include multiple drawings or paintings (in different poses and plumages) of each species, text descriptions of key features, and notes on behavior, range, and voice.

In this chapter, we consider the problem not of performing fine-grained categorization by computer, but of using computer vision techniques to show a human how to perform the categorization. We do this by learning which classes appear similar, discovering features that discriminate between similar classes, and illustrating these features with a series of carefully chosen sample images annotated to indicate the relevant features. We can assemble these visualizations into an automatically-generated digital field guide to birds, showing which species are similar and what a birder should look for to tell them apart. Example figures from a page we have generated for such a guide are shown in Figure 5.1.

In addition to the visualizations in these figures, we borrow a technique from phylogenetics, the study of the evolutionary history of species, to generate a tree of visual similarity. Arranged in a wheel, as shown in Figure 5.2, this tree is suitable as a browsing interface

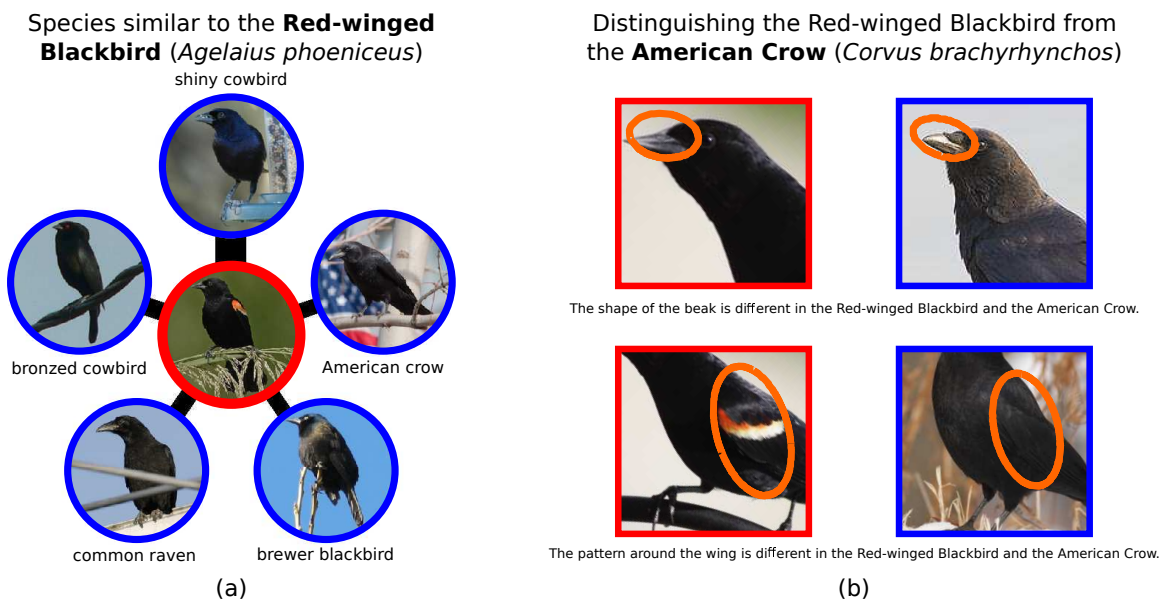


Figure 5.1: (a) For any bird species (here the red-winged blackbird, at center), we display the other species with most similar appearance. More similar species are shown with wider spokes. (b) For each similar species (here the American crow), we generate a “visual field guide” page highlighting differences between the species.

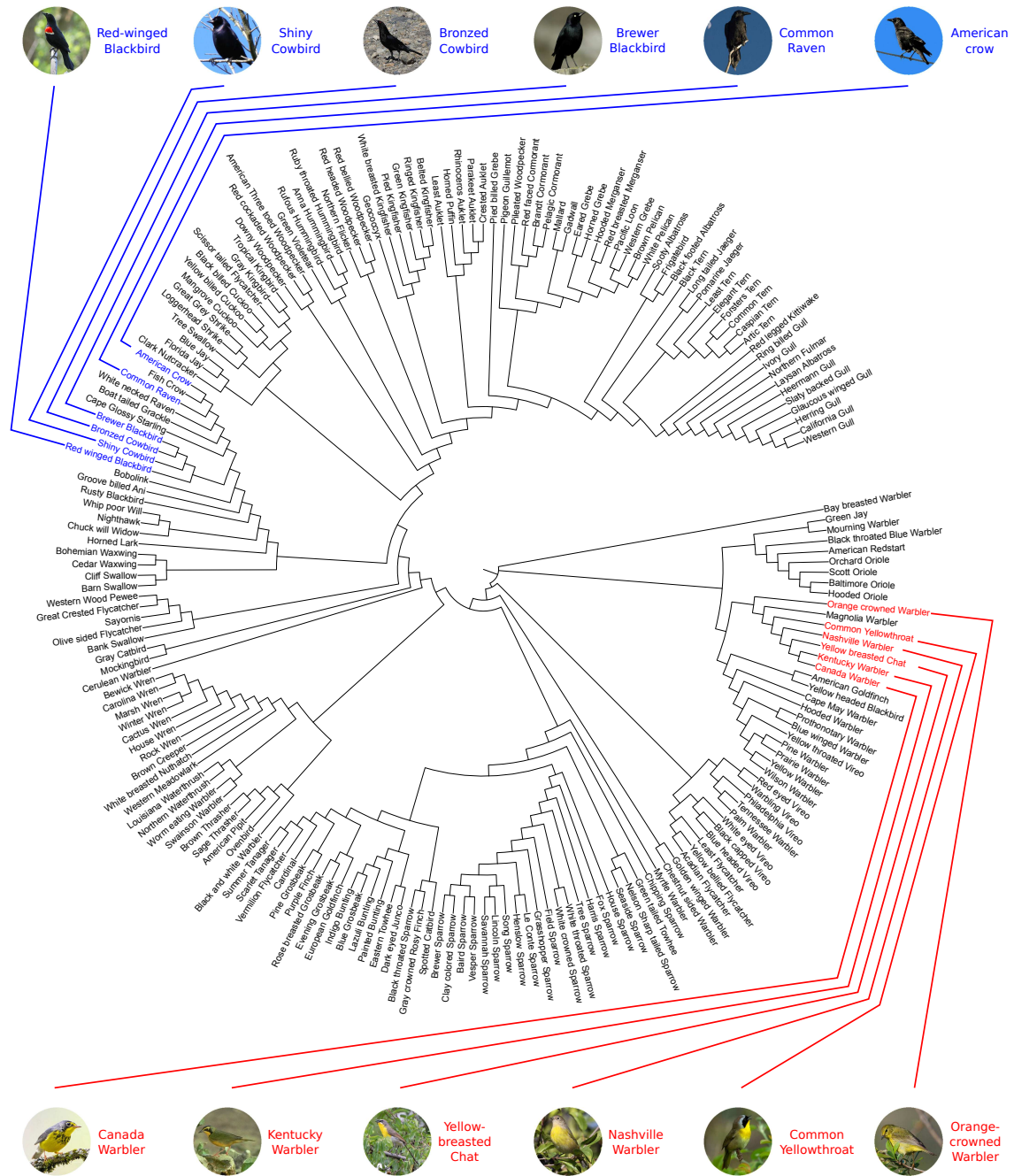


Figure 5.2: A similarity tree of bird species, built from our visual similarity matrix. Species similar to the red-winged blackbird are in blue, and species similar to the Kentucky warbler are in red.

for the field guide, allowing a user to quickly see each species and other species that are visually similar to it. We compare our similarity-based tree with the phylogenetic “tree of life,” in which branches are speciation events. Places where the trees are not in agreement – pairs of species that are close in the similarity tree but far in the evolutionary tree – are of special interest, examples of *convergent evolution* [Futuyma, 1997], where similar traits arise independently in species that are not closely related.

We base our similarity calculations on the POOFs of Chapter 4 for three reasons. First, POOFs have shown strong performance on fine-grained categorization, and in particular on bird species recognition. Second, POOFs are based on differences in part appearance, which is consistent with how people distinguish between subordinate categories, so we hope the features will be meaningful to humans. And third, POOFs are easy to illustrate. Each POOF has a small, learned support region that can be highlighted in a diagram, as shown in the examples in Figure 5.1b.

In this chapter we make the following contributions:

1. We propose and explore a new problem: using computer vision techniques, in particular methods of fine-grained visual categorization, to illustrate the differences between similar classes.
2. We propose an approach to this problem. We demonstrate a fully automatic method for choosing, from a large set of part-based features, those that best show the difference between two similar classes, choosing sample images that exemplify the difference, and annotating the images to show the distinguishing feature.
3. We explore the relation between visual similarity and phylogeny in bird species.

5.1 Related Work

In this chapter, our goal is not classification itself, but an understanding of what features are most relevant and understandable. A similar task is set by Doersch *et al.* [Doersch *et al.*,

2012], who discover the architectural features best suited to recognizing the city in which a street scene was photographed. With a much smaller dataset and a much larger number of classes, we take a careful approach based on labeled parts rather than their random image patches. Shrivastava *et al.* [Shrivastava *et al.*, 2011] weight regions in an image by their distinctiveness for purposes of cross-domain similar image search. This is similar to our method for finding regions to annotate in our illustrative images, but they work with a single image to find its *distinctive* regions, while we work with two classes of image to find the most *discriminative* regions. Both [Doersch *et al.*, 2012] and [Shrivastava *et al.*, 2011] deal with image rather than object classification, so use unaligned image patches rather than our part-based features. Deng *et al.* [Deng *et al.*, 2013] found discriminative regions in bird by explicit human labeling in the guise of a game.

Although we take a part-based approach to allow us to annotate our images, there is also non-part-based work that attempts to describe the features of a class. Parikh and Grauman [Parikh and Grauman, 2011] discover discriminative image attributes and ask users to name them. Yanai and Barnard [Yanai and Barnard, 2005] consider the opposite problem, starting with a named concept and learning whether or not it is a visual attribute, while Berg *et al.* [Berg *et al.*, 2010] discover attribute names and classifiers from web data. This could be used to provide supplementary, non-part-based text descriptions of species differences in our visual field guide.

5.2 Visual Similarity

Our goal is, in a set of visually similar classes, to determine which classes are most similar to each other, and among those most similar classes, to understand and visualize what it is that still distinguishes them. To make this concrete, we consider the problem of bird species recognition, using the Caltech-UCSD Birds 200-2011 dataset (CUBS-200) [Wah *et al.*, 2011b], described in Section 4.2.1.

The first step is to construct a vocabulary of features suitable for differentiating among

classes in our domain. POOFs are suited to our task for two reasons. First of all, they have been shown to be effective at fine-grained categorization. Second, and of special importance to us, POOFs are relatively easy to interpret. If we discover that two bird species are well-separated by a color histogram-based POOF aligned by the beak and the back, and the SVM weights are large at the grid cells around the beak, we can interpret this as “These two species are differentiated by the color of the beak.” This kind of understanding is our goal.

5.2.1 Finding Similar Classes

Few would confuse a cardinal and a pelican. It would be difficult and not useful to describe the particular features that distinguish them; any feature you care to look at will suffice. The interesting problem is to find what details distinguish classes of similar appearance. To do this we must first determine which classes are similar to each other.

Our starting point is our vocabulary of POOFs. We use the set of 5000 POOFs from Section 4.2.1, so each image is described by a 5000-dimensional vector. An L1 or L2 distance-based similarity in this space is appealing for its simplicity, but considers all features to be equally important, which is unlikely as POOFs are based on random classes and parts. We wish to downweight features that are not discriminative, and emphasize those that are. A standard tool for this is linear discriminant analysis (LDA) [Fisher, 1936], which, from a labeled set of samples with n classes, learns a projection to an $n - 1$ dimensional space that minimizes the ratio of within-class variance to between-class variance. We apply LDA, and use the negative L1 distance in the resulting 199-dimensional space as a similarity measure.

By applying this image similarity measure to mean feature vectors over all the images in a class, we obtain a similarity measure between classes, with which we can determine the most similar class to any given class. The red-winged blackbird and its five most similar species are shown at the top of Figure 5.1.

5.2.2 Choosing Discriminative Features

Given a pair of very similar classes, we are now interested in discovering what features can be used to tell them apart. We consider as candidates all POOFs that are based on this pair of classes. With the birds dataset, with twelve parts and two low-level features, there are 264 candidate features. We rank the features by their *discriminativeness*, defining the discriminativeness of feature f as

$$d_f = \frac{(\mu_2 - \mu_1)^2}{\sigma_1 \sigma_2}, \quad (5.1)$$

where μ_1 and μ_2 are the mean feature values for the two classes, and σ_1 and σ_2 are the corresponding standard deviations. Maximizing discriminativeness is similar in spirit to the optimization performed by LDA, which maximizes interclass variation and minimizes intraclass variation. Here we seek a individual score for each feature rather than a projection of the feature space, as it allows us to report particular features as “most discriminative.”

5.2.3 Visualizing the Features

Once we have determined which features are most useful to distinguish between a pair of classes, we would like to present this information in a format that will help a viewer understand what he should look for. We present each feature as a pair of illustrative images, one from each species, with the region of interest indicated in the two images.

The first step is to choose the illustrative images. In doing this, we have several goals:

1. The images should exemplify the difference they are intended to illustrate. If the feature is beak color, where one class has a yellow beak and the other gray, then the images must have the beak clearly visible, with the beak distinctly yellow in one and gray in the other.
2. The images should minimize differences other than the one they are intended to illustrate. If the yellow and gray-beaked species above can both be either brown or

black, it is misleading to show one brown and one black, as this difference does not distinguish the classes.

3. To facilitate direct comparison of the feature, the two samples should have their parts in similar configurations, *i.e.*, the birds should be in the same pose.

We translate these three goals directly into three objective expressions to be minimized. For the first, we take the view that the images should be somewhat farther from the decision boundary than average for their class, but not too far. This corresponds to the feature being somewhat exaggerated, but avoids extreme values from the POOF which may be outliers or particularly unusual in some way. Taking c_1 and c_2 as the classes associated with positive and negative feature values respectively, let b_1 be the 75th percentile of feature values on c_1 , and let b_2 be the 25th percentile of feature values on c_2 . We take these exaggerated, but not extreme feature values as “best,” and attempt to minimize

$$F(I_1, I_2) = (1 + |f(I_1) - b_1|)(1 + |f(I_2) - b_2|), \quad (5.2)$$

where I_1 and I_2 are the candidate illustrative images from classes c_1 and c_2 , and $f()$ is the feature to be illustrated.

To achieve the second goal, we consider an additional set of features, based on POOFs trained on classes other than c_1 and c_2 . We use the 5000 POOFs used to determine inter-class similarity in Section 5.2.1, less those with the same feature part as the POOF to be illustrated, and attempt to minimize the L1 distance between the resulting “other feature” vectors $\mathbf{g}(I_1)$ and $\mathbf{g}(I_2)$.

$$G(I_1, I_2) = \|\mathbf{g}(I_1) - \mathbf{g}(I_2)\|_1 \quad (5.3)$$

To achieve the third goal, we consider the part locations in the two images. We resize the images so that in each, the mean squared distance between parts is 1, then find the best fit similarity transformation from the scaled locations \mathbf{x}_1 in image I_1 to the scaled locations \mathbf{x}_2 in image I_2 . We minimize the squared error of the transformation, which we denote $H(I_1, I_2)$. Overall, we choose the image pair that minimizes

$$k_F F(I_1, I_2) + k_G G(I_1, I_2) + k_H H(I_1, I_2), \quad (5.4)$$

where coefficients k_F , k_G , and k_H determine the importance of each objective. To make them equally important, we set each to the multiplicative inverse of the standard deviation of its term, *i.e.*, $k_F = \frac{1}{\sigma_F}$, $k_G = \frac{1}{\sigma_G}$, and $k_H = \frac{1}{\sigma_H}$.

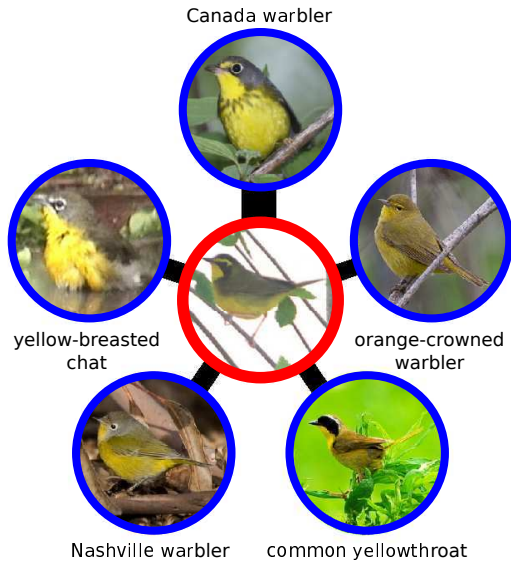
The second step in visualizing the features is annotating the chosen images to indicate the feature in question. Recall that the feature is the output of a POOF, which at its core is a vector of weights to be applied to a base feature extracted over a spatial grid. By taking the norm of the sub-vector of weights corresponding to each grid cell, we obtain a measure of the importance of each cell. An ellipse fit to the grid cells with weight above a small threshold then illustrates the feature.

5.3 A Visual Field Guide to Birds

As a direct application of the techniques in Section 5.2, we can construct a visual field guide to birds. While this guide will not have the notes on habitat and behavior of a traditional guide, it will have a couple advantages. First, it is automatically generated, and so could easily be built for another domain where guides may not be available. Second, it can be in some sense more comprehensive. While a traditional, hand-assembled guide will have an entry for each species, it is not combinatorially feasible to produce an entry on the differences between every *pair* of species. For an automatically-generated, digital guide, this is not an issue.

We envision our field guide with a main entry for each species. Examples are shown in Figures 5.1 (a) and 5.3 (a). The main entry shows the species in question, and the top k most similar other species (we use $k = 5$) as determined by the method of Section 5.2.1. Selecting one of the similar species will lead to a pair entry illustrating the differences between the two species as described in Sections 5.2.2 and 5.2.3. Figures 5.1 (b) and 5.3 (b) and (c) show examples of pair entries. We find that many of the highlighted features, including the dark auriculars (feather below and behind the eye) of the Kentucky warbler, the black “necklace” of the Canada warbler, and the white “spectacles” of the yellow-

Species similar to the **Kentucky Warbler**
(*Oporornis formosus*)



(a)

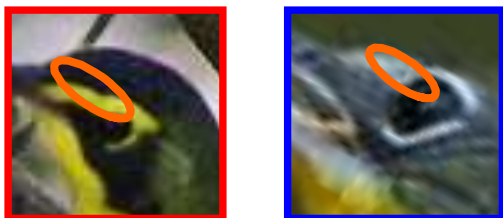
Distinguishing the Kentucky Warbler from the **Yellow-breasted Chat** (*Icteria virens*)



Kentucky Warbler Yellow-breasted Chat
The Kentucky Warbler and the Yellow-breasted Chat can be differentiated by the features illustrated below.



The color around the eye is different in the Kentucky Warbler and the Yellow-breasted Chat.



The color around the forehead is different in the Kentucky Warbler and the Yellow-breasted Chat.

(c)

Distinguishing the Kentucky Warbler from the **Canada Warbler** (*Wilsonia canadensis*)



Kentucky Warbler Canada Warbler
The Kentucky Warbler and the Canada Warbler can be differentiated by the features illustrated below.



The pattern around the beak is different in the Kentucky Warbler and the Canada Warbler.



The pattern around the forehead is different in the Kentucky Warbler and the Canada Warbler.



The pattern around the breast is different in the Kentucky Warbler and the Canada Warbler.

(b)

(a) **Species display:** For any species, we can display the most similar other species. The most similar species are displayed surrounding the species in question, with the thickness of the spokes proportional to the visual difference between species. The Kentucky warbler is most similar to the Canada warbler.

(b) **Species pair display:** After choosing one of the spokes, we display sample images of the two species, followed by a few pairs of images chosen and annotated to illustrate key visual differences. The Canada warbler is distinguished from the Kentucky warbler the curved of the yellow band by the eye, a complete eye-ring, and a black necklace. (c) The next most similar species, the yellow-breasted chat, is distinguished by the color of its eye band. We may show any number of sample images (here we fill the figure), but in general three pairs of images is sufficient.

Figure 5.3: Visual field guide pages for the Kentucky warbler.

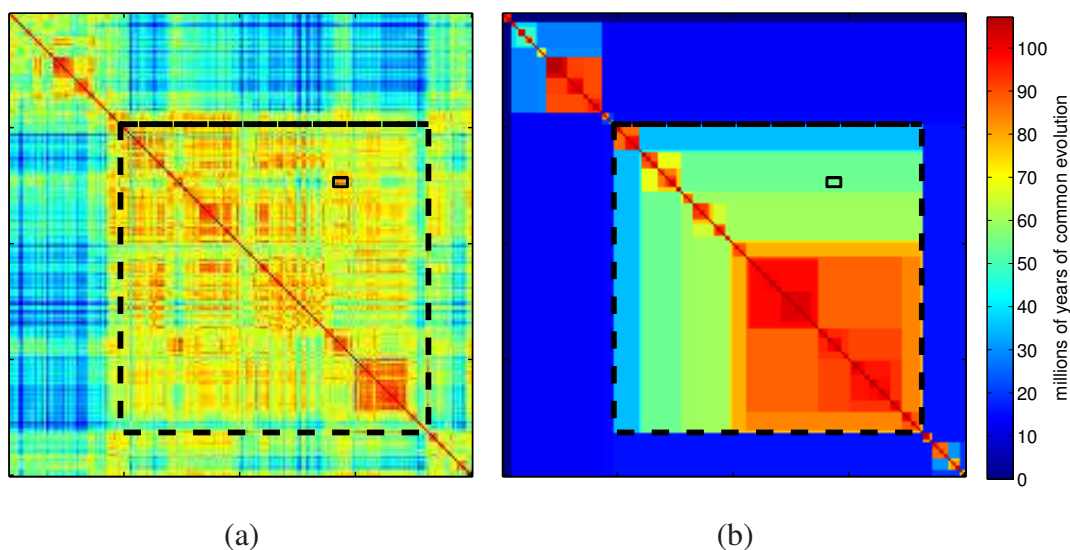


Figure 5.5: Similarity matrices. (a) Visual similarity. (b) Phylogenetic similarity. In both, rows/columns are in order of a depth-first traversal of the evolutionary tree, ensuring a clear structure in (b). The large dashed black box corresponds to the passerine birds (“perching birds,” mostly songbirds), while the small solid black box holds similarities between crows and ravens on the y-axis and blackbirds and cowbirds on the x-axis.

breasted chat (all shown in Figure 5.3), correspond to features mentioned in bird guides (all included in the Sibley Guide [Sibley, 2000]).

5.3.1 A Tree of Visual Similarity

Visual similarity as estimated from the POOFs is the basis for our visual field guide. In similarity estimation, unlike straight classification, there is no obvious ground truth. If we say a blackbird is more like a crow than like a raven, who can say we are wrong? One way to get a ground truth for similarity is to consider the evolutionary “tree of life,” the tree with a root representing the origin of life, a leaf for every extant species or evolutionary dead end, and a branch for every speciation event, with edge lengths representing time between speciations. Species close to each other in the tree of life are in a sense “more similar” than species that are not close, although this will not necessarily correspond to visual similarity.

Rank	Species Pair
1	Gadwall vs Pacific Loon
2	Hooded Merganser vs Pigeon Guillemot
6	Red-breasted Merganser vs Eared Grebe
11	Least Auklet vs Whip poor Will
16	Black billed Cuckoo vs Mockingbird
17	Black Tern vs Belted Kingfisher
19	Groove-billed Ani vs Shiny Cowbird
22	Mallard vs Rhinoceros Auklet
35	Mangrove Cuckoo vs Great Grey Shrike
46	Yellow-billed Cuckoo vs Scissor-tailed Flycatcher

Table 5.1: Species pairs with high visual and low phylogenetic similarity.

The scientific community has not reached consensus on the complete structure of the tree of life, or even the subtree containing just the birds in CUBS-200. However there is progress in that direction. Recently Jetz *et al.* [Jetz *et al.*, 2012] proposed the first complete tree of life for all 9993 extant bird species, complete with estimated dates for all splits, based on a combination of the fossil evidence, morphology, and genetic data. Pruning this tree to include only the species in CUBS-200 yields the tree shown in Figure 5.4 (produced in part with code from [Letunic and Bork, 2007]). This tree shows the overall phylogenetic similarity relations between bird species.

As a browsing interface to our digital field guide, we propose a similar tree, in the same circular format. This tree, however, is based on visual similarity rather than phylogenetic similarity. Producing a tree from a similarity matrix is a basic operation in the study of phylogeny, for which standard methods exist (note the tree of life in Figure 5.4 is based on more advanced techniques that use additional data beyond a similarity matrix). We calculate the full similarity matrix of the bird species using the POOFs, then apply one of these standard methods, Saitou and Nei’s “neighbor-joining” [Saitou and Nei, 1987], to



Figure 5.6: The top three visually similar, phylogenetically dissimilar species pairs from Table 5.1. First row: Gadwall and Pacific Loon. Second row: Hooded Merganser and Pigeon Guillemot. Third row: Red-breasted Merganser and Eared Grebe. Example images are chosen for similar pose.

get a tree based not on evolutionary history but on visual similarity. This tree is shown in Figure 5.2. In an interactive form, it will allow a user to scroll through the birds in an order that respects similarity and shows a hierarchy of groups of similar birds.

We can compare the similarity-based tree in Figure 5.2 with the evolutionary tree in Figure 5.4. They generally agree as to which species are similar, but there are exceptions. For example, crows are close to blackbirds in the similarity tree, but the evolutionary tree shows that they are not closely related. Such cases may be examples of convergent evolution, in which two species independently develop similar traits.

We can find such species pairs, with high visual similarity and low phylogenetic similarity, in a systematic way. The phylogenetic similarity between two species can be quantified as the length of shared evolutionary history, *i.e.*, the path length, in years, from the root of the evolutionary tree to the species' most recent common ancestor (techniques such as the neighbor-joining algorithm [Saitou and Nei, 1987] also use this as a similarity measure). Figure 5.5 (a) shows a similarity matrix calculated in this way for the 200 bird species, with the corresponding matrix based on visual similarity as Figure 5.5 (b). Potential examples of convergent evolution correspond to high values in (a) and relatively low values in (b). The blackbirds-crows region is marked as an example.

We rank all $\binom{200}{2}$ species pairs by visual similarity (most similar first) and by phylogenetic difference (least similar first). We then list all species pairs in order of the sum of these ranks. Table 5.1 shows the top ten pairs, excluding pairs where one of the species has already appeared on the list to avoid excessive repetition (as the pacific loon scores highly when paired with the gadwall, it will also score highly with all near relatives of the gadwall with similar appearance). The top ranked pair is a duck and a loon, two species this amateur birder had mistakenly assumed were closely related based on their visual similarity. Figure 5.6 shows samples of the top species pairs in this ranking.

Chapter 6

Birdsnap

As a demonstration of the effectiveness of POOFs for fine-grained visual categorization (Chapter 4) and visualizing the key differences between similar subcategories (Chapter 5), we have built *Birdsnap*, a digital field guide to North American birds, available online at <http://birdsnap.com> and as an iPhone app in the Apple App Store. It is a complete working system, with photos, text descriptions, and audio recordings of five hundred species, a browsing interface based on visual similarity, search filters based on date and location, illustrations of differences between species with similar appearance, and a visual recognition component that identifies birds in uploaded photos. Figure 6.1 shows the home page of the web site, and Figure 6.2 shows the main screen of the iPhone app.

The large amount of recent work on fine-grained recognition of birds has been spurred in part by the availability of the CUB-200-2011 dataset. Unfortunately this dataset includes species from many parts of the world but does not provide coverage of all or most species in any one part of the world, so it cannot be used to produce a useful field guide. In addition, some of the classes in CUB-200-2011 do not correspond exactly to species (Frigatebird, *Geococcyx*, and *Sayornis* are genera). To produce our guide, we therefore collected a new dataset of bird images, covering 500 of the most common species in North America. This is two-and-a-half times the number of classes in CUB-200-2011, and includes several groups of species with very high visual similarity, (e.g., genera *Sterna* (terns), *Aphelocoma*

birdsnap USA Eastern Western Backyard Local
Birds of New York, NY on May 1

Bird Wheel Bird List Bird Lab About

Sort by Tree of Life Alphabetical Frequency Visual Recognition Text Search

Order: *Passeriformes*
Family: *Hirundinidae*
Subfamily: *Hirundininae*
Genus: *Hirundo*
Species: *H. rustica*

by david m

★ - visually similar
➔ - arriving
➔ - departing
➔ - migrating through

Barn Swallow

Some recordings include other species View original image

Figure 6.1: The main, species-browsing page of the Birdsnap web site. Species can be arranged by the phylogenetic “Tree of Life” (shown), by visual similarity (as described in Section 5.3.1), by sighting frequency at the currently selected place and date (based on the spatio-temporal prior described in Section 6.3), or alphabetically.



Figure 6.2: The main screen of the Birdsnap iPhone app, a simpler version of the browsing wheel on the web site.

(scrub-jays), and *Melospiza* (song sparrows)). With this larger and more difficult dataset, and with the experience of building a practical system with real users, we found the results of the methods in Chapter 4 for automatic recognition and Chapter 5 for the illustration of differences between similar species were sometimes unsatisfactory. In this chapter, we describe the improvements to those methods that were required to build Birdsnap, introducing three ideas that mitigate complications arising from large numbers of highly similar subcategories.

The first we call “one-vs-most” classification, a replacement for the one-vs-all classification found as the last step in most fine-grained categorization pipelines, including ours from Chapter 4. One-vs-all classifiers can have particular difficulty with highly similar classes, as each one-vs-all classifier finds samples very similar to its positive class in its negative training set. We show that reducing this difficulty in the training set leads to better results in both accuracy and in apparent “reasonableness” of highly-ranked species. One-vs-most classification is described in Section 6.2.

The second is based on the observation that modern cameras embed more than image

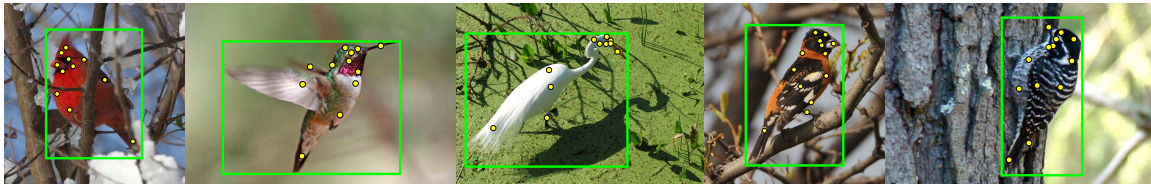


Figure 6.3: Sample images from the Birdsnap dataset, with bounding boxes and part annotations. The species of these samples, from left to right, are Northern Cardinal, Broad-tailed Hummingbird, Great Egret, Black-headed Grosbeak, and Nuttall’s Woodpecker.

data in the images they capture. In particular, almost all smartphone cameras and many recent non-phone cameras embed the time and location of image capture in the image files they produce. Biological categories in particular often have a well-studied geographic distribution, and it is wasteful not to use this information. For migratory animals, which includes most bird species, the distribution depends on time as well as location, and we show how the estimation and use of a spatio-temporal prior on sighting each species dramatically improves classification accuracy. We describe the estimation and use of this prior in Section 6.3.

The third is a small modification to the process described in Section 5.2.2 for choosing POOFs that illustrate the key differences between similar species. We discover that for highly similar species in our more difficult dataset, POOFs often have a high variance, with no POOF having a high “discriminateness” score. We get a better set of POOFs by instead ranking by accuracy on a held out set. In addition, we apply a filter to the list of illustrative POOFs to ensure we get a diverse set of features for illustration. We describe this process in Section 6.6.

6.1 The Birdsnap Dataset

Our dataset contains 49,829 images of 500 of the most common species of North American birds. There are between 69 and 100 images per species, with most species having 100. Each image is labeled with a bounding box and the location of 17 parts: the back, beak,

belly, breast, crown, forehead, nape, tail, throat, left cheek, left eye, left leg, left wing, right cheek, right eye, right leg, and right wing. Of course in most images not all 17 parts are visible; hidden parts are marked as such, with no location. Figure 6.3 shows some example images with their part annotations. A subset of the images are also labeled as male or female, adult or immature, breeding or nonbreeding plumage, or with subspecies information.

The images were found by searching for each species' scientific name on Flickr, based on the intuition that photographers who take the trouble to label their images with the scientific name are more likely to also take the trouble to ensure a correct labeling. For species for which this did not yield enough images, we ran additional searches using the common names. The images were then presented to labelers on Amazon Mechanical Turk [Amazon, 2013], with illustrations of the species from a field guide ([Sibley, 2000]), for confirmation of the species label, and to flag images with no birds or multiple birds, or non-photographs. Labelers on Mechanical Turk also marked the locations of the 17 parts. For species for which the field guide included images for different subcategories, we asked labelers to indicate these categories as well. These subcategories are most commonly sex, age category, or seasonal plumage variant (many species have different, more striking plumage in the breeding season), but in some cases are subspecies labels. All labeling jobs were presented to multiple labelers, and images with inconsistent results were discarded. Full details of the dataset and its construction in the appendix.

Our dataset is similar in structure to CUB-200-2011, but has three important advantages. First, it contains two-and-a-half times the number of species and four times the number of images. Second, it covers all the most commonly sighted birds in one part of the world (North America), which lets us build a field guide that is useful in that region. Third, it better reflects the appearance variation, especially sexual dimorphism and age-based appearance variation, of many species. For example, in the red-winged blackbird, only the male has red markings on the wing. CUB-200-2011 contains only male red-winged blackbirds, while our dataset contains a mix of males and females.

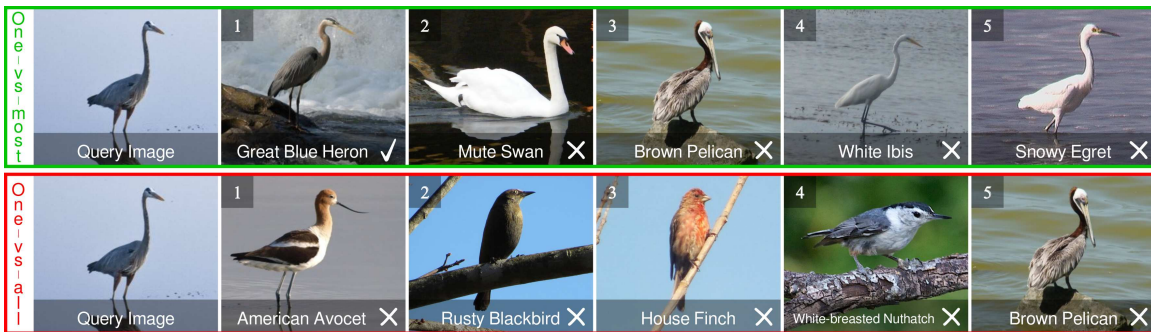
6.2 One-vs-Most Classifiers

A fundamental problem in fine-grained visual categorization is how to handle subcategories that are nearly indistinguishable. In the bird world, an example of this problem is the terns, comprising ten species across six genera in our dataset, all of very similar appearance. If we train a discriminative one-vs-all classifier in the usual way for, say, the Common Tern, that classifier will be trained based on a positive set with images of just the common tern and a negative set that includes, in addition to non-terns, images of nine different species that look very much like the positive species. A classifier in this situation is very likely to latch on to accidental features that distinguish the Common Tern from other terns only in this particular training set and de-emphasize significant features that distinguish terns from non-terns.

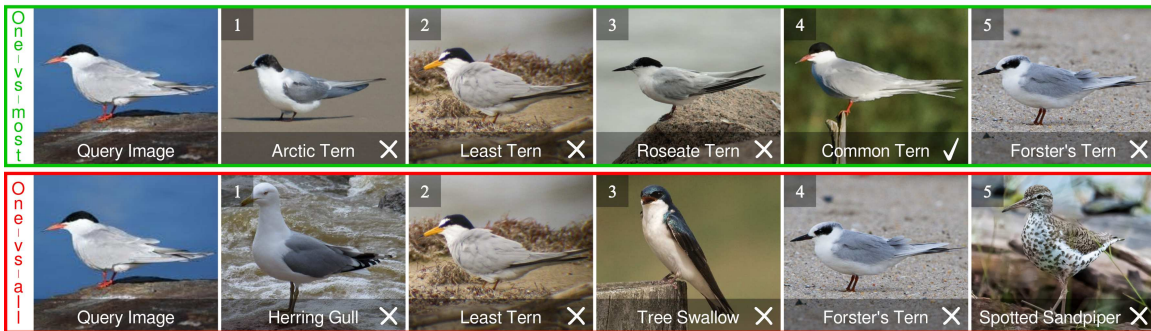
To mitigate this issue, we omit from the negative training set all images of the k species most visually similar to the positive species (we use the inter-class similarity measure described in Chapter 5). We call the resulting classifier a *one-vs-most* classifier. When the classifier omits similar terns from the negative training set, it is free to take advantage of features shared by terns (but different from other birds) as well as features that are unique to the Common Tern. Given a training set and a similarity measure, we choose the best value for k by evaluating performance on a held out set.

Note that one-vs-most classifiers can be implemented as a special case of cost-sensitive learning [Elkan, 2001], by setting the cost of misclassification as the k most similar species to zero. However, while cost-sensitive learning usually sacrifices accuracy for lower cost, we will show in Section 6.4 that one-vs-most classifiers lead to both more reasonable (lower cost) errors and a reduction in overall error rate.

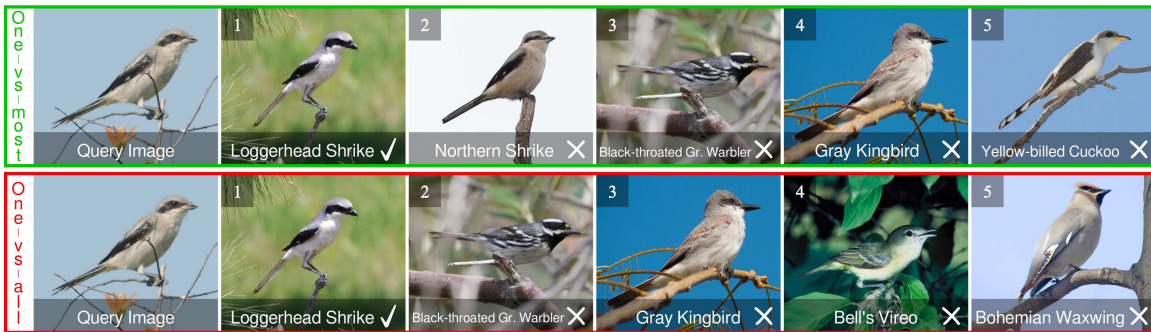
Birdsnap uses a set of one-vs-most SVMs based on POOFs. Using one-vs-most classifiers brings a significant boost to accuracy. In addition, we find a qualitative benefit: the top-ranking wrong classes are more “reasonable” – that is, they look more like the query image. In a system, like Birdsnap, that shows a ranked list of classes rather than a single guess, having the ranking appear reasonable in this way inspires confidence in the clas-



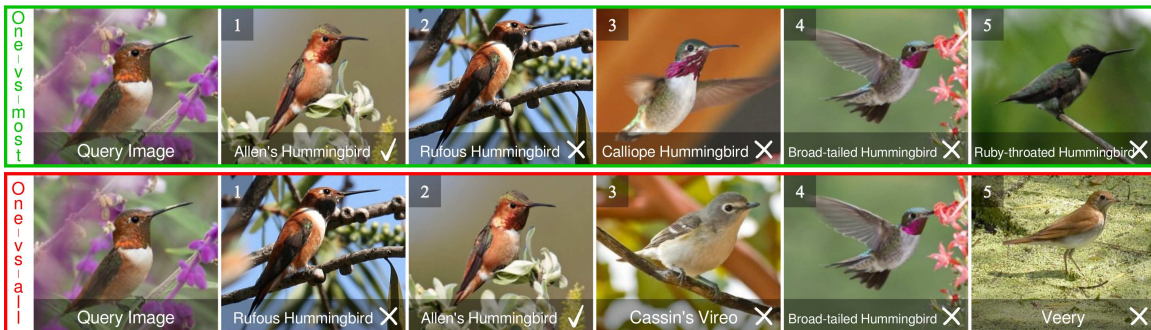
(a)



(b)



(c)



(d)

Figure 6.4: One-vs-most classifiers (top) improve both overall accuracy and the consistency and “reasonableness” of classification results.

sification result. Figure 6.4 shows the top five classes using one-vs-all and one-vs-most classifiers on four query images. For the Great Blue Heron query image in (a), the one-vs-most classifiers are correct at rank one, while the one-vs-all classifiers do not return the correct species in the top five. More than that, the one-vs-most classifiers return only long-necked water birds, while the one-vs-all classifiers include three perching birds that no human birder would consider similar to the query image. For the Common Tern query image in (b), the one-vs-most classifiers return only terns in the top five, while the one-vs-all classifiers include a swallow and a sandpiper. The sandpiper is especially unlike a tern. For the Loggerhead Shrike query image in (c), both sets of classifiers are correct at rank 1, and return white-bellied, gray-backed birds in the top five. But only the one-vs-most classifiers return the Northern Shrike (the only other shrike in the dataset), and only the one-vs-all classifiers return the distinctly different Bohemian Waxwing. And for the Allen’s Hummingbird query image in (d), the one-vs-most classifiers are correct at rank one and return only hummingbirds in the top five, while the one-vs-all classifiers get the correct species at rank two, and include two non-hummingbirds in the top five. This pattern occurs for many queries; the one-vs-all classifiers, whether or not they find the correct species, often include species that are very different from the query image. Even when the rank-1 species is correct, this is a poor user experience. Results from the one-vs-most classifiers are more consistently similar to the query image. Experiments in Section 6.4 show the advantage of one-vs-most classifiers in both accuracy (Figures 6.6 and 6.8) and consistency (Figure 6.7).

6.3 A spatio-temporal prior for bird species

Prior knowledge can improve the performance of classification systems. A spatio-temporal prior is attractive for bird species identification, because the density of bird species varies considerably across the continent and throughout the year, due to migration. We see this in Figure 6.5, where slices of our spatio-temporal prior reveal the migration pattern of the Barn Swallow.

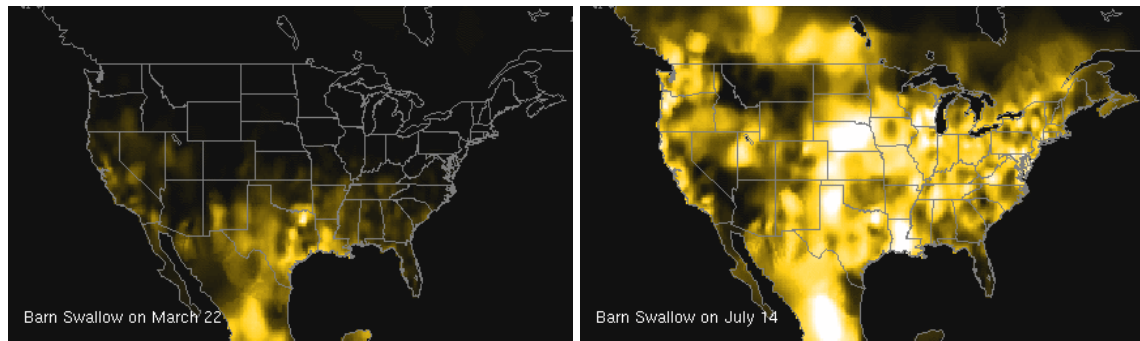


Figure 6.5: Fixed-time slices of our spatio-temporal prior show the Barn Swallow arriving from South America during its spring migration (left) and established in its summer grounds (right). Brighter regions indicate higher likelihood of a sighting.

There is previous work using spatial priors to improve vision system performance. For example, in pedestrian detection, knowledge of the ground plane and street layout can restrict a detector to regions of interest [Enzweiler and Gavrila, 2009]. However, we are not aware of any work estimating spatio-temporal priors from large-scale observations to improve classification.

In order to combine a spatio-temporal prior with classifiers, we must convert the classifier output to a probability. As suggested by [Zadrozny and Elkan, 2002] we use the method of [Platt, 1999] to produce probabilities from the output of the SVMs. This gives an estimate of $P(s|I)$ for each species s given image I , but these estimates may not be consistent with a single probability distribution. [Zadrozny and Elkan, 2002] note that simply normalizing the probabilities so that $\sum_s P(s|I) = 1$ works well in practice, and we follow this suggestion. To take advantage of the location x and date, t at which the photo was captured, we wish to find $P(s|I, x, t)$. Bayes' rule gives us

$$P(s|I, x, t) = P(I, x, t|s)P(s)/P(I, x, t). \quad (6.1)$$

We assume the image and the (location, date) pair are conditionally independent given the species¹, so this becomes

¹This is certainly false. For species with seasonal plumage the image is dependent on date, and for species

$$P(s|I, x, t) = P(I|s)P(x, t|s)P(s)/P(I, x, t). \quad (6.2)$$

Applying Bayes' rule to $P(I|s)$ and $P(x, t|s)$, we get

$$\begin{aligned} P(s|I, x, t) &= \frac{P(s|I)P(I)}{P(s)} \frac{P(s|x, t)P(x, t)}{P(s)} P(s)/P(I, x, t) \\ &\propto \frac{P(s|I)}{P(s)} P(s|x, t), \end{aligned} \quad (6.3)$$

where we have dropped all factors that do not depend on s , as they will not affect the classification decision. $\frac{P(s|I)}{P(s)}$ is the calibrated classifier score ($P(s)$ appears in the denominator because in training the classifier we first equalize the number of images for each species). $P(s|x, t)$ is the spatio-temporal prior for the species, which we estimate in the next section.

6.3.1 Adaptive kernel density estimation of the spatio-temporal prior

In this section we construct an estimate for the prior probability that a bird observed at a given location and date belongs to a particular species. We use this prior to improve recognition performance of our classifiers (Section 6.3) and create visualizations that illustrate the varying distribution of a species throughout the year, or to provide a guide to the current species that one might observe at a particular place and time.

Our prior is based on over 75 million records of North American bird sightings provided by eBird [Sullivan *et al.*, 2009]. In addition, we make use of structural knowledge that some birds migrate annually, while others may remain year-round at a given location. We combine this information by first applying a variant of adaptive kernel density estimation to densely approximate the probability density of expected bird observations throughout the year in all parts of the US. We then post-process this density for each species to

with geographically and visually distinct subspecies (for example the Dark-eyed Junco) it is also dependent on location. However our sightings dataset does not associate images with particular sightings, so we are forced to make this assumption, and find that it still provides a substantial boost to accuracy.

determine whether that species has been observed to migrate, and to determine the timing of migrations.

We wish to estimate the prior probability of a bird observation, $P(s|x, t)$, *i.e.* the probability that an observation made at time t and location x is of species s . As the density of a bird species displays much greater variation throughout the year than across different years [Fink *et al.*, 2010], we let t denote a day and month, pooling observations across years. Although we have a large volume of observational data available, direct estimation of the probability from this data is problematic, because of the uneven distribution of observations. Birding observations are concentrated near areas of high population density and/or at locations known to attract a wide variety of birds (for example, a high proportion of observations in New York City are reported from Central Park), and may occur disproportionately at certain times of year.

To deal with sparse data, we use adaptive kernel density estimation. First, we divide our problem into two parts. We estimate the density that any observation will occur at (x, t) , and we also estimate the density of observations of species s at (x, t) . $P(s|x, t)$ is then the ratio of these two densities.

We use a *balloon estimator* [Terrell and Scott, 1992]:

$$\hat{f}(y) = \frac{1}{nh(y)^d} \sum_{i=1}^n K\left(\frac{y_i - y}{h(y)}\right). \quad (6.4)$$

Here, $\hat{f}(y)$ is the estimated density at $y = (x, t)$, n is the number of samples, d is the dimension of the space, $y_i = (x_i, t_i)$ is the i th sample, K is the kernel, in our case a Gaussian, and h is the bandwidth, which depends on the location and time, y , at which we are estimating the density. As noted by [Terrell and Scott, 1992], the estimated density does not globally integrate to 1, but this is not a problem in our context, since we are taking the ratio of two estimates in which the same h is used for bandwidth. We set h , the standard deviation of the Gaussian, to half the distance to the 500th-nearest observation. We sum only over nearby observations, as distant observations contribute only small values to the sum. So we take

k	rank 1	rank 3	rank 5	rank 10
0	0.649	0.753	0.798	0.846
1	0.658	0.755	0.799	0.851
3	0.660	0.762	0.807	0.863
5	0.665	0.768	0.810	0.863
7	0.666	0.779	0.816	0.869
10	0.664	0.783	0.819	0.872
15	0.666	0.785	0.824	0.873
20	0.661	0.786	0.823	0.877
30	0.657	0.792	0.836	0.879
40	0.659	0.790	0.830	0.885
50	0.648	0.787	0.830	0.882

Table 6.1: Accuracy of the one-vs-most classifiers increases at all ranks as k increases to 15. Beyond $k = 15$, high-rank accuracy continues to increase, but rank-1 accuracy decreases.

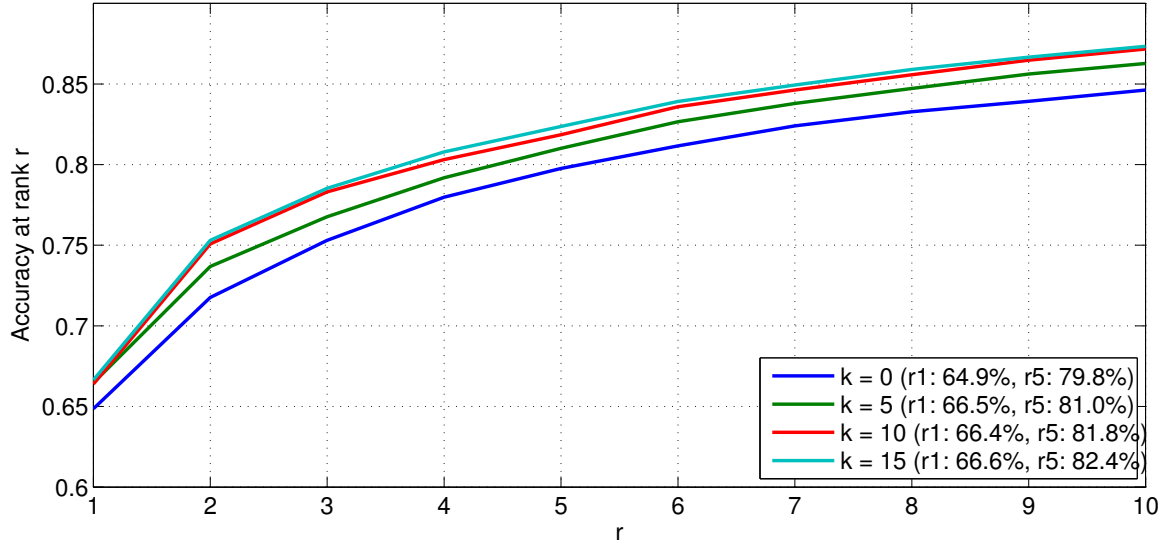


Figure 6.6: One-vs-most accuracy omitting the k most similar classes from training. As we increase k , accuracy of the one-vs-most classifiers initially increases at all ranks. Results for additional values of k , shown in Table 6.1, are omitted for clarity.

$$P(s|x, t) \approx \frac{\sum_{y_i \in N(y), s} K\left(\frac{y_i - y}{h_o(y)}\right)}{\sum_{y_i \in N(y)} K\left(\frac{y_i - y}{h_o(y)}\right)}. \quad (6.5)$$

The sum in the numerator is only over observations of species s . Note that h_o depends on all observations, not just those of species s . We take $N(y)$ to include all observations within a distance of $2h$ from y , guaranteeing that the estimate will be derived from a neighborhood containing at least 500 observations.

Even when we restrict sums to $N(y)$, this computation is potentially expensive. For this reason, we begin by discretizing all observations into spatio-temporal cubes with a spatial width of one-quarter degree of latitude/longitude and a temporal width of six days. This allows us to represent many observations with a single point, weighted by the number of observations. Distance calculations are done in units of these cubes, so a spatial distance between observations of a quarter degree is “equal” to a temporal distance of six days for purposes of kernel calculation.

The problem of building spatio-temporal models of species distribution has been previously studied in the ecology literature. [Fink *et al.*, 2010] contains a discussion of a number of prior methods, and proposes a new method in which spatially overlapping decision trees are combined to estimate the density of species observations. The input to the decision tree classifiers is a location and time, along with other meta-data about that location such as the elevation and type of land cover. Intuitively, one expects that this type of information can be useful, although [Fink *et al.*, 2010] do not compare to a model that does not use this information. Unfortunately, while interesting, their system is rather complex, and they do not describe all parameters needed to replicate their results, nor do they make an implementation available for purposes of comparison.

6.4 Experiments on the Birdsnap Dataset

To evaluate our classification system of POOFs combined with a spatio-temporal prior based on recorded sightings, we hold out a test set of 2443 images from the Birdsnap dataset—two to five per species—and train on the rest. Where images for a species include multiple images from a single Flickr account, we ensure those images are all in training or all in test, to avoid having test images of the same individual bird at the same time and place as any training image.

We learn 5000 random POOFs from the training images using the labeled part locations, then extract the POOFs for one-vs-most training using detected part locations. We use the part detector of [Liu and Belhumeur, 2013], which includes a random component, so we run it three times on each training image to augment the training set. This gives 250-285 training (image, parts) pairs per class, from which we use the 200 most accurate detections, reasoning that if the part detection fails badly, classification cannot succeed. Each one-vs-most classifier is a linear SVM trained on these 200 positive samples and 100 samples (randomly chosen from the 200) for each negative class. The extra positive samples improve the balance of the training set.

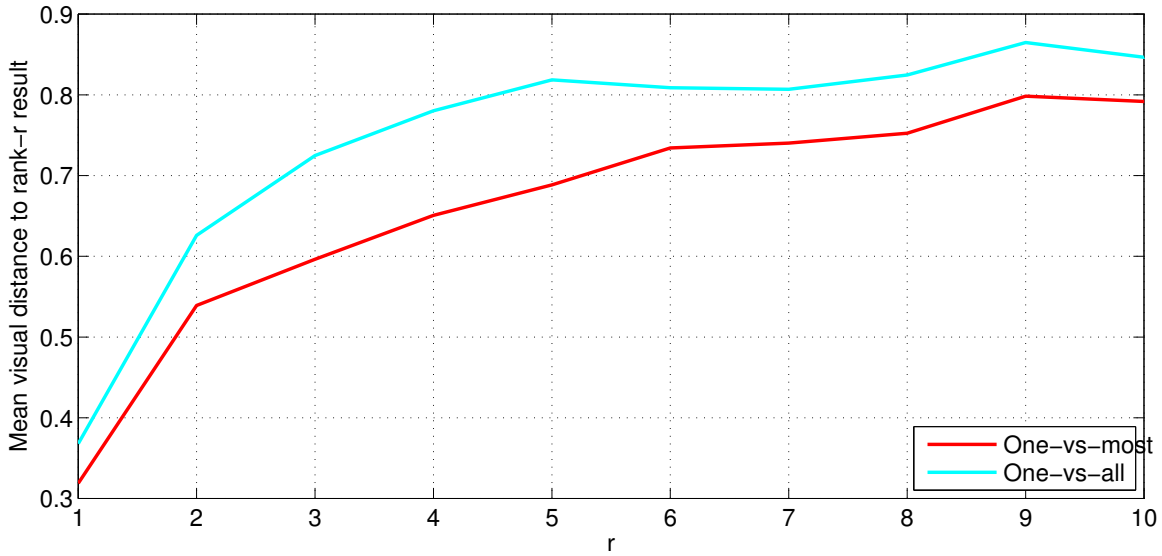


Figure 6.7: Mean visual distance between query species and returned species. One-vs-most classifiers return species that are more similar to the query species.

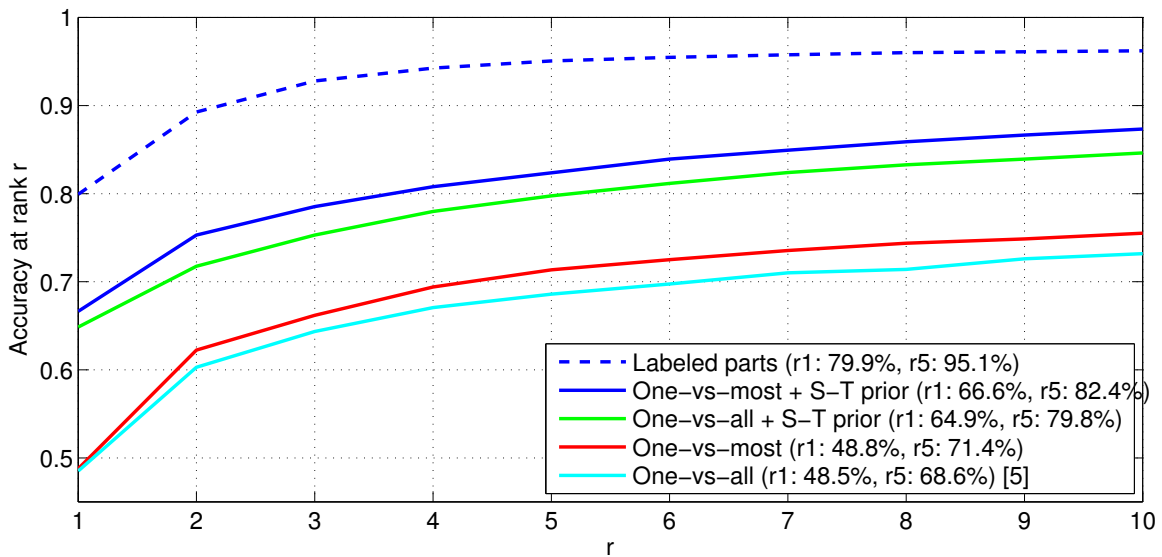


Figure 6.8: The one-vs-most classifiers and spatio-temporal prior each contributes significantly to overall performance. The dashed line, using labeled part locations, shows hypothetical performance with human-level part localization.

Many birds form flocks, and photographs often contain multiple birds—not always of the same species. To resolve this ambiguity and reduce response time in our application, we ask users to click the rough location of the head and tail, giving us an approximate bounding box. This limits the search space considered by the part detector. In experiments, we generate these click locations by randomly perturbing the true location of the eye and tail in x and y by up to an eighth of the side length of the bounding box.

As with the images, we hold out a random subset of the bird sightings for testing. The North American portion of the eBird dataset includes 6,249,584 *checklists*—lists of the birds seen by an observer on a particular outing—with a total of 76,833,202 individual bird sightings. We hold out a randomly selected ten percent of the checklists for testing, and estimate the spatio-temporal prior from the remainder.

Each submission to the identification system consists of an (image, location, date) triple. We construct a test set by first choosing a random 10,000 sightings from the held-out eBird data, yielding a set of 10,000 (species, location, date) samples. For each sample, we randomly choose an image of that species from the held-out image set. This produces a test set of 10,000 (image, location, date) triplets.

Having learned our set of 5000 POOFs, we next seek the optimal value of k for the one-vs-most classifiers, *i.e.* how many species should be left out of the negative training sets. Figure 6.6 and Table 6.1 show accuracy within the top r guesses for several values of k . We see that while rank-1 accuracy peaks at $5 \leq k \leq 15$, rank-5 accuracy increases through $k = 30$, and rank-10 through at least $k = 40$. This is expected: at higher ranks, it is less useful to distinguish between highly similar species. For Birdsnap, we choose $k = 15$, which produces a nice boost at rank 5 without sacrificing accuracy at rank 1.

Figure 6.7 demonstrates the effect seen qualitatively in Figure 6.4: that the top few species returned by the one-vs-most classifiers are more consistently similar to the query species than those returned by one-vs-all classifiers. We use the visual distance measure of Chapter 5, normalized so that the average distance between species is one, and find the mean over the test set of the distance from the species of the query image to the species returned at

rank r . As suggested by Figure 6.4 and confirmed by Figure 6.7, the species returned by our one-vs-most classifiers are more visually similar to the query species than those returned by one-vs-all classifiers.

Figure 6.8 shows the contributions of the one-vs-most classifiers and the spatio-temporal prior over the standard one-vs-all classifiers (equivalent to one-vs-most with $k = 0$) without the prior. Note that this baseline—POOF-based one-vs-all classifiers—is the method we described in Chapter 4. We see that at rank 5, the prior increases accuracy from 68.6% to 79.8%. This translates to a reduction in error rate of 35.6%, *i.e.* 35.6% of the errors of the baseline system are corrected by use of the spatio-temporal prior. Use of the one-vs-most classifiers brings rank-5 accuracy to 82.4%, an additional 12.9% reduction in error rate. Figure 6.8 also shows our system’s accuracy if we use the manually labeled part location at training and test time. With manually labeled parts we achieve 79.9% accuracy at rank 1 and 95.1% at rank 5. The large boost from using manually labeled parts suggests there is still plenty of room for improvement in part detection.

6.5 Visualizing species frequency and migration

The density estimation method described in the previous section smooths our observation data and fills in the prior in locations with few observations. Still, some noise remains. We can use structural knowledge of bird migrations to reduce this noise. For example, if we can determine that a bird has migrated away from a location in the winter, a few scattered observations can be treated as noise, and thresholded to zero. There is particular value in determining when a species is not present at a location, because we can use this knowledge to limit the species shown to a user browsing local birds. Also, we provide users with information about the timing of migration, which is of general interest.

Figure 6.9 shows the densities of three species at fixed locations over the course of a year. While most estimated densities are smooth over time, some rarely reported species, such as the Wild Turkey, have noisy densities. To smooth the noise without moving the

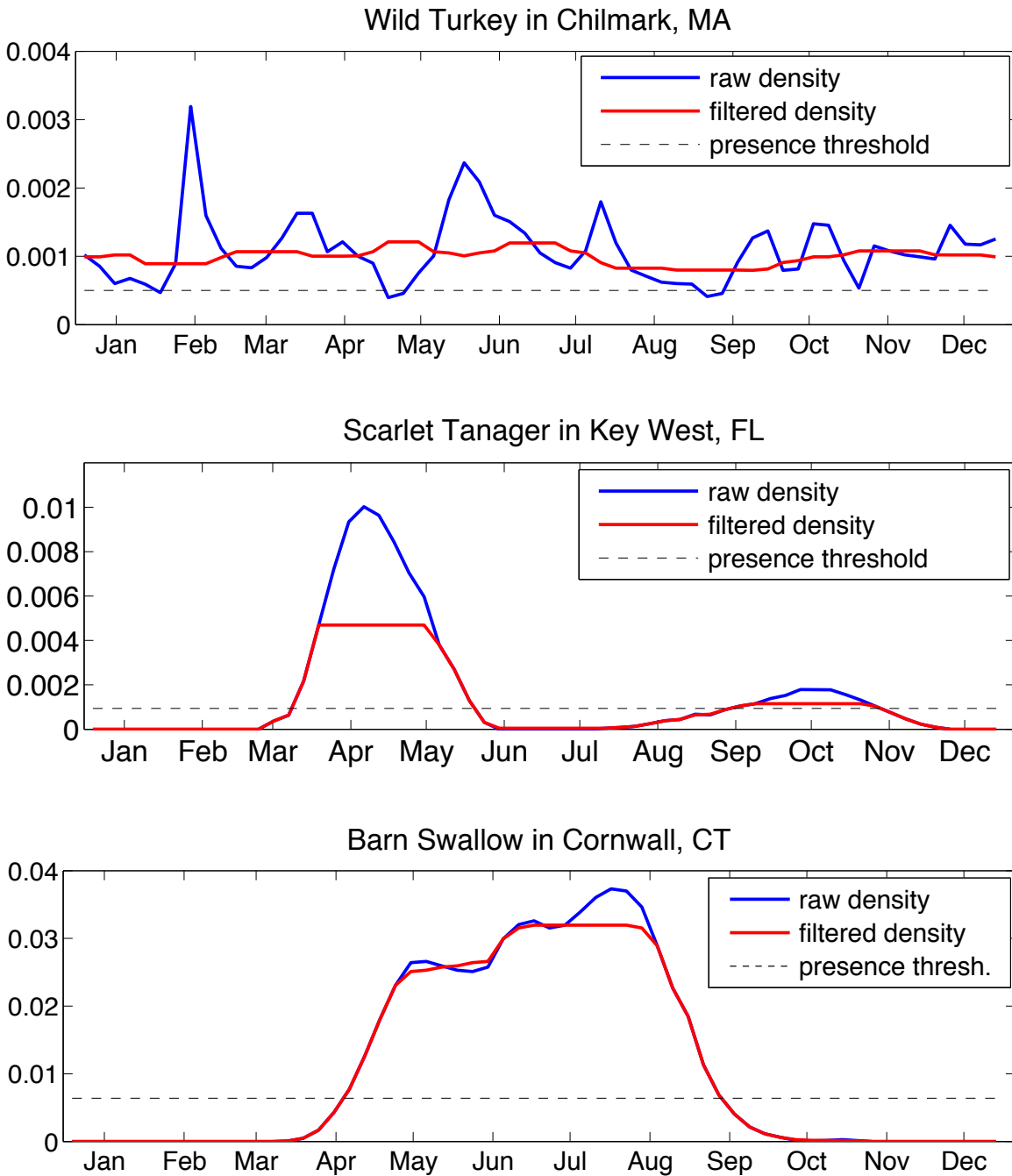


Figure 6.9: Species density over time in a fixed location. The “raw density” is the estimate from Section 6.3.1. Applying a median filter and adaptive threshold lets us recognize the Wild Turkey as present year round, despite the low frequency.

edges, where the bird transitions between presence and absence, we apply a median filter. We then apply an adaptive threshold of 20% of the peak density to determine presence and absence.

At each location, a species can exhibit one of the following patterns of presence and absence:

1. In some locations, the species is never present.
2. In some locations, the species is present year-round, *e.g.*, the Wild Turkey in Chilmark, MA,
3. In a species' summer or winter grounds, it is present for one interval, *e.g.*, the Barn Swallow in Cornwall, CT, or
4. Along a species' migration route, it is present for two intervals, *e.g.*, the Scarlet Tanager in Key West, FL.

(These example densities are shown in Figure 6.9.) The 20% threshold is chosen empirically to make most species follow these patterns. To give users a sense of the bird activity around them, we give them the option of only showing birds that are currently in their area. Birds that follow the third pattern (indicated by two transition points during the year) and are close to transition are marked as “arriving” or “departing,” while birds following the fourth pattern are marked as “migrating through.”

6.6 Illustrating field marks

A field guide is not a black box that identifies birds. Rather, through text and illustrations, it describes the distinguishing features, or *field marks*, of each species. This allows the user to justify the identification decision, and, once the field marks have been learned, to make future identifications without reference to the guide.

To achieve this in our online field guide, we create, for any pair of visually similar species, a set of images illustrating the differences between the species. In Section 5.2.3

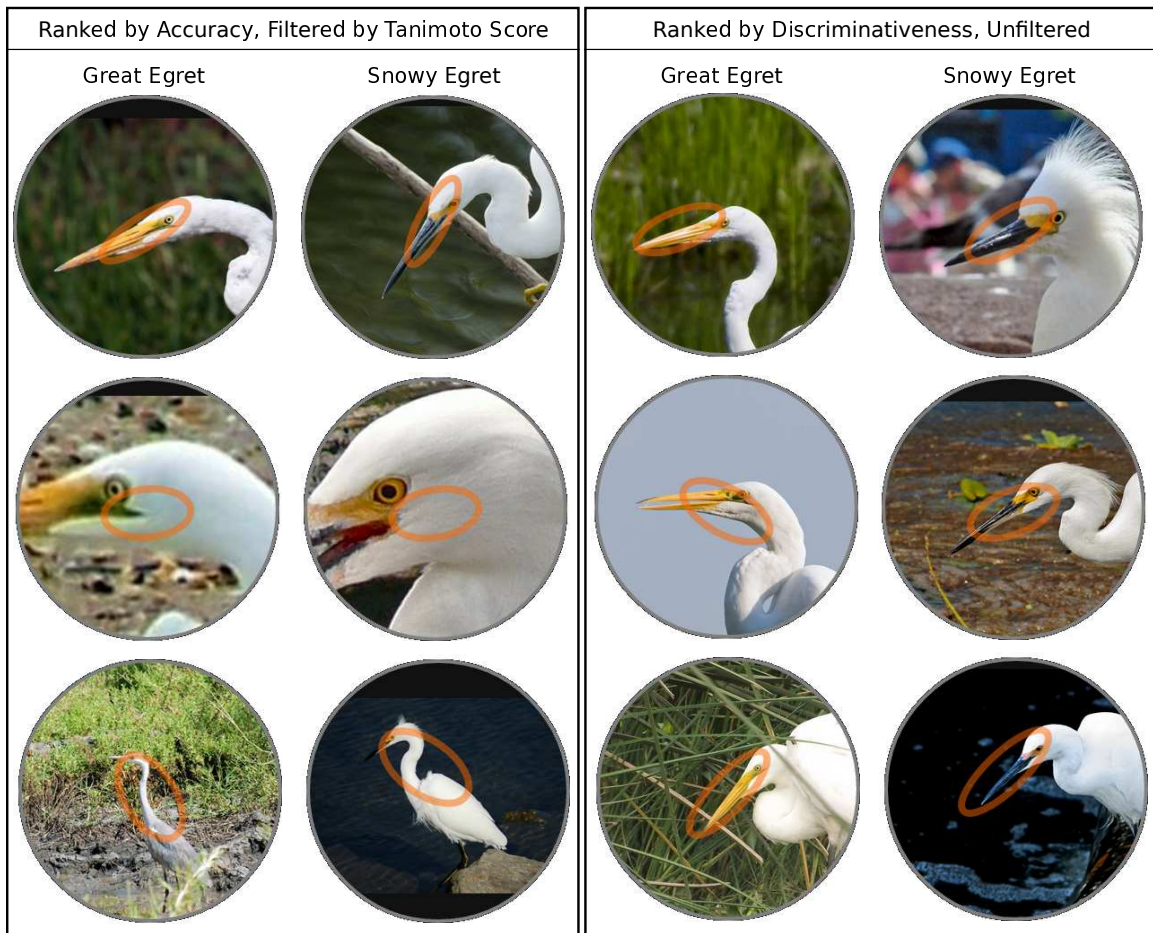


Figure 6.10: Field marks differentiating the Great Egret and the Snowy Egret. By filtering based on Tanimoto similarity, we ensure that we find three *different* features: beak color, the extension of the mouth beneath the eye, and the long, slender neck. In contrast, the top three features found by the method of Chapter 5 without filtering all appear to relate to beak color.

we have described a way to do this by ranking the POOFs trained on the species pair by a “discriminateness” score and illustrating the differences embodied by each of the top-ranking POOFs. We use as-is the method described in that section for generating illustrations from a POOF, but with the more difficult and diverse Birdsnap dataset find it better to use a simpler ranking score, classification accuracy on a held out set of images of the two species, to choose the POOFs to illustrate. By ranking POOFs on their classification accuracy, we choose POOFs for our field marks directly based on their ability to distinguish the two classes. We speculate that the discriminative score is less effective in this case due to the high degree of intraspecies variation in the dataset, which can lead to high POOF score variance and thus low discriminateness score even for POOFs that effectively discriminate at least some samples (for example, just the males – often the best one can do with very similar species).

Whether ranking by discriminateness score or classification accuracy, we find that there are frequently multiple high-ranking POOFs that produce very similar illustrations, because the illustrative ellipses for the POOFs have significant overlap. For example the ellipse for a POOF based on the beak and the crown often overlaps with one based on the beak and the forehead – and if one of these POOFs is discriminative, the other is fairly likely to be as well. To present a list of *distinct* field marks, we filter the ranked list of POOFs based on the Tanimoto similarity of the two ellipses. The Tanimoto similarity between two shapes is the ratio of the shapes’ area of intersection to area of union. We define a *Tanimoto score* for any pair of POOFs that discriminate between species s_i and s_j as the mean Tanimoto similarity between the ellipses illustrating the two POOFs, taken over the held-out images of s_i and s_j . When choosing the POOFs to illustrate the differences between two similar species, we exclude any POOF whose Tanimoto score with a higher-ranked (and not already excluded) POOF is above a threshold. We find that a threshold of 0.05 gives a clear distinction among the illustrations of the POOFs in the final list. Birdsnap displays the annotated image pairs for the top three POOFs in the filtered list. Figure 6.10 shows a comparison of field mark illustrations for the top three features chosen by the two

selection procedures.

6.7 A Tour of Birdsnap

To combine the work described in this thesis into an online guide to birds, we begin with the Birdsnap and eBird datasets, and build the recognition system, similarity tree, illustrative image sets, and spatio-temporal species distributions as described in this chapter and Chapters 4 and 5. To build a richer experience, we include additional information, such as text descriptions and audio recordings of bird calls, from other sources. In this section, we give a tour of the Birdsnap web site and iPhone app, and describe the sources and presentation of the additional information.

The Birdsnap system consists of a web site, launched in October 2013, and mobile app, launched in May 2014. Since launch, we have had about 40,000 unique web visitors and 40,000 mobile app downloads, and have processed 100,000 uploaded images through the automatic recognition system.

6.7.1 The Birdsnap Web Site

The main browsing page of the Birdsnap web site is shown in Figure 6.1. It shows all the species in the current species set arranged in a wheel. The current species set can be set to the full dataset, just the Eastern or Western species, just the “Backyard” birds (birds commonly seen in populated areas), or just the species present at a particular location and date. The Eastern, Western, and Backyard subsets are based on species inclusion in National Geographic guides to these categories [Dunn and Alderfer, 2008a; Dunn and Alderfer, 2008b; Alderfer and Hess, 2011], while the location-and-date-based subsets are obtained by thresholding our spatio-temporal prior.

In Figure 6.1, the wheel is organized by the phylogenetic “tree of life” (rendered with its tree structure), which organizes species by taxonomy. In this case we also display the taxonomic categories (order, family, subfamily, genus, and species) of the selected species.

The screenshot shows the Birdsnap website interface. At the top, the logo "birdsnap" is displayed in yellow and white. Navigation links include "USA", "Eastern", "Western", "Backyard", and "Local" (highlighted). Below this, it specifies "Birds of New York, NY on May 1". The main navigation bar includes "Bird Wheel", "Bird List" (selected), "Bird Lab", and "About". Under "Bird List", there are tabs for "Tree of Life", "Alphabetical", and "Frequency" (selected). A search bar with "Visual Recognition" and "Text Search" options is present. The main content area displays a list of species with columns for "Species Photos", "Common Name", and "Scientific Name". Each species entry consists of three circular photos and a speaker icon. The species listed are:

Species Photos	Common Name	Scientific Name
	European Starling	<i>Sturnus vulgaris</i>
	American Robin	<i>Turdus migratorius</i>
	Mourning Dove	<i>Zenaida macroura</i>
	Northern Cardinal	<i>Cardinalis cardinalis</i>
	House Sparrow	<i>Passer domesticus</i>
	Red-winged Blackbird	<i>Agelaius phoeniceus</i>

Figure 6.11: List view of species on the Birdsnap web site, here sorted by sighting frequency at the specified date and location.

Golden-winged Warbler
(*Vermivora chrysoptera*)

Description

The Golden-winged Warbler (*Vermivora chrysoptera*) is a New World warbler. It breeds in southeastern and south-central Canada and the Appalachian Mountains northeastern to north-central USA. The majority (~70%) of the global population breeds in Wisconsin, Minnesota, and Manitoba. Golden-winged Warbler populations are slowly expanding northwards, but are generally declining across its range. ... [Wikipedia](#)

Similar Species

Click < VS > below for comparison.

Black-throated Gray Warbler

Chestnut-sided Warbler

Cerulean Warbler

Tennessee Warbler

Bay-breasted Warbler

Likelihood of Sighting

Golden-winged Warbler on May 21

This map shows an estimate of how likely the Golden-winged Warbler is to be reported by a birder at a place and time. Calculation by Birdsnap based on sightings data from eBird.

Range Map

Summer (breeding)
Winter (non-breeding)
Migration

Key
♂ - male ♀ - female B - breeding NB - nonbreeding A - adult I - immature

Figure 6.12: Detail view for the Golden-winged Warbler on the web site (where it appears as a single, scrollable column).

We extract this information from a dataset provided by the Integrated Taxonomic Information System, a government-sponsored database of taxonomy [The ITIS Organization, 2014]. The wheel can also be organized alphabetically. When the species set is based on a particular location and date, the wheel can also be sorted by frequency, *i.e.* by value of the spatio-temporal prior at that location and date. The central image cycles through images of the selected species from our dataset. We also cycle through audio recordings of the selected species, obtained from xeno-canto.org [Xeno-canto Foundation, 2014], an online forum and audio recording repository maintained by birding enthusiasts. Species visually similar to the currently selected species are highlighted and marked with a star to help the user avoid a mis-identification. When a location and date are set, additional symbols are used to mark species that have recently arrived or will soon leave the area, based on the smoothed distributions described in Section 6.5, as many birders make a point of looking for these birds when they have the chance.

The main page can also be viewed as a list, as shown in Figure 6.11. This view can be more intuitive when the current species set is ordered by frequency, or when showing a ranked list of recognition results for a user image.

When a species is selected, we present a detail view, shown in Figure 6.12, with five sections arranged top to bottom.

- A header with a sampling of images from the dataset, with annotations indicating sex, age category, and plumage where available.
- A text description of the species, the first paragraph of the species' entry in Wikipedia [Wikipedia Foundation, 2014], with a link to the article.
- A radiating diagram showing the five species most visually similar to the selected species. The radial images are links to the detail views of the other species, allowing the user to traverse the graph of similar species. The “vs” arms of the diagram are links to a the field marks view shown in Figure 6.13.



Figure 6.13: Fields marks view from the web site, showing the differences between the Golden-winged Warbler and the Chestnut-sided Warbler with illustrations generated by the process described in Section 6.6.

- An animated map showing the sightings density for this species over the course of a year, generated from weekly slices of the spatio-temporal prior.
- An additional, static map showing the season range of the species according to ornithological experts. The data for these maps is provided by BirdLife International [BirdLife International, 2011], a coalition of conservation organizations.

Users upload their images for automatic recognition by clicking on the “Visual Recognition” button on the main page, which takes them to a view where they can choose an image, specify the date and location of capture, and click on the rough locations for the eye and tail that are used to accelerate the part detector (Figure 6.14). If the image file includes embedded date or location information, we prepopulate those fields. The ranked

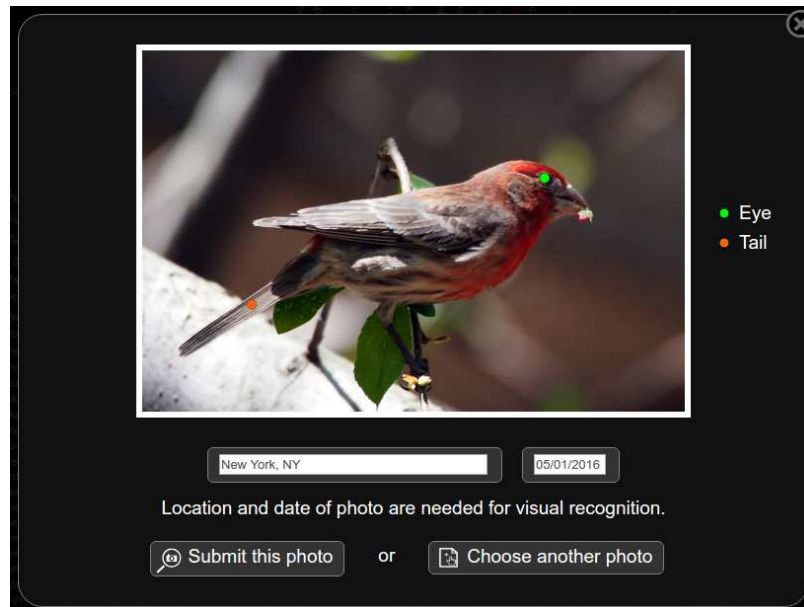


Figure 6.14: The recognition submission window on the web site, after the user has clicked on the eye and tail.

recognition results are then displayed in either the wheel or list view.

6.7.2 The Birdsnap Mobile App

The Birdsnap mobile app includes the same features as the web site, with a modified design to suit the smaller screen and touch-based interface. The main screens are shown in Figure 6.15. The top row shows the main list of species (this becomes the wheel shown in Figure 6.2 when the phone is turned to landscape orientation), a screen from the image upload flow after the user has tapped the approximate eye location, and the recognition results screen showing the ranked list of species. When uploading an image for recognition, users can either choose a photo from their library or use the device camera. The second and third rows show, for the Canada Warbler, the photos screen with images from our dataset, the description screen with text from Wikipedia, the similar species screen, the field marks screen showing differences with the Magnolia Warbler, the animated sightings density map, and the static range map. These screens correspond to, and have the same features as, the

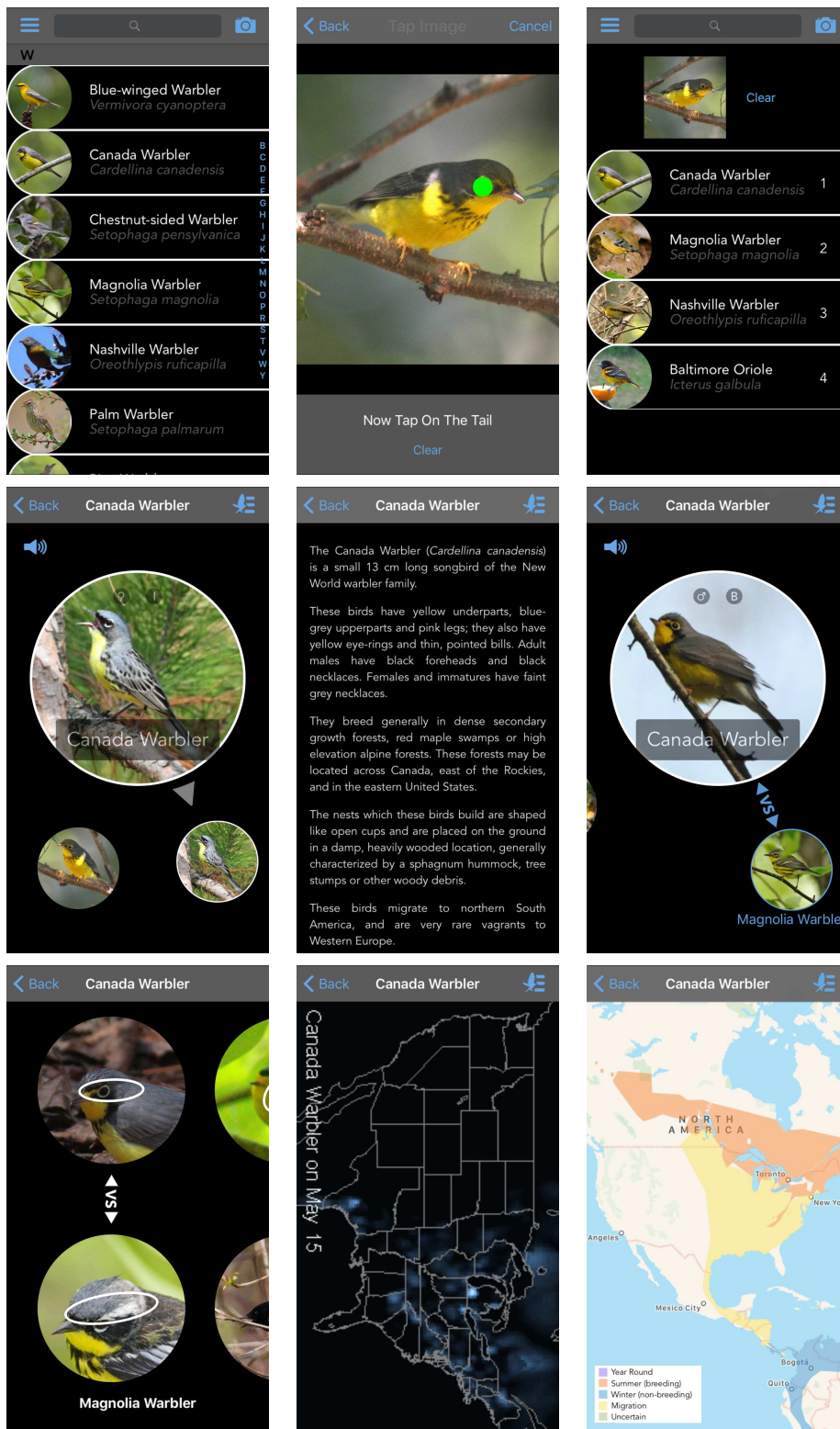


Figure 6.15: Screens from the Birdsnap iPhone application.

detail view and field marks view of the web site, shown in Figures 6.12 and 6.13.

Chapter 7

Conclusions

In this thesis, we have shown how to automatically learn a large library of part-based features for fine-grained classification within a particular basic-level category, based on a set of images with subcategory and part labels. We built an initial system for face verification, then generalized and simplified it to create “POOFs” – Part-based One-vs-One Features. Through experiments, we confirmed that POOFs work well for classification in both domains we’ve experimented with, faces and bird species. We also showed that they generalize effectively not only to classes outside the set on which they were learned (as seen in our face verification experiments), but also to semantically different *types* of classes (as shown in our experiments on attribute classification). This ability to generalize makes them especially useful when labels for the classes of interest are difficult or expensive to obtain.

We have also shown that POOFs are useful for more than just classification, demonstrating how they can be used both to find classes that are visually similar to each other and, given two similar classes, show the subtle differences that distinguish them from each other.

To showcase POOFs effectiveness for both fine-grained classification and the illustration of similarity and differences between classes, we have described how to generate a field guide to a category from a dataset of images labeled with part and subcategory labels. We’ve used these techniques to actually such a guide, Birdsnap, which we have made

available as a web site and mobile application.

To build Birdsnap, we collected a new, 500-species dataset. This is a much larger set of classes than previously available datasets for fine-grained classification, and includes some species that are nearly (visually) indistinguishable from each other. To mitigate the difficulty in recognizing these classes, we developed a modified “one-vs-most” classification scheme, and incorporated side information in the form of a spatio-temporal prior into the classifier. Birdsnap has seen 80,000 users and has run classification on over 100,000 uploaded user images.

7.1 Recent Developments

Since the publication of the work described in this thesis, methods using convolutional neural networks (CNNs) have achieved excellent results in many areas of computer vision (and elsewhere). In fine-grained categorization, the need for features extracted from corresponding part locations is now well established, so recent work from several authors has focused on how to best make use of part locations in a CNN-based classification system.

[Branson *et al.*, 2014] do a systematic comparison of prior efforts, including ours, on fine-grained classification of the CUB-200-2011 dataset. In their analysis, they categorize classification methods by how parts are used for alignment, the features on which the classifier is based, and the type of the classifier itself. Their best method has much in common with ours in the first and last respects. For the first, they use subsets of parts to find a similarity transform between corresponding regions, but where we use just two parts to solve exactly for a transformation, they use up to five parts to solve more robustly for a least-squared-error transformation. For the last, they use one-vs-all SVMs as we do. The key difference is their features themselves. The most effective features in their experiments are the concatenated outputs from several layers of a convolutional network (they use AlexNET [Krizhevsky *et al.*, 2012]), pre-trained on ImageNET and then fine-tuned on the CUB-200-2011 training data. Using these alignments and features, with a new part detector capable

of accurate part localization without bounding boxes ([Branson *et al.*, 2013]), their method achieves 75.7% accuracy on the test set, or 85.4% when using ground truth part locations.

[Krause *et al.*, 2015] use no manually-labeled part annotations, at test time or training time. Instead, they perform a foreground-background segmentation of the images in the training set, use shape context to find correspondences for points sampled along the boundary, and use these “parts” in place of the semantically meaningful parts included with the dataset. They achieve an accuracy of 73.7% without use of bounding boxes, using features from a very similar convolutional network (Caffe’s standard “CaffeNet” [Jia *et al.*, 2014], again trained on ImageNET and fine-tuned on CUB-200-2011) for the sake of fair comparison. This pays a small accuracy cost relative to the results from Branson *et al.*’s method above, but avoids the substantial expense of gathering part labels. By replacing the network with the deeper VGG-19 network from [Simonyan and Zisserman, 2015], they boost accuracy to 82.0%, although it’s very plausible that Branson *et al.* could similarly benefit from switching networks.

Our conclusion is that while detecting semantic, manually-chosen parts to set regions for feature extraction still provides the best accuracy, convolutional networks first trained on a very large, more general dataset (ImageNet), and then fine-tuned on a category-specific dataset (CUB-200-2011) are a more powerful representation than our POOF features, which do not take advantage of the additional, out-of-domain dataset. While POOFs can also make use of a dataset not labeled with the subcategories, as we showed in the attribute classification experiments in Chapter 4, it must be a dataset of the same domain and include part labels in order to train POOFs. The core intent of using classifier outputs (now the outputs of pre-trained convolutional networks, rather than our outputs of domain-relevant SVMs) is the same, and very successful.

Bibliography

- [Alderfer and Hess, 2011] Jonathan Alderfer and Paul Hess. *National Geographic Backyard Guide to the Birds of North America*. National Geographic, 2011.
- [Amazon, 2013] Amazon. Amazon mechanical turk, 2013.
- [Asthana *et al.*, 2011a] Akshay Asthana, Michael J. Jones, Tim K. Marks, Kinh H. Tieu, and Roland Goecke. Pose normalization via learned 2d warping for fully automatic face recognition. In *Proceedings of the British Machine Vision Conference*, 2011.
- [Asthana *et al.*, 2011b] Akshay Asthana, Tim K. Marks, Michael J. Jones, Kinh H. Tieu, and M. V. Rohith. Fully automatic pose-invariant face recognition via 3d pose normalization. In *Proceedings of the IEEE International Conference on Computer Vision*, 2011.
- [Bay *et al.*, 2006] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *Proceedings of the European Conference on Computer Vision*, 2006.
- [Belhumeur *et al.*, 2011] Peter N. Belhumeur, David W. Jacobs, David J. Kriegman, and Neeraj Kumar. Localizing parts of faces using a consensus of exemplars. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011.
- [Berg *et al.*, 2010] Tamara L Berg, Alexander C Berg, and Jonathan Shih. Automatic attribute discovery and characterization from noisy web data. In *Proceedings of the European Conference on Computer Vision*, 2010.

- [BirdLife International, 2011] BirdLife International. birdlife.org, 2011.
- [Blanz and Vetter, 2003] V. Blanz and T. Vetter. Face recognition based on fitting a 3d morphable model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, September 2003.
- [Branson *et al.*, 2010] Steve Branson, Catherine Wah, Boris Babenko, Florian Schroff, Peter Welinder, Pietro Perona, and Serge Belongie. Visual recognition with humans in the loop. In *Proceedings of the European Conference on Computer Vision*, 2010.
- [Branson *et al.*, 2013] Steve Branson, Oscar Beijbom, and Serge Belongie. Efficient large-scale structured learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013.
- [Branson *et al.*, 2014] Steve Branson, Grant Van Horn, Pietro Perona, and Serge Belongie. Bird species recognition using pose normalized deep convolutional nets. In *Proceedings of the British Machine Vision Conference*, 2014.
- [Cootes *et al.*, 2000] T. F. Cootes, K. Walker, and C. J. Taylor. View-based active appearance models. In *Proceedings of the IEEE Conference on Automatic Face and Gesture Recognition*, 2000.
- [Dalal and Triggs, 2005] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2005.
- [Deng *et al.*, 2013] Jia Deng, Jonathan Krause, and Li Fei-Fei. Fine-grained crowdsourcing for fine-grained recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013.
- [Doersch *et al.*, 2012] Carl Doersch, Saurabh Singh, Abhinav Gupta, Josef Sivic, and Alexei A. Efros. What makes paris look like paris? *ACM Transactions on Graphics*, 31(4), 2012.

- [Duan *et al.*, 2012] Kun Duan, Devi Parikh, David Crandall, and Kristen Grauman. Discovering localized attributes for fine-grained recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012.
- [Dunn and Alderfer, 2008a] Jon L. Dunn and Jonathan Alderfer. *National Geographic Field Guide to the Birds of Eastern North America*. National Geographic, 2008.
- [Dunn and Alderfer, 2008b] Jon L. Dunn and Jonathan Alderfer. *National Geographic Field Guide to the Birds of Western North America*. National Geographic, 2008.
- [Edwards *et al.*, 1998] G.J. Edwards, C. J. Taylor, and T. F. Cootes. Interpreting face images using active appearance models. In *Proceedings of the IEEE Conference on Automatic Face and Gesture Recognition*, 1998.
- [Elkan, 2001] Charles Elkan. The foundations of cost-sensitive learning. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 2001.
- [Enzweiler and Gavrilu, 2009] Markus Enzweiler and Dariu Gavrilu. Monocular Pedestrian Detection: Survey and Experiments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(12), 2009.
- [Everingham *et al.*, 2005 2013] Mark Everingham, Luc van Gool, Chris Williams, John Winn, and Andrew Zisserman. The pascal visual object classes homepage, 2005-2013.
- [Everingham *et al.*, 2014] Mark Everingham, S. M. Ali Eslami, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge – a retrospective. *International Journal of Computer Vision*, 2014.
- [Everingham *et al.*, 2016] Mark Everingham, Luc van Gool, Chris Williams, John Winn, and Andrew Zisserman. The pascal visual object classes leaderboard, 2016.
- [Fan *et al.*, 2008] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9, 2008.

- [Farrell *et al.*, 2011] 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 *Proceedings of the IEEE International Conference on Computer Vision*, 2011.
- [Felzenszwalb *et al.*, 2010] Pedro F. Felzenszwalb, Ross B. Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9), 2010.
- [Fink *et al.*, 2010] Daniel Fink, Wesley Hochachka, Benjamin Zuckerberg, David Winkler, Ben Shaby, M. Arthur Munson, Giles Hooker, Mirek Riedewald, Daniel Sheldon, and Steve Kelling. Spatiotemporal Exploratory Models for Broad-scale Survey Data. *Ecological Applications*, 20(8), 2010.
- [Fisher, 1936] Ronald A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2), 1936.
- [Freund and Schapire, 1997] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, August 1997.
- [Futuyma, 1997] Douglas J. Futuyma. *Evolutionary Biology*, page 763. Sinauer Associates, 1997.
- [Gu and Kanade, 2008] Leon Gu and Takeo Kanade. A generative shape regularization model for robust face alignment. In *Proceedings of the European Conference on Computer Vision*, 2008.
- [Hoiem *et al.*, 2012] Derek Hoiem, Yodsawalai Chodpathumwan, and Qieyun Dai. Diagnosing error in object detectors. In *Proceedings of the European Conference on Computer Vision*, 2012.

- [Huang *et al.*, 2007a] Gary B. Huang, Vidit Jain, and Erik Learned-Miller. Unsupervised joint alignment of complex images. In *Proceedings of the IEEE International Conference on Computer Vision*, 2007.
- [Huang *et al.*, 2007b] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical report, University of Massachusetts, Amherst, 2007.
- [Jetz *et al.*, 2012] W. Jetz, G. H. Thomas, J. B. Joy, K. Hartmann, and A. O. Mooers. The global diversity of birds in space and time. *Nature*, 491(7424), 2012.
- [Jia *et al.*, 2014] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the ACM International Conference on Multimedia*. ACM, 2014.
- [Khosla *et al.*, 2011] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Li Fei-Fei. Novel dataset for fine-grained image categorization. In *Workshop on Fine-Grained Visual Categorization at the IEEE Conference on Computer Vision and Pattern Recognition*, 2011.
- [Krause *et al.*, 2013] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3D object representations for fine-grained categorization. In *Workshop on Fine-Grained Visual Categorization at the IEEE International Conference on Computer Vision*, 2013.
- [Krause *et al.*, 2015] Jonathan Krause, Hailin Jin, Jianchao Yang, and Li Fei-Fei. Fine-grained recognition without part annotations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5546–5555, 2015.
- [Krizhevsky *et al.*, 2012] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2012.

- [Kumar *et al.*, 2009] Neeraj Kumar, Alexander C. Berg, Peter N. Belhumeur, and Shree K. Nayar. Attribute and simile classifiers for face verification. In *Proceedings of the IEEE International Conference on Computer Vision*, 2009.
- [Kumar *et al.*, 2011] Neeraj Kumar, Alexander C. Berg, Peter N. Belhumeur, and Shree K. Nayar. Describable visual attributes for face verification and image search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, October 2011.
- [Kumar *et al.*, 2012] Neeraj Kumar, Peter N. Belhumeur, Arijit Biswas, David W. Jacobs, W. John Kress, Ida Lopez, and Joo V. B. Soares. Leafsnap: A computer vision system for automatic plant species identification. In *Proceedings of the European Conference on Computer Vision*, 2012.
- [Learned-Miller, 2006] Erik G. Learned-Miller. Data driven image models through continuous joint alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, February 2006.
- [Letunic and Bork, 2007] Ivica Letunic and Peer Bork. Interactive tree of life (itol): An online tool for phylogenetic tree display and annotation. *Bioinformatics*, 23(1), 2007.
- [Liu and Belhumeur, 2013] Jiongxin Liu and Peter N. Belhumeur. Bird part localization using exemplar-based models with enforced pose and subcategory consistency. In *Proceedings of the IEEE International Conference on Computer Vision*, 2013.
- [Liu *et al.*, 2012] Jiongxin Liu, Angjoo Kanazawa, David Jacobs, and Peter Belhumeur. Dog breed classification using part localization. In *Proceedings of the European Conference on Computer Vision*, 2012.
- [Liu *et al.*, 2015] Wei Liu, Olga Russakovsky, Jia Deng, Fei-Fei Li, and Alex Berg. Imagenet large scale visual recognition challenge 2015, 2015.
- [Liu *et al.*, 2016] Wei Liu, Bolei Zhou, Olga Russakovsky, Jia Deng, Fei-Fei Li, and Alex Berg. Imagenet large scale visual recognition challenge 2016, 2016.

- [Lowe, 1999] D.G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the IEEE International Conference on Computer Vision*, 1999.
- [Maji *et al.*, 2013] S. Maji, J. Kannala, E. Rahtu, M. Blaschko, and A. Vedaldi. Fine-grained visual classification of aircraft. Technical report, 2013.
- [Martinez-Munoz *et al.*, 2009] G. Martinez-Munoz, N. Larios, E. Mortensen, Wei Zhang, A. Yamamuro, R. Paasch, N. Payet, D. Lytle, L. Shapiro, S. Todorovic, A. Moldenke, and T.G. Dietterich. Dictionary-free categorization of very similar objects via stacked evidence trees. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009.
- [Nguyen and Bai, 2011] Hieu V. Nguyen and Li Bai. Cosine similarity metric learning for face verification. In *Proceedings of the Asian Conference on Computer Vision*, 2011.
- [Nilsback and Zisserman, 2008] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian Conf. Computer Vision Graphics and Image Processing*, 2008.
- [Ojala *et al.*, 2002] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7), 2002.
- [Omron,] Omron. OKAO vision. http://www.omron.com/r_d/coretech/vision/okao.html.
- [Parikh and Grauman, 2011] Devi Parikh and Kristen Grauman. Interactively building a discriminative vocabulary of nameable attributes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011.
- [Parkhi *et al.*, 2012] Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

- [Pinto and Cox, 2011] Nicolas Pinto and David D. Cox. Beyond Simple Features: A Large-Scale Feature Search Approach to Unconstrained Face Recognition. In *Conference on Automatic Face and Gesture Recognition*, 2011.
- [Pinto *et al.*, 2011] Nicolas Pinto, Zak Stone, Todd Zickler, and David D. Cox. Scaling-up biologically-inspired computer vision: A case-study on facebook. In *Workshop on Biologically Consistent Vision at the IEEE Conference on Computer Vision and Pattern Recognition*, 2011.
- [Platt, 1999] John C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3), 1999.
- [Prasong and Chamnongthai, 2012] Pusig Prasong and Kosin Chamnongthai. Face-Recognition-Based dog-Breed classification using size and position of each local part, and pca. In *Proceedings of the International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, 2012.
- [Russakovsky *et al.*, 2013] Olga Russakovsky, Jia Deng, Zhiheng Huang, Alexander C. Berg, and Li Fei-Fei. Detecting avocados to zucchinis: What have we done, and where are we going? In *Proceedings of the IEEE International Conference on Computer Vision*, 2013.
- [Russakovsky *et al.*, 2015] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.
- [Saitou and Nei, 1987] Naruya Saitou and Masatoshi Nei. The neighbor-joining method: A new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution*, 4(4), 1987.

- [Schroff *et al.*, 2015] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 815–823, 2015.
- [Shrivastava *et al.*, 2011] Abhinav Shrivastava, Tomasz Malisiewicz, Abhinav Gupta, and Alexei A Efros. Data-driven visual similarity for cross-domain image matching. *ACM Transactions on Graphics*, 30(6), 2011.
- [Sibley, 2000] David Allen Sibley. *The Sibley Guide to Birds*. Knopf, 2000.
- [Simonyan and Zisserman, 2015] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proceedings of the International Conference on Learning Representations*, 2015.
- [Sterry, 2006] Paul Sterry. *Collins Complete Guide to British Wild Flowers*. Collins, 2006.
- [Sullivan *et al.*, 2009] Brian L. Sullivan, Christopher L. Wood, Marshall J. Iliff, Rick E. Bonney, Daniel Fink, and Steve Kelling. eBird: A citizen-based bird observation network in the biological sciences. *Biological Conservation*, 142(10), 2009.
- [Svensson *et al.*, 2011] Lars Svensson, Killian Mullarney, and Dan Zetterström. *Collins Bird Guide*. Collins, 2011.
- [Terrell and Scott, 1992] George Terrell and David Scott. Variable Kernel Density Estimation. *The Annals of Statistics*, 20(3), 1992.
- [The ITIS Organization, 2014] The ITIS Organization. The integrated taxonomic information system, 2014.
- [Tversky and Hemenway, 1984] Barbara Tversky and Kathleen Hemenway. Objects, parts, and categories. *J. Experimental Psychology: General*, 113(2), 1984.
- [University of Massachusetts,] University of Massachusetts. LFW web site. <http://www.cs.umass.edu/lfw/>.

- [Van Horn *et al.*, 2015] Grant Van Horn, Steve Branson, Ryan Farrell, Scott Haber, Jessie Barry, Panos Ipeirotis, Pietro Perona, and Serge Belongie. Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.
- [Wah *et al.*, 2011a] Catherine Wah, Steve Branson, Pietro Perona, and Serge Belongie. Multiclass recognition and part localization with humans in the loop. In *Proceedings of the IEEE International Conference on Computer Vision*, 2011.
- [Wah *et al.*, 2011b] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.
- [Wang *et al.*, 2006] Peng Wang, Lam Cam Tran, and Qiang Ji. Improving face recognition by online image alignment. In *Proceedings of the International Conference on Pattern Recognition*, 2006.
- [Wang *et al.*, 2009] Josiah Wang, Katja Markert, and Mark Everingham. Learning models for object recognition from natural language descriptions. In *Proceedings of the British Machine Vision Conference*, 2009.
- [Welinder *et al.*, 2010] 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, 2010.
- [Wikipedia Foundation, 2014] Wikipedia Foundation. Wikipedia, 2014.
- [Wolf *et al.*, 2008] Lior Wolf, Tal Hassner, and Yaniv Taigman. Descriptor based methods in the wild. In *Workshop on Faces in Real Life Images at the European Conference on Computer Vision*, 2008.

- [Wolf *et al.*, 2009] Lior Wolf, Tal Hassner, and Yaniv Taigman. Similarity scores based on background samples. In *Proceedings of the Asian Conference on Computer Vision*, 2009.
- [Wolpert, 1992] David H Wolpert. Stacked generalization. *Neural networks*, 5(2):241–259, 1992.
- [World Institute for Conservation and Environment, 2013] World Institute for Conservation and Environment. birdlist.org, 2013.
- [Xeno-canto Foundation, 2014] Xeno-canto Foundation. Xeno-canto, 2014.
- [Yanai and Barnard, 2005] Keiji Yanai and Kobus Barnard. Image region entropy: A measure of visualness of web images associated with one concept. In *Proceedings of the ACM International Conference on Multimedia*, 2005.
- [Yang *et al.*, 2015] Linjie Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang. A large-scale car dataset for fine-grained categorization and verification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.
- [Yao *et al.*, 2011] Bangpeng Yao, Aditya Khosla, and Li Fei-Fei. Combining randomization and discrimination for fine-grained image categorization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011.
- [Yao *et al.*, 2012] Bangpeng Yao, Gray Bradski, and Li Fei-Fei. A codebook-free and annotation-free approach for fine-grained image categorization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012.
- [Yin *et al.*, 2011] Qi Yin, Xiaoou Tang, and Jian Sun. An associate-Predict model for face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

- [Zadrozny and Elkan, 2002] Bianca Zadrozny and Charles Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2002.
- [Zhang *et al.*, 2012] Ning Zhang, Ryan Farrell, and Trevor Darrell. Pose pooling kernels for sub-category recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

Appendix: The Birdsnap Dataset

To experiment with fine-grained visual categorization with a large number of subcategories, and to build the Birdsnap web site and mobile app, we assembled a new dataset of bird images, the Birdsnap dataset. The Birdsnap dataset contains 49,829 images of 500 of North American bird species. Each image is labeled with species, bounding box, and the locations of 17 parts on the bird's body (or however many of those parts are visible in the image). The dataset is considerably larger than any previously-existing fine-grained classification dataset we are aware of. In particular, it is natural to compare it with the Caltech / USCD Birds-200-2011 dataset (CUB-200-2011) from [Wah *et al.*, 2011b], which contains 11,788 images of 200 species. In this appendix we describe the motivation behind the dataset, give the details of how it was constructed, and compare it with other fine-grained categorization datasets.

A.1 Motivation Behind the Dataset

The initial work that led to the creation of Birdsnap, described in Chapters 4 and 5, used the well-known CUB-200-2011 dataset. When we set out to actually build a guide to birds, however, we discovered that a guide to the birds in the this dataset would not be useful to anyone. Bird guides are generally regional, and must be comprehensive of at least the common species in that region, so that a user sighting a bird in the region can be confident of finding it in the guide. CUB-200-2011 was not built with this in mind. While about two-thirds of the species in it are found in the United States, some very common

American species, for example the American Robin, Canada Goose, and Rock Pigeon, are not included. Deciding our guide would cover birds commonly occurring in the United States, we set out to create a comprehensive dataset in this domain.

A.2 Building the Dataset

First, we determined the species we would like to include in the dataset. Conservation site birdlist.org [World Institute for Conservation and Environment, 2013] provides presence and abundance information for bird species by region. The list for the United States includes 548 species marked as “common to occasional” generally meaning at least 100 sightings have been recorded. We begin with this as our list of species.

To obtain the pool of images from which we would build the dataset, we searched for the scientific name of each species on Flickr. Our expectation was that Flickr users who tag their photos with the scientific names of the species are likely to have some expertise in identifying birds, so these labels would be more accurate than those we would obtain by searching for the common names, and this was confirmed by examination of the downloaded images. We set a target of 100 confirmed (as described below) images for each species. For species where we did not find enough images to meet this target, we supplemented the scientific name search results with common name searches. In all, we downloaded over 600,000 images.


To label the dataset, we relied on Amazon Mechanical Turk [Amazon, 2013], which allows us to post jobs as online forms to be completed by human workers. With Mechanical Turk we are able to obtain a large number of labels quickly and inexpensively, but the quality of labeling is inconsistent, so it’s important to have each labeling job done by multiple workers, and do some averaging or outlier detection. We label the images in three steps, first establishing that the image is a photograph of a single bird and obtaining the bird’s bounding box, then labeling the part locations, and finally confirming the species identification and getting subtype information (sex, age, plumage, or subspecies).

Draw a tight rectangle around the bird in the image below.
 The rectangle should fit *tightly* around the bird, and include the whole body (as much as you can see). If the image is not a photograph, or there is no bird in the image, or there multiple birds in the image, *do not* draw a rectangle. Instead choose the reason and click "submit bad image."

Please do not try to do this HIT using Microsoft Internet Explorer. It doesn't work. The HIT works in [Google Chrome](#), [Mozilla Firefox](#), and Apple Safari.

After you submit each image, the next image will appear. There are 6 images in the HIT.

Good Examples:



Bad Examples:





Image 3 of 6



Draw a tight rectangle around the bird and

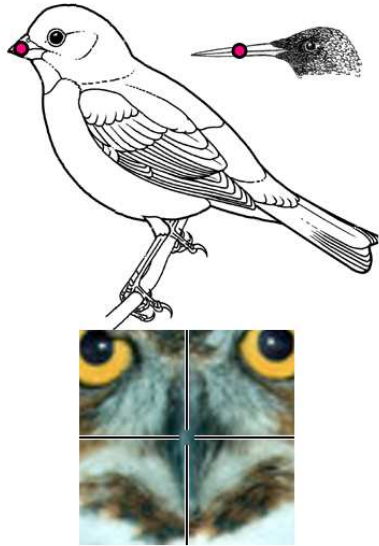
This is a bad image, because

Figure A.1: The Amazon Mechanical Turk interface for bounding box labeling.

Click on the center of the beak in the bird photograph below on the right.

- The meaning of "center of the beak" is shown by the diagram below on the left.
- If the center of the beak is **not visible**, the image is **not a bird**, the image has **multiple birds**, or the image is **not a photograph**, choose the correct option from the "choose a reason" box and click "submit bad image."
- This HIT includes **5** images.

Please do not try to do this HIT using Microsoft Internet Explorer. It doesn't work. The HIT works in [Google Chrome](#), [Mozilla Firefox](#), and Apple Safari.






Image 1 of 5

Click on the **center of the beak** and

or

This is a bad image, because

Figure A.2: The Amazon Mechanical Turk interface for part labeling.

- **Image filtering and Bounding box.** The first task is to confirm that the image is suitable for our dataset at all. We want to exclude three types of images: (a) drawings or other non-photograph images, (b) images that do not actually contain a bird at all, and (c) images that include multiple birds (we exclude these for simplicity and to increase the rate at which the bird matches the species tag we searched for). We combine this task with the labeling of bounding boxes, asking the labeler to either identify which of the three rejection criteria apply to the image or draw a tight bounding box around the bird. Figure A.1 shows the Mechanical Turk interface for this task.

For this job, each image is presented to six labelers. Based on the six responses, we discard any image that more than half the workers indicated should be rejected. The remaining images have between three and six bounding box labels. For an image with n bounding box labels, we find the subset of $\lceil n/2 \rceil$ labels with the smallest sum of variances over the top-left and bottom-right corners of the box and use the mean of this subset as the final bounding box label. We find that this simple method of outlier rejection gives us reliable, tight bounding boxes.

- **Part location and visibility.** In the next task, we collect locations for the seventeen parts of the bird. The parts in our dataset are the back, beak, belly, breast, crown, forehead, nape, tail, throat, right cheek, right eye, right leg, right wing, left cheek, left eye, left leg, and left wing. These were chosen as a superset of the parts in CUB-200-2011, to allow easy comparisons with that dataset. We added “left cheek” and “right cheek” ad hoc based on their common occurrence in bird descriptions in guide books.

Each part location job is specific to one of the seventeen parts, to allow the labeler to understand the part we’re interested and label a large number of images quickly. The task specifies the name of the part to be labeled, shows its location on a diagram of a generic bird, and asks the labeler to click the location of the part in an image

from the dataset, cropped based on the bounding box from the previous labeling step (slightly expanded to allow labeling of parts just on the edge of the bounding box). A zoomed-in view of the part of the image under the cursor is included to allow precise placement of the annotation. The worker can also specify that the image meets one of the rejection criteria from the previous labeling step, to catch images that should have been filtered out at that step, or that the part is not visible in the image. The Mechanical Turk interface is shown in Figure A.2.

Each part location job is shown to four labelers, with the image kept if no more than one labeler indicates that it should be rejected. For images that are kept, we consider the part visible or hidden based on a majority vote of the labelers that did not reject the image, and for parts that are visible, we take an inlier mean of the labeled locations, as we did with the bounding boxes, by calculating the mean location from the subset of $\lceil n/2 \rceil$ labels with the smallest variance, where n is the number of labelers who marked the part as visible. Where necessary, to break ties in the visible / hidden decision or to ensure that the low-variance subset of labels includes at least two labels, we present the job to additional workers beyond the initial four.

- **Species and subtype.** In the final task, we confirm that the image we found by searching for a particular species is actually an image of that species. While this is usually the case, incorrect tags occur as a result of both mis-identifications by Flickr users and user captions that mention a species but do not indicate that the species is in the image. As our labelers are not expert birders, for the most part they cannot confirm species labels unaided, so in the labeling interface we include both the name of the species and its images from a guidebook book [Sibley, 2000]. The guidebook generally includes an image for each distinct appearance taken by the species – for example if males and females of the species have different appearances, or if the male has different plumage in the breeding and nonbreeding seasons, enough illustrations are included to cover the variation. By including all the images for the species and asking the labeler to choose one (or none, indicating the species is incorrect), we



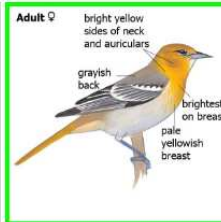
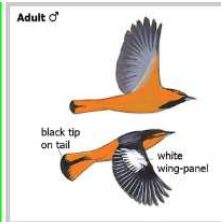


Below are 6 drawings, showing 6 different **categories** of bird in the species "Bullock's Oriole". Below the drawings is a single **test photograph** of a Bullock's Oriole. Click on the category whose appearance most closely matches the test photograph, then click "next" (or type "n") to see another photograph. After choosing a drawing for 9 photographs, click "submit" to submit the job.

How to match: The most important thing is to match the bird itself, not the bird's activity (flying/swimming/standing) or pose. For example, it is okay to match a flying bird photograph to a standing bird category drawing. However, if there is a drawing that matches the bird *and* its action, please choose it.


Category drawings of flying birds are an exception. Please **only choose a flying category image if it matches the appearance of the bird in the test photograph and the bird in the test photograph is also flying.**

If the photograph does not belong any of these categories (for example if it is not a Bullock's Oriole at all, or there is no bird in the photograph), please choose "Other Species." Please use [Google Chrome](#), [Mozilla Firefox](#), or Apple Safari for this HIT. Do not use Internet Explorer.

Categories

<p>Adult ♂</p> 	<p>1st year ♂</p> <p>black throat appears by Oct</p>  <p>black eye-line and orange auriculars</p>	<p>Adult ♀</p> <p>bright yellow sides of neck and auriculars</p>  <p>grayish back</p> <p>brightest on breast</p> <p>pale yellowish breast</p>	<p>Adult ♂</p>  <p>black tip on tail</p> <p>white wing-panel</p>
<p>Adult ♀</p> 	<p>Bullock's x Baltimore Oriole hybrid adult ♂</p> <p>intermediate pattern on head, wing coverts, and tail; blackcrosses produce complete range of variation between parent species</p> 	<p style="font-size: 2em; font-weight: bold;">Other Species</p>	

Test Photograph



Previous

Next

Submit

Photograph 5 of 9

Figure A.3: The Amazon Mechanical Turk interface for species and sub-species class labeling.

get not only species confirmation, but an additional subtype label. This interface is shown in Figure A.3.

Each species and subtype job is shown to five labelers, with the overall species label accepted if more than half the labelers mark it as correct (even if they disagree on the subtype). For the subtype labels, we mark each illustration from the guidebook to indicate what age (adult or immature), sex (male or female), and plumage (breeding or nonbreeding) information it provides. For example the selected subtype image in Figure A.3 shows an adult female with no plumage specification. For each of the three variables, if more than half of the labelers agree on its value, and no more than one labeler has applied the opposing value, we record the value as a subtype label for the image. As an example, if two workers choose an “adult male” illustration, two workers choose an “immature male” illustration, and one worker chooses an “adult” illustration not marked with sex, we record the image as male but do not apply an age label. The result is that only a minority of the images in the dataset include subtype labels – the subtype distinctions are difficult to make in many photographs – but the labels are reliable when present.

After several rounds of searching, downloading, and labeling, we had reached the target of 100 images for most species – indeed, for most species, it was easy to find thousands of images, although we labeled only enough to reach our target. For some species, mostly species of very limited range either overall (*e.g.* Abert’s Towhee) or in the United States (*e.g.* the Northern Beardless Tyrannulet), we fell well short of 100 images. To avoid underrepresented classes, we cut the dataset to include the 500 species with the largest numbers of successfully labeled images. This leaves us with a dataset in which 24 of the 500 species have fewer than 100 images. The most poorly represented species is the White-throated Swift with 69 images, and there are 48,829 images in all. Table A.1 shows the final number of images per species, along with the number of those images with additional sex, age, or plumage labels.

A.3 Comparisons with Other Datasets

While the most obvious comparison to our dataset is with CUB-200-2011, there are several other prior datasets for fine-grained classification, and in particular a number of species datasets. All of them are substantially smaller than the Birdsnap dataset, and most of them are not sufficiently comprehensive of a set suitable for building a guide. The Oxford Flowers [Nilsback and Zisserman, 2008] dataset includes 8189 images spanning 102 species of British flower, a small fraction of the 1039 species in the comprehensive *Collins Complete Guide to British Wild Flowers* [Sterry, 2006]. The Leafsnap dataset [Kumar *et al.*, 2012] is comprehensive over all tree species in the northeastern United States – and in fact, its curators have built an online guide – but is smaller than Birdsnap, with 30,866 images of 185 species. The STONEFLY9 dataset [Martinez-Munoz *et al.*, 2009] is comprehensive but small, covering the niche domain of stoneflies, with 9 classes, 3826 images of 773 specimens. None of these datasets include part annotations.

Outside of species datasets, there are several datasets of cat and dog breeds. The Stanford Dogs dataset [Khosla *et al.*, 2011] includes 20,580 images of 120 dog breeds, with bounding boxes, the Oxford-IIIT Pet dataset [Parkhi *et al.*, 2012] has 7349 images of 37 breeds of cat and dog with bounding boxes of the head and foreground-background segmentation of the full body – combined, these form a rough parts annotation. And the Columbia Dogs with Parts dataset [Liu *et al.*, 2012] holds 8351 images of 133 breeds with face bounding boxes and 8 part locations (all on the face).

In non-biological domains, there are datasets for cars (Stanford Cars [Krause *et al.*, 2013] with 16,185 images over 196 make-model-year classes) and aircraft (the FGVC-Aircraft Benchmark dataset [Maji *et al.*, 2013], with 10,200 images over 102 aircraft variants).

Since the creation of the Birdsnap dataset and publication of the work in this thesis, two large datasets for fine-grained classification have been released. First is the NABirds Dataset [Van Horn *et al.*, 2015] of North American birds, which covers the same domain as Birdsnap. This dataset was collected by soliciting photographs and labeling work from

birding enthusiasts, to ensure very high quality class and part labels. This excellent dataset, at 48,526 images and 400 species, is somewhat smaller than ours, but includes more comprehensive subtype information about sex, age, and plumage, which can be used to expand the 400 species into 555 visual categories.

Another recently released dataset, CompCars dataset [Yang *et al.*, 2015], is the only fine-grained classification dataset we know of with more images and classes than Birdsnap. This dataset of car photographs contains 214,345 images of 1687 classes (make-model-year) of car, labeled with five attributes. However, due to the difficulty of distinguishing between the same make and model in different years, the authors collapse the dataset down to just 431 (make-model) classes in their classification experiments.

Species	Im	Sub	Species	Im	Sub	Species	Im	Sub
Acadian Flycatcher	100	1	Black-crested Titmouse	100	3	Carolina Wren	100	7
Acorn Woodpecker	100	6	Black-crowned Night-Heron	100	5	Caspian Tern	100	5
Alder Flycatcher	100	1	Black-headed Grosbeak	100	5	Cassin's Finch	100	5
Allen's Hummingbird	100	5	Black-legged Kittiwake	100	3	Cassin's Kingbird	100	1
Altamira Oriole	100	2	Black-necked Stilt	100	4	Cassin's Sparrow	100	1
American Avocet	100	5	Black-throated Blue Warbler	100	2	Cassin's Vireo	100	1
American Bittern	100	2	Black-throated Gray Warbler	98	2	Cattle Egret	100	4
American Black Duck	100	4	Black-throated Green Warbler	100	5	Cave Swallow	100	1
American Coot	100	7	Black-throated Sparrow	100	2	Cedar Waxwing	100	6
American Crow	100	9	Blackburnian Warbler	100	2	Cerulean Warbler	98	2
American Dipper	100	1	Blackpoll Warbler	100	2	Chestnut-backed Chickadee	100	5
American Golden-Plover	100	3	Blue Grosbeak	100	4	Chestnut-collared Longspur	100	2
American Goldfinch	100	8	Blue Jay	100	8	Chestnut-sided Warbler	100	6
American Kestrel	100	8	Blue-gray Gnatcatcher	100	4	Chihuahuan Raven	97	1
American Oystercatcher	100	1	Blue-headed Vireo	100	5	Chimney Swift	82	1
American Pipit	100	5	Blue-winged Teal	100	8	Chipping Sparrow	100	10
American Redstart	100	5	Blue-winged Warbler	100	1	Cinnamon Teal	100	4
American Robin	100	10	Boat-tailed Grackle	100	7	Clapper Rail	100	1
Am. Three-toed Woodpecker	100	2	Bobolink	100	4	Clark's Grebe	100	2
American Tree Sparrow	100	6	Bohemian Waxwing	100	1	Clark's Nutcracker	100	1
American White Pelican	100	4	Bonaparte's Gull	100	2	Clay-colored Sparrow	100	0
American Wigeon	100	5	Boreal Chickadee	100	1	Cliff Swallow	93	5
American Woodcock	100	4	Brandt's Cormorant	100	2	Common Black-Hawk	100	0
Anhinga	100	6	Brant	97	3	Common Eider	100	2
Anna's Hummingbird	100	8	Brewer's Blackbird	100	6	Common Gallinule	100	4
Arctic Tern	100	3	Brewer's Sparrow	100	0	Common Goldeneye	100	4
Ash-throated Flycatcher	100	6	Bridled Titmouse	100	1	Common Grackle	100	9
Audubon's Oriole	100	1	Broad-billed Hummingbird	100	2	Common Ground-Dove	100	3
Baird's Sandpiper	100	2	Broad-tailed Hummingbird	100	1	Common Loon	100	10
Bald Eagle	100	12	Broad-winged Hawk	100	2	Common Merganser	100	6
Baltimore Oriole	100	10	Bronzed Cowbird	100	1	Common Murre	98	2
Band-tailed Pigeon	100	5	Brown Creeper	100	4	Common Nighthawk	100	8
Barn Swallow	100	8	Brown Pelican	100	10	Common Raven	100	5
Barred Owl	100	8	Brown Thrasher	100	5	Common Redpoll	100	9
Barrow's Goldeneye	100	2	Brown-capped Rosy-Finch	100	1	Common Tern	100	6
Bay-breasted Warbler	100	3	Brown-crested Flycatcher	100	0	Common Yellowthroat	100	6
Bell's Vireo	100	1	Brown-headed Cowbird	100	8	Connecticut Warbler	100	2
Belted Kingfisher	100	8	Brown-headed Nuthatch	100	1	Cooper's Hawk	100	9
Bewick's Wren	100	6	Bufflehead	100	5	Cordilleran Flycatcher	100	1
Black Guillemot	100	1	Bullock's Oriole	100	6	Costa's Hummingbird	100	2
Black Oystercatcher	100	1	Burrowing Owl	100	1	Couch's Kingbird	100	1
Black Phoebe	100	4	Bushtit	100	5	Crested Caracara	100	1
Black Rosy-Finch	95	1	Cackling Goose	92	1	Curve-billed Thrasher	100	5
Black Scoter	94	2	Cactus Wren	100	4	Dark-eyed Junco	100	10
Black Skimmer	100	8	California Gull	100	5	Dickcissel	100	2
Black Tern	100	3	California Quail	100	7	Double-crested Cormorant	100	7
Black Turnstone	100	2	California Thrasher	100	4	Downy Woodpecker	100	9
Black Vulture	100	4	California Towhee	100	6	Dunlin	100	4
Black-and-white Warbler	100	4	Calliope Hummingbird	100	3	Dusky Flycatcher	100	1
Black-backed Woodpecker	100	2	Canada Goose	100	8	Dusky Grouse	100	2
Black-bellied Plover	100	4	Canada Warbler	98	2	Eared Grebe	100	4
Black-billed Cuckoo	100	6	Canvasback	100	2	Eastern Bluebird	101	7
Black-billed Magpie	100	8	Canyon Towhee	100	5	Eastern Kingbird	100	6
Black-capped Chickadee	100	11	Canyon Wren	100	1	Eastern Meadowlark	100	7
Black-chinned Hummingbird	100	8	Cape May Warbler	100	2	Eastern Phoebe	100	6
Black-chinned Sparrow	99	0	Carolina Chickadee	100	7	Eastern Screech-Owl	100	6

Table A.1: Species of the Birdsnap dataset, with image and category counts. Part 1 of 3.

Species	Im	Sub	Species	Im	Sub	Species	Im	Sub
Eastern Towhee	100	7	Hammond's Flycatcher	100	1	Merlin	100	9
Eastern Wood-Pewee	100	4	Harlequin Duck	97	1	Mew Gull	100	5
Elegant Trogon	100	0	Harris's Hawk	100	2	Mexican Jay	100	2
Elf Owl	76	0	Harris's Sparrow	100	1	Mississippi Kite	100	2
Eurasian Collared-Dove	100	6	Heermann's Gull	100	3	Monk Parakeet	100	0
Eurasian Wigeon	100	1	Henslow's Sparrow	100	1	Mottled Duck	99	1
European Starling	100	16	Hepatic Tanager	100	1	Mountain Bluebird	99	4
Evening Grosbeak	100	7	Hermit Thrush	100	7	Mountain Chickadee	100	7
Ferruginous Hawk	100	2	Herring Gull	100	15	Mountain Plover	100	1
Ferruginous Pygmy-Owl	100	0	Hoary Redpoll	100	3	Mourning Dove	100	10
Field Sparrow	100	4	Hooded Merganser	100	8	Mourning Warbler	100	2
Fish Crow	100	4	Hooded Oriole	100	5	Muscovy Duck	78	0
Florida Scrub-Jay	100	3	Hooded Warbler	100	2	Mute Swan	100	2
Forster's Tern	100	7	Horned Grebe	100	2	Nashville Warbler	100	4
Fox Sparrow	100	9	Horned Lark	100	7	Nelson's Sparrow	100	1
Franklin's Gull	100	2	House Finch	100	8	Neotropic Cormorant	100	2
Fulvous Whistling-Duck	100	1	House Sparrow	100	7	Northern Bobwhite	100	8
Gadwall	100	4	House Wren	100	6	Northern Cardinal	100	10
Gambel's Quail	100	4	Hutton's Vireo	100	4	Northern Flicker	100	12
Gila Woodpecker	100	5	Iceland Gull	100	2	Northern Gannet	100	3
Glaucous Gull	100	5	Inca Dove	100	7	Northern Goshawk	100	1
Glaucous-winged Gull	100	4	Indigo Bunting	100	8	Northern Harrier	100	6
Glossy Ibis	100	2	Killdeer	100	6	Northern Hawk Owl	100	1
Golden Eagle	100	7	King Rail	99	2	Northern Mockingbird	100	8
Golden-crowned Kinglet	100	6	Ladder-backed Woodpecker	100	4	Northern Parula	100	4
Golden-crowned Sparrow	100	4	Lapland Longspur	100	1	Northern Pintail	100	4
Golden-fronted Woodpecker	100	4	Lark Bunting	100	3	Northern Rough-winged Swallow	100	4
Golden-winged Warbler	100	6	Lark Sparrow	100	4	Northern Saw-whet Owl	100	8
Grasshopper Sparrow	100	1	Laughing Gull	100	11	Northern Shrike	100	6
Gray Catbird	100	6	Lazuli Bunting	100	6	Northern Waterthrush	100	2
Gray Flycatcher	100	1	Le Conte's Sparrow	100	1	Nuttall's Woodpecker	100	4
Gray Jay	100	3	Least Bittern	100	2	Oak Titmouse	100	3
Gray Kingbird	100	0	Least Flycatcher	100	5	Olive Sparrow	100	0
Gray-cheeked Thrush	100	1	Least Grebe	100	3	Olive-sided Flycatcher	100	1
Gray-crowned Rosy-Finich	100	2	Least Sandpiper	100	9	Orange-crowned Warbler	100	4
Great Black-backed Gull	100	7	Least Tern	100	3	Orchard Oriole	100	3
Great Blue Heron	100	9	Lesser Goldfinch	100	7	Osprey	100	10
Great Cormorant	100	0	Lesser Nighthawk	100	0	Ovenbird	100	3
Great Crested Flycatcher	100	4	Lesser Scaup	100	3	Pacific Golden-Plover	100	3
Great Egret	100	6	Lesser Yellowlegs	100	3	Pacific Loon	100	2
Great Gray Owl	100	1	Lewis's Woodpecker	100	1	Pacific Wren	100	0
Great Horned Owl	100	7	Limpkin	100	1	Pacific-slope Flycatcher	100	4
Great Kiskadee	100	3	Lincoln's Sparrow	100	3	Painted Bunting	100	6
Great-tailed Grackle	100	11	Little Blue Heron	100	3	Painted Redstart	100	1
Greater Prairie-Chicken	100	3	Loggerhead Shrike	100	8	Palm Warbler	100	4
Greater Roadrunner	100	4	Long-billed Curlew	100	4	Pectoral Sandpiper	100	4
Greater Sage-Grouse	100	2	Long-billed Dowitcher	100	3	Peregrine Falcon	100	4
Greater Scaup	100	3	Long-billed Thrasher	100	1	Phainopepla	100	2
Greater White-fronted Goose	92	1	Long-eared Owl	100	5	Philadelphia Vireo	100	1
Greater Yellowlegs	100	3	Long-tailed Duck	100	5	Pied-billed Grebe	100	3
Green Jay	100	1	Louisiana Waterthrush	100	1	Pigeon Guillemot	100	3
Green-tailed Towhee	100	4	Magnificent Frigatebird	100	3	Pileated Woodpecker	100	8
Green-winged Teal	96	5	Magnolia Warbler	100	4	Pine Grosbeak	100	4
Groove-billed Ani	100	1	Mallard	100	10	Pine Siskin	100	8
Gull-billed Tern	100	1	Marbled Godwit	100	3	Pine Warbler	100	5
Hairy Woodpecker	100	10	Marsh Wren	100	3	Piping Plover	100	2

Species of the Birdsnap dataset, with image and category counts. Part 2 of 3.

Species	Im	Sub	Species	Im	Sub	Species	Im	Sub
Plumbeous Vireo	100	1	Scaled Quail	100	3	Warbling Vireo	100	4
Prairie Falcon	100	1	Scarlet Tanager	100	6	Western Bluebird	100	3
Prairie Warbler	100	2	Scissor-tailed Flycatcher	100	7	Western Grebe	100	3
Prothonotary Warbler	100	7	Scott's Oriole	100	3	Western Gull	100	4
Purple Finch	100	9	Seaside Sparrow	100	1	Western Kingbird	100	6
Purple Gallinule	100	4	Sedge Wren	100	1	Western Meadowlark	100	8
Purple Martin	100	7	Semipalmated Plover	100	3	Western Sandpiper	100	4
Purple Sandpiper	100	2	Semipalmated Sandpiper	100	5	Western Screech-Owl	100	1
Pygmy Nuthatch	100	4	Sharp-shinned Hawk	100	10	Western Scrub-Jay	100	8
Pyrrhuloxia	100	2	Sharp-tailed Grouse	100	2	Western Tanager	100	7
Red Crossbill	100	12	Short-billed Dowitcher	100	3	Western Wood-Pewee	100	3
Red Knot	100	7	Short-eared Owl	100	1	Whimbrel	100	1
Red Phalarope	100	0	Snail Kite	100	2	White Ibis	100	3
Red-bellied Woodpecker	100	12	Snow Bunting	100	4	White-breasted Nuthatch	100	10
Red-breasted Merganser	100	2	Snow Goose	87	16	White-crowned Sparrow	100	7
Red-breasted Nuthatch	100	9	Snowy Egret	100	3	White-eyed Vireo	100	3
Red-breasted Sapsucker	100	2	Snowy Owl	100	8	White-faced Ibis	100	3
Red-cockaded Woodpecker	100	1	Snowy Plover	100	2	White-headed Woodpecker	100	2
Red-eyed Vireo	100	5	Solitary Sandpiper	100	3	White-rumped Sandpiper	100	3
Red-headed Woodpecker	100	6	Song Sparrow	100	9	White-tailed Hawk	100	2
Red-naped Sapsucker	100	1	Sooty Grouse	100	0	White-tailed Kite	100	1
Red-necked Grebe	100	2	Sora	100	2	White-tailed Ptarmigan	100	4
Red-necked Phalarope	100	0	Spotted Owl	100	1	White-throated Sparrow	100	8
Red-shouldered Hawk	100	12	Spotted Sandpiper	100	9	White-throated Swift	69	2
Red-tailed Hawk	100	21	Spotted Towhee	100	10	White-winged Crossbill	100	3
Red-throated Loon	100	2	Spruce Grouse	100	2	White-winged Dove	100	5
Red-winged Blackbird	100	10	Steller's Jay	100	4	White-winged Scoter	100	1
Reddish Egret	100	3	Stilt Sandpiper	100	2	Wild Turkey	99	8
Redhead	100	2	Summer Tanager	100	6	Willet	100	12
Ring-billed Gull	100	10	Surf Scoter	100	3	Williamson's Sapsucker	100	2
Ring-necked Duck	100	9	Surfbird	100	2	Willow Flycatcher	100	1
Ring-necked Pheasant	100	6	Swainson's Hawk	100	4	Willow Ptarmigan	100	4
Rock Pigeon	100	10	Swainson's Thrush	100	3	Wilson's Phalarope	100	0
Rock Ptarmigan	100	3	Swallow-tailed Kite	100	1	Wilson's Plover	100	2
Rock Sandpiper	96	3	Swamp Sparrow	100	3	Wilson's Snipe	100	3
Rock Wren	100	1	Tennessee Warbler	100	2	Wilson's Warbler	100	3
Rose-breasted Grosbeak	100	6	Thayer's Gull	100	2	Winter Wren	100	1
Roseate Tern	100	3	Townsend's Solitaire	100	1	Wood Stork	100	3
Ross's Goose	100	1	Townsend's Warbler	100	3	Wood Thrush	100	4
Rough-legged Hawk	100	1	Tree Swallow	100	9	Worm-eating Warbler	100	1
Royal Tern	100	2	Tricolored Heron	100	2	Wrentit	100	2
Ruby-crowned Kinglet	100	7	Tropical Kingbird	100	1	Yellow Warbler	100	8
Ruby-throated Hummingbird	100	9	Trumpeter Swan	100	2	Yellow-bellied Flycatcher	100	5
Ruddy Duck	100	8	Tufted Titmouse	100	6	Yellow-bellied Sapsucker	100	6
Ruddy Turnstone	100	2	Tundra Swan	100	2	Yellow-billed Cuckoo	100	3
Ruffed Grouse	100	2	Turkey Vulture	100	9	Yellow-billed Magpie	100	1
Rufous Hummingbird	100	10	Upland Sandpiper	100	1	Yellow-breasted Chat	100	2
Rufous-crowned Sparrow	100	0	Varied Thrush	100	4	Yellow-crowned Night-Heron	100	8
Rusty Blackbird	100	6	Veery	100	1	Yellow-eyed Junco	100	0
Sage Thrasher	100	1	Verdin	100	2	Yellow-headed Blackbird	100	8
Saltmarsh Sparrow	100	1	Vermilion Flycatcher	100	2	Yellow-rumped Warbler	100	14
Sanderling	100	6	Vesper Sparrow	100	1	Yellow-throated Vireo	100	3
Sandhill Crane	100	2	Violet-green Swallow	100	3	Yellow-throated Warbler	100	1
Sandwich Tern	100	5	Virginia Rail	100	1	Zone-tailed Hawk	100	1
Say's Phoebe	100	4	Wandering Tattler	100	2			

Species of the Birdsnap dataset, with image and category counts. Part 3 of 3.