UNIVERSITY OF CALIFORNIA, SAN DIEGO

Leveraging Human Perception and Computer Vision Algorithms for Interactive Fine-Grained Visual Categorization

A dissertation submitted in partial satisfaction of the
requirements for the degree of Doctor of Philosophy

in

Computer Science

by

Catherine Lih-Lian Wah

Committee in charge:

Professor Serge Belongie, Chair
Professor David Kriegman
Professor Gert Lanckriet
Professor Lawrence Saul
Professor Nuno Vasconcelos

2014

The Dissertation of Catherine Lih-Lian Wah is approved and is acceptable in quality and form for publication on microfilm and electronically:

Chair

University of California, San Diego

2014

DEDICATION

To my parents and twin sister, who continue to support and inspire me everyday.

EPIGRAPH

Do not fear going forward slowly; fear only to stand still.

Chinese proverb

Nothing is difficult, only unfamiliar.

Unknown

GOD: I own you like I own the caves.
THE OCEAN: Not a chance. No comparison.
GOD: I made you. I could tame you.
THE OCEAN: At one time, maybe. But not now.
GOD: I will come to you, freeze you, break you.
THE OCEAN: I will spread myself like wings. I am a billion tiny feathers. You have no idea what's happened to me.

Dave Eggers

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENTS

First and foremost, I wish to thank my parents Benjamin and Christine for their continued love, support, and guidance throughout graduate school and my life. I cannot adequately express my gratitude and deep appreciation for all the sacrifices they have made to help me get to where I am today. My father epitomizes what can be accomplished through diligence and hard work, and I strive each day to live up to his example. I am incredibly lucky to have the opportunity to learn from his insight and experience about the research process. My mother is the most selfless person I know. Even when half a world away, she is always willing to provide a sympathetic ear. I am continually inspired by her strength and courage; she is my personal hero. My parents' steadfast confidence in my abilities has seen me through countless obstacles and setbacks. Their encouragement has motivated me to undertake every endeavor with optimism. They have nurtured my interests in computer science and engineering from an early age, and their unwavering support has been key in the completion of my doctoral studies.

Second, I would like to thank my twin sister Elaine, who is and always has been my best friend (since birth), my wombmate, and my better half. Through her kindness, compassion, and generosity, she continues to inspire me to be a better version of myself.

This academic journey would not have been possible without the guidance of my advisor, Serge Belongie, who is truly dedicated to his students and always has their best interests at heart. I am deeply grateful for his tremendous patience, understanding, and constant support through the years, especially as I pursued multiple internship opportunities. It is a true honor and pleasure to work with someone as inspiring a scientist as Serge.

I am fortunate to have collaborated with some exceptional researchers over the years. I am thankful to Subhransu Maji for taking me on as an intern, as he has been an excellent collaborator and mentor. Working with him has been an extremely fruitful and

rewarding experience that has been vital to my dissertation research work. I am grateful to have had the privilege of working with and learning from Pietro Perona, who is truly impressive in his research vision and cares deeply about his students. I am also indebted to my undergraduate advisor Thomas Huang, who generously welcomed me into his research lab as an college freshman, as well as Dennis Lin, who as a graduate student took me under his wing; my undergraduate research experience was heavily influential in my decision to pursue a PhD in computer vision.

I also wish to express my gratitude to David Kriegman, Gert Lanckriet, Lawrence Saul, and Nuno Vasconcelos for serving on my thesis committee and for their valuable feedback and support.

I am especially thankful to Steve Branson, who has been a phenomenal mentor and from whom I have learned an incredible amount about the research process. Collaborating with him has been profoundly influential on the course of my graduate career, and I continue to look up to him as a role model. I am thankful to Boris Babenko, for being a mentor to me in both research and in life, and Kai Wang, for always giving me insightful research and career advice. I would also like to thank Marti Motoyama, for his friendship and for always motivating me with his exemplary work ethic.

Collaborating with the Visipedia team has been hugely instrumental in shaping my research interests and ideas. I am profoundly thankful to them for the countless discussions and crucial feedback: Grant Van Horn, Ron Appel, Peter Welinder, Florian Schroff, and Ryan Farrell, among others.

In addition, I am thankful to the past and present members of the UCSD Computer Vision Lab, whose insights and moral support have been critical to my achievements: Sam Kwak, Tsung-Yi Lin, Oscar Beijbom, Vincent Rabaud, Piotr Dollar, Carolina Galleguillos, Ana Cristina Murillo, Hani Altwaijry, Mohammad Moghimi, Zachary Murez, Phuc Nguyen, Arturo Flores, Eric Christiansen, Jan Jakeš, Tomas Matera, and

Michael Wilber, among others.

Finally, I would like to thank the current students and alumni who have made the past years in graduate school one of the most memorable and enriching times of my life: Ryan Braud, Neha Chachra, Chris Kanich, Ryan Kanoknukulchai, Alan Leung, Greg Long, Dan Moeller, Nima Nikzad, Laura Pina, Alex Tsiatas, Nakul Verma, Ming Wang, and Tess Winlock, among others.

Portions of this dissertation are based on papers I have co-authored with others, and my contributions to each are listed below as follows:

- Chapters 2 and 3, in part, are based on the material as it appears in *International Journal of Computer Vision*, 2014 as "The Ignorant Led by the Blind: A Hybrid Human-Machine Vision System for Fine-Grained Categorization" by S. Branson, G. Van Horn, C. Wah, P. Perona, and S. Belongie [12]. The dissertation author contributed to algorithm development, implemented code and experiments, and contributed to the writing of the paper.

- Chapter 3, in part, is based on the material as it appears in *European Conference on Computer Vision*, 2010 as "Visual Recognition with Humans in the Loop" by S. Branson, C. Wah, B. Babenko, F. Schroff, P. Welinder, P. Perona, and S. Belongie [14]. The dissertation author implemented parts of the code and experiments, and contributed to the writing of the paper.

- Chapter 3, in part, is based on the material as it appears in *International Conference on Computer Vision*, 2011 as "Multiclass Recognition and Localization with Humans in the Loop" by C. Wah, S. Branson, P. Perona, and S. Belongie [112]. The dissertation author contributed to algorithm development, implemented code and experiments, and contributed to the writing of the paper.

- Chapter 4, in part, is based on the material as it will appear in *Conference on*

Computer Vision and Pattern Recognition, 2014 as "Similarity Comparisons for Interactive Fine-Grained Categorization" by C. Wah, G. Van Horn, S. Branson, S. Maji, P. Perona, S. Belongie [114]. The dissertation author developed the algorithm and experiments, and wrote most of the paper.

- Chapter 4, in part, is based on material that has been submitted for publication, as it may appear in *British Machine Vision Conference*, 2014 as "Learning Localized Perceptual Similarity Metrics for Interactive Categorization" by C. Wah, S. Maji, and S. Belongie. The dissertation author developed the algorithm and experiments, and wrote most of the paper.

VITA

2008	B.S. in Electrical Engineering, University of Illinois at Urbana-Champaign
2011	M.S. in Computer Science, University of California, San Diego
2014	Ph.D. in Computer Science, University of California, San Diego

PUBLICATIONS

Steve Branson, Catherine Wah, Florian Schroff, Peter Welinder, Boris Babenko, Pietro Perona, and Serge Belongie, "Visual Recognition With Humans in the Loop," In *European Conference on Computer Vision (ECCV)*, 2010.

Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona, "Caltech-UCSD Birds 200," In *California Institute of Technology Tech Report*, 2010.

Catherine Wah, Steve Branson, Pietro Perona, and Serge Belongie, "Multiclass Recognition and Part Localization with Humans in the Loop," In *International Conference on Computer Vision (ICCV)*, 2011.

Catherine Wah, Steve Branson, Pietro Perona, and Serge Belongie, "Interactive Localization and Recognition of Fine-Grained Visual Categories," In *First Workshop on Fine-Grained Visual Categorization (CVPR Workshop)*, 2011.

Catherine Wah, Steve Branson, Pietro Perona, and Serge Belongie, "Birds-200: A Dataset For Fine-Grained Visual Categorization," In *First Workshop on Fine-Grained Visual Categorization (CVPR Workshop)*, 2011.

Catherine Wah, Steve Branson, Peter Welinder, Serge Belongie, and Pietro Perona, "The Caltech-UCSD Birds-200-2011 Dataset," In *California Institute of Technology Tech Report*, 2012.

Catherine Wah and Serge Belongie, "Attribute-Based Detection of Unfamiliar Classes with Humans in the Loop," In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

Catherine Wah and Serge Belongie, "Attribute-Based Detection of Unfamiliar Classes with Humans in the Loop," In *Second Workshop on Fine-Grained Visual Categorization (CVPR Workshop)*, 2013.

Wei Di, Catherine Wah, Anurag Bhardwaj, Robinson Piramuthu, and Neel Sundaresan, "Style Finder: Fine-Grained Clothing Style Recognition and Retrieval," In *Third IEEE International Workshop on Mobile Vision (CVPR Workshop)*, 2013.

Steve Branson, Grant Van Horn, Catherine Wah, Pietro Perona, and Serge Belongie, "The Ignorant Led by the Blind: A Hybrid Human-Machine Vision System for Fine-Grained Categorization," In *International Journal of Computer Vision (IJCV)*, 2014.

Catherine Wah, Grant Van Horn, Steve Branson, Subhransu Maji, Pietro Perona, and Serge Belongie, "Similarity Comparisons for Interactive Fine-Grained Categorization," To appear in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

Catherine Wah, Grant Van Horn, Steve Branson, Subhransu Maji, Pietro Perona, and Serge Belongie, "Similarity Comparisons for Interactive Fine-Grained Categorization," To appear in *Workshop on Computer Vision and Human Computation (CVPR Workshop)*, 2014.

Catherine Wah, Subhransu Maji, and Serge Belongie, "Learning Localized Perceptual Similarity Metrics for Interactive Categorization," Submitted to *British Machine Vision Conference*, 2014.

ABSTRACT OF THE DISSERTATION

Leveraging Human Perception and Computer Vision Algorithms for Interactive Fine-Grained Visual Categorization

by

Catherine Lih-Lian Wah

Doctor of Philosophy in Computer Science

University of California, San Diego, 2014

Professor Serge Belongie, Chair

Fine-grained categorization has emerged in recent years as a problem of great interest to the computer vision community, given its wide range of applications including species identification for animals, plants, and insects, as well as classification of man-made objects such as vehicle makes and models and architectural styles. The goal of fine-grained categorization is to distinguish between subcategories (*e.g.*, Pembroke Welsh Corgi, Shiba Inu) that belong to the same entry-level category (*e.g.*, Dog). As fine-grained categories are often visually similar, a general-purpose computer vision algorithm for basic-level category recognition is often ineffective in the fine-grained case.

Moreover, fine-grained categories are typically recognizable only by experts (*e.g.*, the average person cannot recognize a Myrtle Warbler, a species of bird), while a layperson can immediately recognize entry-level categories like motorcycles or cats.

While fine-grained categorization is difficult for both humans and machines, we combine their respective strengths to create an effective human-in-the-loop classification system. These types of systems integrate machine vision algorithms with user feedback at test time in order to interactively arrive at the correct answer. Incorporating user input drives up recognition accuracy to levels sufficient for practical applications; at the same time, computer vision reduces the amount of human interaction required. Moreover, we are able to incrementally improve our models and algorithms while providing a useful service to users.

In this dissertation, we explore two paradigms for interactive categorization. The first relies on a comprehensive vocabulary of semantic parts and attributes to discriminate categories. A bird species recognition system, for example, may request feedback from the user regarding a particular image, such as "Where is the beak?" or "Is the wing blue?" Semantic vocabulary-based methods, however, present certain challenges in terms of scalability and finding experts with necessary domain knowledge, as experts can be a scarce resource. The second paradigm we present eliminates the need for such a vocabulary; instead, it is based on perceptual similarity metrics learned from human-provided similarity comparisons. By leveraging these continuous embedded similarity spaces, we exploit a vastly more powerful representation that can be readily applied to other basic-level categories.

Chapter 1

Introduction

Fine-grained categorization, also known as subordinate categorization in the psychology literature [85, 68, 10], has emerged in recent years as a problem of great interest to the computer vision community, with applications including species identification for animals [112, 61, 49], plants [52], flowers [71] and insects [55] as well as classification of man-made objects such as vehicle makes and models [94] and architectural styles [63]. Fine-grained visual categories lie in the space between basic (or entry) level categories [86] (*e.g.*, the 20 classes from PASCAL-VOC [29] including motorbikes, dining tables, etc.) and identification of individuals (e.g., face or fingerprint biometrics). As the visual distinctions among fine-grained categories are often quite subtle, a given general-purpose tool popular for basic-level category recognition can be rendered a rather blunt instrument in the fine-grained case.

Fine-grained categories are usually recognized only by experts (*e.g.*, the average person cannot recognize a Myrtle Warbler), while a layperson can recognize entry level categories like bicycles or sheep immediately. This work arises from a key realization: while fine-grained visual categorization is difficult for both humans and machines, humans and machines have radically different strengths and weaknesses. A visual interactive categorization system composed of a human and a machine can carry out the task, and do so efficiently, by combining the strength of each; this requires a dynamic

collaboration between the two agents. Humans are able to detect and broadly categorize objects, even when they do not recognize them. They can localize basic shapes and parts, and recognize colors and materials. Human errors arise primarily because people have (1) limited experiences and memory and (2) subjective and perceptual differences. In contrast, computers can run deterministic software and aggregate large databases of information. They excel at memory intensive problems like recognizing movie posters or cereal boxes but struggle with objects that are textureless, immersed in clutter, highly articulated or non-trivially deformed.

My dissertation focuses on exploring the different methods by which can we incorporate human interaction into an interactive categorization system. Incorporating user input drives up recognition accuracy to levels sufficient for practical applications; at the same time, computer vision reduces the amount of human interaction required. Moreover, we are able to incrementally improve our models and algorithms while providing a useful service to users. To make the scope of this research problem reasonable, we focus our attention and experiments on a single entry-level category, birds (see Appendix A for more details on the dataset that we collected).

In Chapter 2, I describe Visipedia, which is the motivating application of this work, and I provide an overview of relevant work in related areas of research.

In Chapter 3, I present the first of two novel paradigms for interactive categorization that relies on a comprehensive vocabulary of semantic parts and attributes to discriminate categories. The classification method can be seen as a visual version of the *20 Questions Game*, where questions based on simple visual attributes are posed interactively. Our models and algorithms for object detection, part localization, and category recognition scale efficiently to large numbers of categories. In addition, we evaluate the usefulness of different types of human input and take into account varying levels of human error, time spent and informativeness in a multiclass or multitask setting,

and combine computer vision algorithms, forms of user input and question selection techniques in an integrated framework.

In Chapter 4, I present a second paradigm that is a departure from the previous expert-driven and attribute-centric approach; instead, we rely on relative similarity comparisons provided by users, incorporating computer vision and learned perceptual similarity metrics in a unified framework. We also observe how localization of these similarity metrics improves classification performance. At test time, users are asked to judge relative similarity between a query image and various sets of images; these general queries do not require expert-defined terminology and are applicable to other domains and basic-level categories, enabling a flexible, efficient, and scalable system for fine-grained categorization with humans in the loop. By leveraging these continuous embedded similarity spaces, we exploit a vastly more powerful representation that can be readily applied to other basic-level categories.

Acknowledgements

Parts of this section are based on the paper “The Ignorant Led by the Blind: A Hybrid Human-Machine Vision System for Fine-Grained Categorization” by S. Branson, G. Van Horn, C. Wah, P. Perona, and S. Belongie [12]. The dissertation author contributed to algorithm development, implemented code and experiments, and contributed to the writing of the paper.

Chapter 2

Background and Related Work

2.1 Visipedia

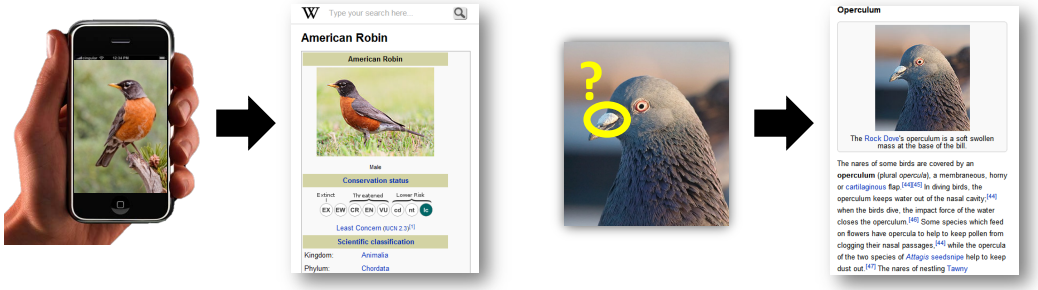
The motivating application of this work is Visipedia [81]; short for "Visual Encyclopedia," it is a user-generated knowledge base of visual objects, where visual and semantic concepts can be linked. As an augmented version of Wikipedia, it enables us to provide services that Wikipedia in its current incarnation lacks, such as improved text-to-image search and image-to-article visual search (see Figure 2.1).

The goals of Visipedia include the creation of hyperlinked, interactive images embedded in Wikipedia articles, scalable representations of visual knowledge, large-scale machine vision datasets, and visual search capabilities. Visipedia, like Wikipedia, is built on and relies on fine-grained visual categories. We discuss in Section 2.2 various areas of research that are relevant to realizing Visipedia.

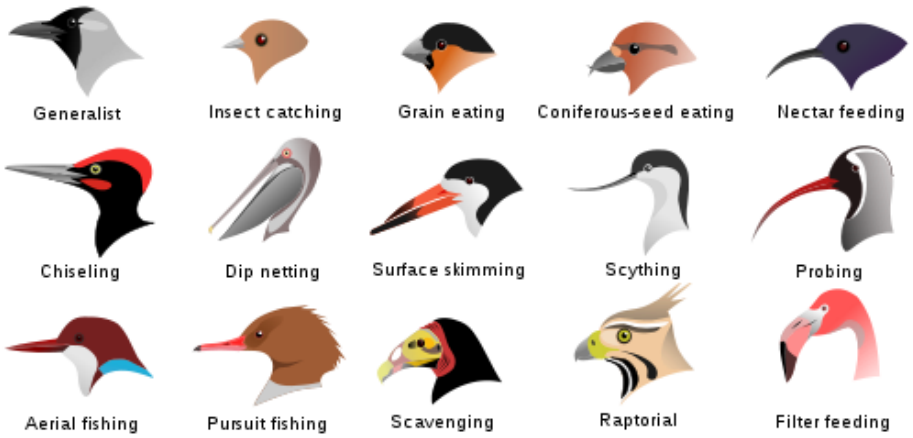
2.2 Related Work

2.2.1 Fine-Grained Categorization

Fine-grained visual categorization (FGVC) is a challenging problem that has recently become a popular topic in computer vision. Applications include recognizing different species of leaves [52, 5], flowers [70, 71], dogs [80, 61, 79, 49], birds [14, 34,



(a) Visipedia



(b)

Figure 2.1. Visipedia. 2.1(a): Short for "Visual Encyclopedia," Visipedia is an augmented version of Wikipedia, where pictures are first-class citizens alongisde text, and it enables us to provide services such as improved text-to-image search and image-to-article visual search. 2.1(b): Visipedia, like Wikipedia, is built on and relies on fine-grained visual categories.

112, 125, 57], and stonefly larvae [64, 55]. Each of these can be seen as interesting scientific applications with a significant appeal to a specific demographic of users, enthusiasts, or citizen scientists. In conjunction with this, many new FGVC datasets have emerged with richer annotations, such as CUB-200-2011 [113] (birds with parts and attributes), Columbia Dogs With Parts, Leeds Butterflies [117] (segmentations and text descriptions), Oxford-IIIT Pets (cats and dogs with segmentations and bounding boxes), and Stanford Dogs (bounding boxes).

Most research in FGVC is related to finding less lossy features, models, or representations to deal with tightly related categories. The work of Yao et al. [124, 123] and Martinez et al. [64] relates to learning features that go beyond traditional codebook-based methods in object recognition. Nilsback et al. [71] and Chai et al. [15, 17] introduce techniques that improve ROI for feature extraction by simultaneously segmenting and recognizing FGVCs. Other methods focus on incorporating part/pose detectors that supplant or augment bag-of-words methods by allowing for more strongly localized visual features [34, 112, 80, 125, 61, 79]. Most of these methods exploit new types of annotation. The work of Farrell et al. [34, 125] explores different methods for pose normalization using Poselets, including an original method that is based on 3D volumetric primitives.

2.2.2 Human-In-The-Loop Methods

An interactive algorithm that assists a human in discovering the true class is useful and preferable to a fully automatic yet error-prone algorithm. Human-in-the-loop methods have recently experienced a strong resurgence in popularity. Parikh et al. introduced an innovative human debugging framework [77, 76], using human experiments to help diagnose bottlenecks in computer vision research. This work is similar in spirit to our work in that it involves comparing the visual capabilities of humans and computers.

A number of exciting active learning algorithms that incorporate new types of human interactivity have come about in recent years [107, 108, 24, 110, 91, 78, 13]. Our work is related to this area; however, it is different in the sense that it pertains to active classification (*e.g.*, incorporating similar types of interactive feedback at classification time instead of during learning).

Interactive methods for generating vocabularies of parts or attributes [62, 74, 25], incorporating annotator rationales [24], relevance feedback methods [38, 127], and runtime interactive computer vision systems [121, 87, 59, 67, 111] are all interesting related lines of research. The main distinguishing feature of our work is the integration of modern object recognition techniques with interactivity at test time, and developing this area in more depth than prior work. This includes integrating interactive algorithms with multiclass recognition techniques, part-based methods, attribute-based methods, and similarity-based methods.

2.2.3 Active Classification

Our methodology for selecting which questions to pose to human users is an instance of active testing [42, 41, 99, 3], where a sequence of questions are chosen at runtime to minimize as much uncertainty as possible about some prediction task (*e.g.*, consider the Twenty Questions Game). Similar to decision trees [83], the criterion for choosing the next question is information theoretic; however, unlike decision trees, questions are chosen on-the-fly at runtime—precomputed decision trees would be intractably large (*i.e.*, due to an excessively large branching factor or depth as a result of more complex sources of information).

This relates to the area of expert systems [69, 9], which are used for applications such as medical diagnosis, accounting, process control, and software troubleshooting. Expert systems attempt to answer a problem that could normally only be solved by one

or more experts, and involve construction of a knowledge base and inference rules that can be used to synthesize a set of steps or input queries dynamically. Our approach can be seen as an application of expert systems to object recognition, with the key addition that we are able to use the observed image pixels as an additional source of information. Computationally, our method also has similarities to algorithms based on information gain, entropy calculation, and decision trees [98, 26, 48, 83].

Active testing has been applied to computer vision to speedup object localization and tracking problems [42, 41, 96, 95], where the active testing system sequentially chooses locations to evaluate a detector (rather than brute force evaluate a sliding window detector), interactively refining its belief of where the object is located. The main difference between these methods and ours is the use of a hybrid model where computer vision estimates are augmented with questions that are posed interactively to humans (as opposed to a computer).

Ferecatu et al. [38, 39, 31] applied active testing to image retrieval with relevance feedback, developing a system that intelligently selects similarity questions to pose to human users. The main difference between this approach and ours is the incorporation of computer vision at runtime (*i.e.*, [38] considers the "mental matching" problem where no image is present at runtime).

Acknowledgements

This chapter is based on material from "The Ignorant Led by the Blind: A Hybrid Human-Machine Vision System for Fine-Grained Categorization" by S. Branson, G. Van Horn, C. Wah, P. Perona, and S. Belongie [12]. The dissertation author contributed to algorithm development, implemented code and experiments, and contributed to the writing of the paper.

Chapter 3

Interactive Categorization with Parts and Attributes

3.1 Introduction

We present in this chapter an interactive, hybrid human-computer method for object classification. The method applies to classes of problems that are difficult for most people, but are recognizable by people with the appropriate expertise (*e.g.*, animal species or airplane model recognition). The classification method can be seen as a visual version of the *20 Questions Game*, where questions based on simple visual attributes are posed interactively. The goal is to identify the true class while minimizing the number of questions asked, using the visual content of the image. Incorporating user input drives up recognition accuracy to levels that are good enough for practical applications; at the same time, computer vision reduces the amount of human interaction required. The resulting hybrid system is able to handle difficult, large multi-class problems with tightly-related categories.

We introduce a general framework for incorporating almost any off-the-shelf multiclass object recognition algorithm into the visual 20 questions game, and provide methodologies to account for imperfect user responses and unreliable computer vision algorithms. We evaluate the accuracy and computational properties of different computer

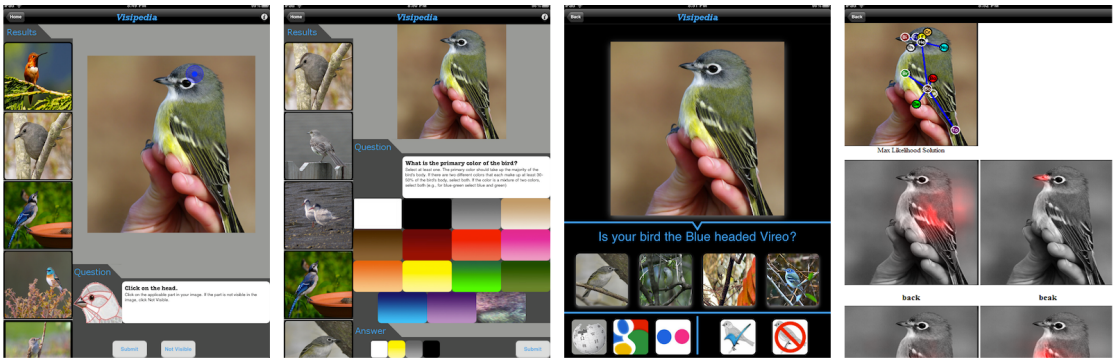


Figure 3.1. Screen capture of an iPad app for bird species recognition. A user takes a picture of a bird she wants to recognize, and it is uploaded to a server. The server runs computer vision algorithms to localize parts of the bird and predict bird species (debugging output of the algorithms is shown in the image on the lower right). The computer system intelligently selects a series of questions to ask (*click on the head*, *what is the primary color of the bird?*) that are designed to reduce its ambiguity about the predicted bird species as quickly as possible.

vision algorithms and the effects of noisy user responses on a dataset of 200 bird species and on the Animals With Attributes dataset. Our results demonstrate the effectiveness and practicality of the hybrid human-computer classification paradigm. A real-life application of bird species recognition is shown in Figure 3.1.

3.2 Related Work

Methods based on parts [36, 37, 11, 72, 122] and attributes [33, 54, 53, 32, 116, 75] have both become popular, mainstream topics in computer vision research. An interesting component of FGVC problems is that similarities between classes are exploitable for transfer learning or model sharing methods (*e.g.*, different bird species share the same types of parts and attributes). FGVC methods that incorporate a super-category detection model [34, 112, 80, 125, 61, 79] (*e.g.*, running a universal bird detector before a species classifier) implicitly use a form of part sharing. Similarly, many attribute-based methods [54, 53, 32] are motivated as a mechanism for model sharing.

An equally important motivation for parts and attributes is that they allow richer

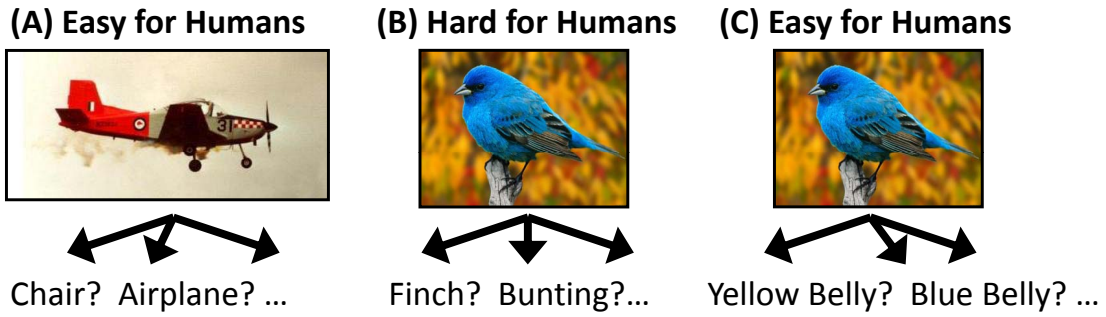


Figure 3.2. Examples of classification problems that are easy or hard for humans. While basic-level category recognition (left) and recognition of low-level visual attributes (right) are easy for humans, most people struggle with finer-grained categories (middle). By defining categories in terms of low-level visual properties, hard classification problems can be turned into a sequence of easy ones.

types of communication between humans and computers [75, 78, 33].

3.3 Visual Recognition with Humans in the Loop

Multi-class object recognition is a widely studied field in computer vision that has undergone rapid change and progress over the last decade. These advances have largely focused on types of object categories that are easy for humans to recognize, such as motorbikes, chairs, horses, bottles, *etc*. Finer-grained categories, such as specific types of motorbikes, chairs, or horses are more difficult for humans and have received comparatively little attention. One could argue that object recognition as a field is simply not mature enough to tackle these types of finer-grained categories. Performance on basic-level categories is still lower than what people would consider acceptable for practical applications (state-of-the-art accuracy on Caltech-256[45] is $\approx 45\%$, and the winner of the 2009 VOC detection challenge [28] achieved only $\approx 28\%$ average precision). Moreover, the number of object categories in most object recognition datasets is still fairly low, and increasing the number of categories further is usually detrimental to performance [45].

Figure 3.3. Examples of the visual 20 questions game on the 200 class Bird dataset. Human responses (shown in red) to questions posed by the computer (shown in blue) are used to drive up recognition accuracy. In the left image, computer vision algorithms can guess the bird species correctly without any user interaction. In the middle image, computer vision reduces the number of questions to 2. In the right image, computer vision provides little help.

On the other hand, recognition of finer-grained categories is an important problem to study—it can help people recognize types of objects they don't yet know how to identify. We believe a hybrid human-computer recognition method is a practical intermediate solution toward applying contemporary computer vision algorithms to these types of problems. Rather than trying to solve object recognition entirely, we take on the objective of minimizing the amount of human labor required. As research in object recognition progresses, tasks will become increasingly automated, until eventually we will no longer need humans in the loop. This approach differs from some of the prevailing ways in which people approach research in computer vision, where researchers begin with simpler and less realistic datasets and progressively make them more difficult and realistic as computer vision improves (*e.g.*, Caltech-4 $\rightarrow$ Caltech-101 $\rightarrow$ Caltech-256). The advantage of the human-computer paradigm is that we can provide usable services to people in the interim-period where computer vision is still unsolved. This may help increase demand for computer vision, spur data collection, and provide solutions for the types of problems people outside the field want solved.

Our goal is to provide a simple framework that makes it as effortless as possible for researchers to plug their existing algorithms into the human-computer framework and use humans to drive up performance to levels that are good enough for real-life

applications. Implicit to our model is the assumption that lay-people generally cannot recognize finer-grained categories (*e.g.*, Myrtle Warbler, Thruxton Jackaroo, *etc.*) due to imperfect memory or limited experiences; however, they do have the fundamental visual capabilities to recognize the parts and attributes that collectively make recognition possible (see Fig. 3.2). By contrast, computers lack many of the fundamental visual capabilities that humans have, but have perfect memory and are able to pool knowledge collected from large groups of people. Users interact with our system by answering simple yes/no or multiple choice questions about an image or object, as shown in Fig. 3.3. Similar to the *20-Questions Game*[1], we observe that the number of questions needed to classify an object from a database of C classes is usually $O(\log C)$ (when user responses are accurate), and can be faster when computer vision is in the loop. Our method of choosing the next question to ask uses an information gain criterion and can deal with noisy (probabilistic) user responses. We show that it is easy to incorporate any computer vision algorithm that can be made to produce a probabilistic output over object classes.

Our experiments in this paper focus on bird species categorization, which we take to be a representative example of recognition of tightly-related categories. The bird dataset contains 200 bird species and over 6,000 images [120]. We believe that the same types of methodologies used for birds will apply to other object domains.

In Section 3.3.1, we define the hybrid human-computer problem and basic algorithm, which includes methodologies for modeling noisy user responses and incorporating computer vision into the framework. We describe an extension to part-based models in Section 3.4, our datasets and implementation details in Section 3.5, and present empirical results in Section 3.6.

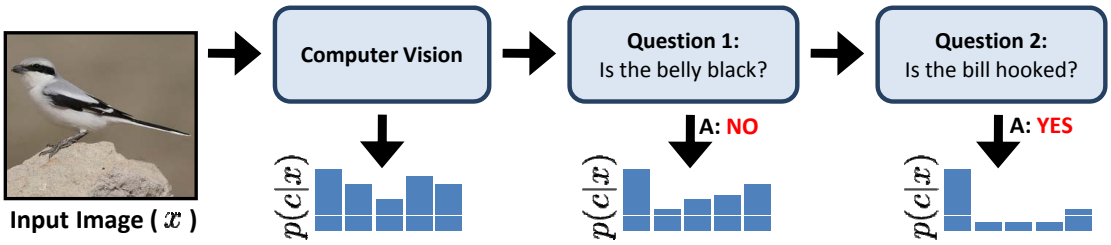


Figure 3.4. Visualization of the basic algorithm flow. The system poses questions to the user, which along with computer vision, incrementally refine the probability distribution over classes.

3.3.1 Algorithms and Framework

Given an image x, our goal is to determine the true object class $c \in \{1...C\}$ by posing questions based on visual properties that are easy for the user to answer (see Fig. 3.2). At each step, we aim to exploit the visual content of the image and the current history of question responses to intelligently select the next question. The basic algorithm flow is summarized in Fig. 3.4.

Let $\mathscr{Q} = \{q_1...q_n\}$ be a set of random variables corresponding to all possible questions (*e.g.*, IsRed?, HasStripes?, *etc.*), and $\mathscr{A}$ be the set of possible answers[2], such that $q_i \in \mathscr{A}$. We can also ask users to select a confidence value for each question; let r_i be a random variable corresponding to the confidence reported for question i, and $\mathscr{V}$ be the set of possible confidence scores (*e.g.*, Guessing, Probably, Definitely), such that $r_i \in \mathscr{V}$. For convenience we define variables $u_i = (q_i, r_i)$; we will refer to these as "questions" from now and it should be clear from context when we mean question/confidence pairs.

Let $j \in \{1...n\}^T$ be an array of T indices to questions we will ask the user. $U^{t-1} = \{u_{j(1)}...u_{j(t-1)}\}$ is the set of questions asked by time step $t-1$. At time step t we would like to find the question $j(t)$, that maximizes the expected information gain. Information gain is widely used in decision trees (*e.g.* [83]) and can be computed from

[1] See for example http://20q.net.

[2] We model user answers as binary questions; extensions to multiple choice questions are readily available and may be desirable in a future version of the system.

Algorithm 1. Visual 20 Questions Game

1: $U^0 \leftarrow \emptyset$
2: **for** $t = 1$ to 20 **do**
3: $\quad j(t) = \max_k \mathrm{I}(c; u_k | x, U^{t-1})$
4: $\quad$ Ask user question $u_{j(t)}$, and $U^t \leftarrow U^{t-1} \cup u_{j(t)}$.
5: **end for**
6: Return class $c^* = \max_c p(c|x, U^t)$

an estimate of $p(c|x,U)$.

We define $\mathrm{I}(c;u|x,U)$, the expected information gain of posing the additional question u, as follows:

$$\mathrm{I}(c;u|x,U) = \mathbb{E}_u \left[\mathrm{KL}\left(p(c|x,u \cup U) \parallel p(c|x,U) \right) \right] \tag{3.1}$$

$$= \sum_{u \in \mathscr{A} \times \mathscr{V}} p(u|x,U) \left(\mathrm{H}(c|x,u \cup U) - \mathrm{H}(c|x,U) \right) \tag{3.2}$$

and $\mathrm{H}(c|x,U)$ is the entropy of $p(c|x,U)$

$$\mathrm{H}(c|x,U) = -\sum_{c=1}^{C} p(c|x,U) \log p(c|x,U) \tag{3.3}$$

The general algorithm for interactive object recognition is shown in Algorithm 1. In the next sections, we describe in greater detail methods for modeling user responses and different methods for incorporating computer vision algorithms, which correspond to different ways to estimate $p(c|x,U)$.

3.3.2 Incorporating Computer Vision

When no computer vision is involved it is possible to pre-compute a decision tree that defines which question to ask for every possible sequence of question answers. With computer vision in the loop, however, the best questions depend dynamically on the contents of the image.

In this section, we propose a simple framework for incorporating any multi-class object recognition algorithm that produces a probabilistic output over classes. We can compute the posterior as follows:

$$p(c|x,U) \propto p(U|c,x)p(c|x) = p(U|c)p(c|x) \tag{3.4}$$

Here we make an assumption that $p(U|c,x) = p(U|c)$; effectively this assumes that the types of noise or randomness that we see in user responses is class-dependent and not image-dependent. We can still accommodate for variation in user responses due to user error, subjectivity, external factors, and intraclass variance; however we throw away some image-related information (for example, we lose ability to model a change in the distribution of user responses as a result of a computer-vision-based estimate of object pose).

In terms of computation, we estimate $p(c|x)$ using a classifier trained offline (more details in Section 3.5). Upon receiving an image, we run the classifier once at the beginning of the process, and incrementally update $p(c|x,U)$ by gathering more answers to questions from the user. One could imagine a system where computer vision is invoked several times during the process; as categories are weeded out by answers, the system would use a more tuned classifier to update the estimate of $p(c|x)$. However, our preliminary experiments with such methods did not show an advantage. Note that when no computer vision is involved, we simply replace $p(c|x)$ with a prior $p(c)$.

3.3.3 Modeling User Responses

Recall that for each question we may also ask a corresponding confidence value from the user, which may be necessary when an attribute cannot be determined (for example, when the associated part(s) are not visible). We estimate the distribution $p(U|c)$

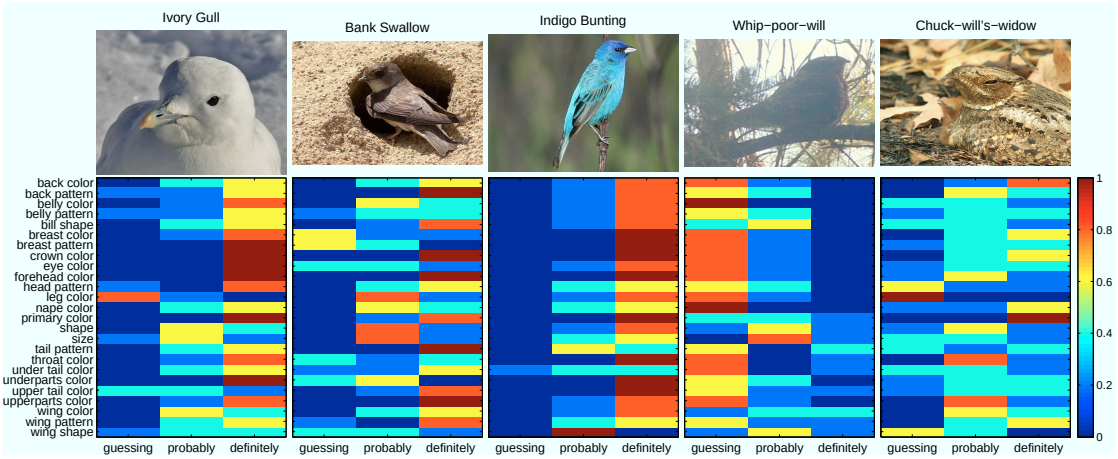


Figure 3.5. Examples of user responses for each of the 25 attributes. The distribution over $\{$*Guessing, Probably, Definitely*$\}$ is color coded with blue denoting 0% and red denoting 100% of the five answers per image attribute pair.

as follows:

$$p(U|c) = \prod_{i}^{t} p(u_i|c) \tag{3.5}$$

In the above we assume that the questions are answered independently given the category. If a single user is answering all of the questions, then this assumption might not hold (responses are correlated due to per-user subjectivity); however, in other variants of our application we may want to crowd source other users to answer these questions for the purpose of labeling many images (similar to ReCAPTCHA [109]), in which case answers would come from different people. It may also be possible to use a more sophisticated model in which we estimate a full joint distribution for $p(U|c)$; in our preliminary experiments this approach did not work well due to insufficient training data.

To compute $p(u_i|c) = p(q_i, r_i|c) = p(q_i|r_i, c)p(r_i|c)$, we assume that $p(r_i|c)$ is uniform. Next, we compute each $p(q_i|r_i, c)$ as the posterior of a multinomial distribution with Dirichlet prior $\mathrm{Dir}\big(\alpha_r p(q_i|r_i) + \alpha_c p(q_i|c)\big)$, where α_r and α_c are constants, $p(q_i|r_i)$ is a global attribute prior, and $p(q_i|c)$ is estimated by pooling together certainty labels. Incorporating prior terms is important in order to avoid over-fitting when the training

examples for any attribute-class pair are sparse. In practice, we use a larger prior term for *Guessing* than *Definitely*, $\alpha_{guess} > \alpha_{def}$, which effectively down weights the importance of any response with certainty level *Guessing*. In Figure 3.17(a), we present an example of an attribute question posed to the user.

3.4 Extension to Part-Based Models

Vision researchers have become increasingly interested in recognition of parts [11, 36, 118], attributes [33, 53, 54], and fine-grained categories (*e.g.* specific species of birds, flowers, or insects) [5, 14, 64, 71]. Beyond traditionally studied basic-level categories, these interests have led to progress in transfer learning and learning from fewer training examples [35, 36, 47, 71, 118], larger scale computer vision algorithms that share processing between tasks [71, 93], and new methodologies for data collection and annotation [11, 27].

Parts, attributes, and fine-grained categories push the limits of human expertise and are often inherently ambiguous concepts. For example, perception of the precise location of a particular part (such as a bird's beak) can vary from person to person, as does perception of whether or not an object is shiny.

Consider for example different types of human annotation tasks in the domain of bird species recognition. For the task "*Click on the beak*," the location a human user clicks is a noisy representation of the ground truth location of the beak. It may not in isolation solve any single recognition task; however, it provides information that is useful to a machine vision algorithm for localizing other parts of the bird, measuring attributes (*e.g.* cone-shaped), recognizing actions (*e.g.* eating or flying), and ultimately recognizing the bird species. The answer to the question "*Is the belly striped?*" similarly provides information towards recognizing a variety of bird species. Each type of annotation takes a different amount of human time to complete and provides varying amounts of

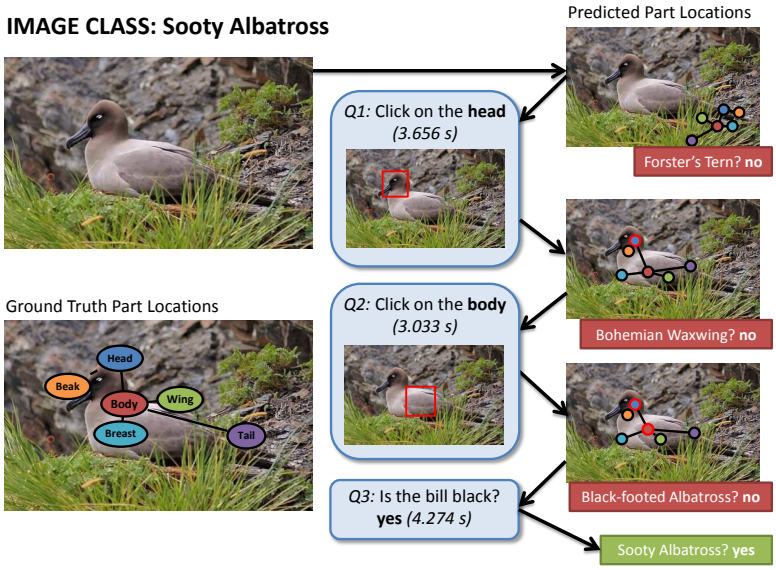


Figure 3.6. Interactive visual recognition with localization. Our system can query the user for input in the form of binary attribute questions or part clicks. In this illustrative example, the system provides an estimate for the pose and part locations of the object at each stage. Given a user-clicked location of a part, the probability distributions for locations of the other parts in each pose will adjust accordingly. The rightmost column depicts the maximum likelihood estimate for part locations.

information.

In this section, we discuss an extension to [14], with three key modifications:

- While [14] used non-localized computer vision methods based on bag-of-words features extracted from the entire image, we use localized part and attribute detectors. Thus [14] relied on experiments with test images cropped by ground truth bounding boxes; we are able to evaluate performance on uncropped images in unconstrained environments.

- Whereas [14] incorporated only one type of user input – binary questions pertaining to attributes – we allow heterogeneous forms of user input including user-clicked part locations. Users can click on any pixel location in an image, introducing significant algorithmic and computational challenges as we must reason over

hundreds of thousands of possible click point and part locations.

- Whereas [14] measured human effort in terms of the total number of questions asked, we introduce an extended question selection criterion that factors in the expected amount of human time needed to answer each type of question.

Our integrated approach builds on two areas in computer vision: part-based models and attribute-based learning, which have both been explored in depth in other works. Specifically, we use a part representation similar to a Felzenszwalb-style deformable part model [36, 37] (sliding window HOG-based part detectors fused with tree-structured spatial dependencies). Whereas most attribute-based methods [33, 54] use non-localized classifiers, [32, 116] incorporate object or part-level localization with attribute detectors. Our methods differ from earlier work on parts and attributes by (1) the specific combination of a Felzenszwalb-style deformable part model with localized attribute detectors, (2) the additional ability to combine part and attribute models with different types of user input, and (3) the deployment of such methods on a dataset of larger scale, localizing 200 object classes, 13 parts, 11 aspects, and 312 binary attributes in a fraction of a second.

We introduce models and algorithms for object detection, part localization, and category recognition that scale efficiently to large numbers of categories. Our algorithms can localize and classify objects on a 200-class dataset in a fraction of a second, using part and attribute detectors that are shared among classes. We introduce a formal model for evaluating the usefulness of different types of human input that takes into account varying levels of human error, time spent, and informativeness in a multiclass or multitask setting. We introduce fast algorithms that are able to predict the informativeness of 312 binary questions and 13 part click questions in a fraction of a second. All such computer vision algorithms, forms of user input, and question selection techniques are combined into an integrated framework. We present a thorough experimental comparison of a

number of methods for optimizing human input.

Algorithm Description

In this section, we introduce a principled framework for integrating part-based detectors, multi-class categorization algorithms, and different types of human feedback into a common probabilistic model. We also introduce efficient algorithms for inferring and updating object class and localization predictions as additional user input is obtained. We begin by formally defining the problem.

Given an image x, our goal is to predict an object class from a set of C possible classes (*e.g*. Myrtle Warbler, Blue Jay, Indigo Bunting) within a common basic-level category (*e.g*. Birds). We assume that the C classes fall within a reasonably homogeneous basic-level category such as birds that can be represented using a common vocabulary of P parts (*e.g*. head, belly, wing), and A attributes (*e.g*. cone-shaped beak, white belly, striped breast). We use a class-attribute model based on the direct-attribute model of Lampert et al. [54], where each class $c \in 1...C$ is represented using a unique, deterministic vector of attribute memberships $\mathbf{a}^c = [a_1^c...a_A^c]$, $a_i^c \in 0,1$. We extend this model to include part localized attributes, such that each attribute $a \in 1...A$ can optionally be associated with a part part$(a) \in 1...P$ (*e.g*. the attributes *white belly* and *striped belly* are both associated with the part belly). In this case, we express the set of all ground truth part locations for a particular object as $\Theta = \{\theta_1...\theta_P\}$, where the location θ_p of a particular part p is represented as an x_p, y_p image location, a scale s_p, and an aspect v_p (*e.g*. side view left, side view right, frontal view, not visible, *etc*.):

$$\theta_p = \{x_p, y_p, s_p, v_p\}. \tag{3.6}$$

Note that the special aspect *not visible* is used to handle parts that are occluded or self-occluded.

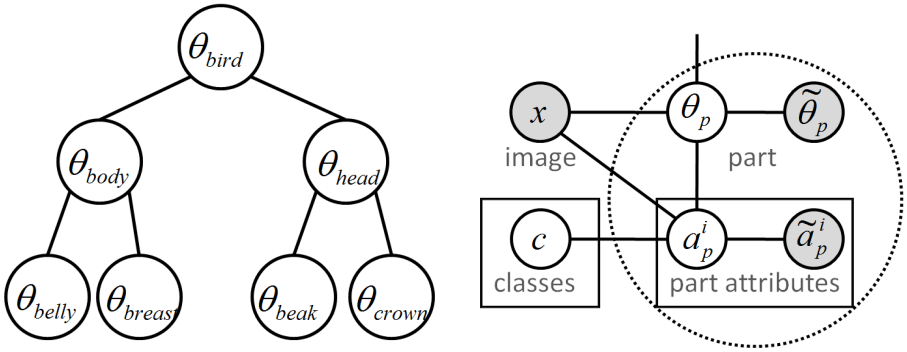
We can optionally combine our computer vision algorithms with human input, by intelligently querying user input at runtime. A human is capable of providing two types of user input which indirectly provide information relevant for predicting the object's class: mouse click locations $\tilde{\theta}_p$ and attribute question answers $\tilde{a}_i$. The random variable $\tilde{\theta}_p$ represents a user's input of the part location θ_p, which may differ from user to user due to both clicking inaccuracies and subjective differences in human perception (Figure 3.4). Similarly, $\tilde{a}_i$ is a random variable defining a user's perception of the attribute value a_i.

We assume a pool of $A+P$ possible questions that can be posed to a human user $\mathscr{Q} = \{q_1 ... q_A, q_{A+1} ... q_{A+P}\}$, where the first A questions query $\tilde{a}_i$ and the remaining P questions query $\tilde{\theta}_p$. Let $\mathscr{A}_j$ be the set of possible answers to question q_j. At each time step t, our algorithm considers the visual content of the image and the current history of question responses to estimate a distribution over the location of each part, predict the probability of each class, and intelligently select the next question to ask $q_{j(t)}$. A user provides the response $u_{j(t)}$ to a question $q_{j(t)}$, which is the value of $\tilde{\theta}_p$ or $\tilde{a}_i$ for part location or attribute questions, respectively. The set of all user responses up to timestep t is denoted by the symbol $U^t = \{u_{j(1)} ... u_{j(t)}\}$. We assume that the user is consistent in answering questions and therefore the same question is never asked twice.

Probabilistic Model

Our probabilistic model incorporating both computer vision and human user responses is summarized in Figure 3.7(b). Our goal is to estimate the probability of each class given an arbitrary collection of user responses U^t and observed image pixels x:

$$p(c|U^t, x) = \frac{p(\mathbf{a^c}, U^t|x)}{\sum_c p(\mathbf{a^c}, U^t|x)}, \tag{3.7}$$



(a) Part Model (b) Part-Attribute Model

Figure 3.7. Probabilistic Model. 3.7(a): The spatial relationship between parts has a hierarchical independence structure. 3.7(b): Our model employs attribute estimators, where part variables θ_p are connected using the hierarchical model shown in 3.7(a).

which follows from the assumption of unique, class-deterministic attribute memberships $\mathbf{a}^c$ [54]. We can incorporate localization information Θ into the model by integrating over all possible assignments to part locations

$$p(\mathbf{a}^c, U^t|x) = \int_{\Theta} p(\mathbf{a}^c, U^t, \Theta|x) d\Theta. \tag{3.8}$$

We can write out each component of Eq 3.8 as

$$p(\mathbf{a}^c, U^t, \Theta|x) = p(\mathbf{a}^c|\Theta, x) p(\Theta|x) p(U^t|\mathbf{a}^c, \Theta, x) \tag{3.9}$$

where $p(\mathbf{a}^c|\Theta, x)$ is the response of a set of attribute detectors evaluated at locations Θ, $p(\Theta|x)$ is the response of a part-based detector, and $p(U^t|\mathbf{a}^c, \Theta, x)$ models the way users answer questions. In the following sections, we describe each of these probability distributions as well as describe inference procedures for evaluating Eq 3.8 efficiently.

Computer Vision Model

As described in Eq 3.9, we require two basic types of computer vision algorithms: one that estimates attribute probabilities $p(\mathbf{a}^c|\Theta,x)$ on a particular set of predicted part locations Θ, and another that estimates part location probabilities $p(\Theta|x)$.

Attribute Detection

Using the independence assumptions depicted in Figure 3.7(b), we can write the probability

$$p(\mathbf{a}^c|\Theta,x) = \prod_{a_i^c \in \mathbf{a}^c} p(a_i^c|\theta_{\text{part}(a_i)},x). \tag{3.10}$$

Given a training set with labeled part locations $\theta_{\text{part}(a_i)}$, one can use standard computer vision techniques to learn an estimator for each $p(a_i|\theta_{\text{part}(a_i)},x)$. In practice, we train a separate binary classifier for each attribute, extracting localized features from the ground truth location $\theta_{\text{part}(a_i)}$. As in [54], we convert attribute classification scores $z_i = f_a(x;\text{part}(a_i))$ to probabilities by fitting a sigmoid function $\sigma(\gamma_a z_i)$ and learning the sigmoid parameter γ_a using cross-validation. When $v_{\text{part}(a_i)} = \textit{not visible}$, we assume the attribute detection score is zero.

Part Detection

We use a pictorial structure to model part relationships (see Figure 3.7(a)), where parts are arranged in a tree-structured graph $T = (V,E)$. Our part model is a variant of the model used by Felzenszwalb et al. [36], which models the detection score $g(x;\Theta)$ as a sum over unary and pairwise potentials $\log(p(\Theta|x)) \propto g(x;\Theta)$ with

$$g(x;\Theta) = \sum_{p=1}^{P} \psi(x;\theta_p) + \sum_{(p,q)\in E} \lambda(\theta_p,\theta_q) \tag{3.11}$$

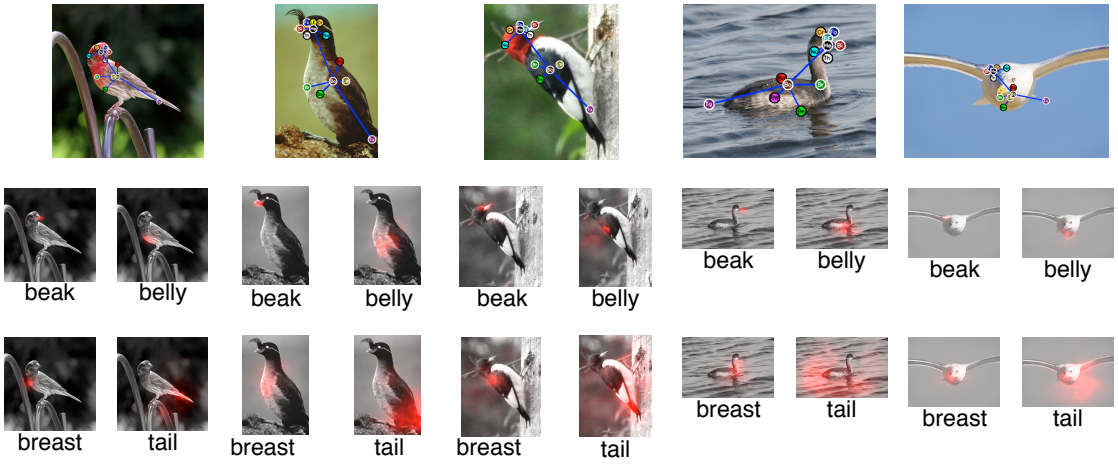


Figure 3.8. Fully Automated Part Detection Results: 5 test images with maximum likelihood estimates of 15 semantic parts superimposed on the image. Our system does a good job localizing all parts for the first two images, as is typical with side and frontal views of birds. The 3rd image is in an unusual horizontal pose; our system detects the parts of the head correctly but flips the orientation of the body upside down. The 4th image is an unusual bird shape; our system detects all parts more or less correctly but with some degree of noise. The last image is an uncommon pose for which detection fails entirely.

where each unary potential $\psi(x;\theta_p)$ is the response of a sliding window detector, and each pairwise score $\lambda(\theta_p,\theta_q)$ encodes a likelihood over the relative displacement between adjacent parts. We use the same learning algorithms and parametrization of each term in Eq 3.11 as in [122]. Here, parts and aspects are semantically defined, multiple aspects are handled using mixture models, and weight parameters for appearance and spatial terms are learned jointly using a structured SVM [100]. After training, we convert detection scores to probabilities $p(\Theta|x) \propto \exp(\gamma g(x;\Theta))$, where γ is a scaling parameter that is learned using cross-validation. Examples of fully automated part detection results are shown in Fig 3.8.

User Model

Readers interested in a computer-vision-only system with no human-in-the-loop can skip to Section 3.4. We assume that the probability of a set of user responses U^t can

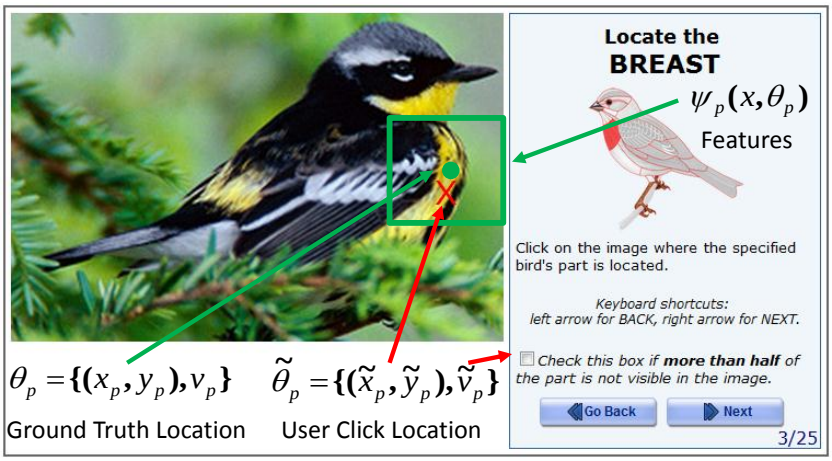


Figure 3.9. User interface for part locations input. The user clicks on his/her perceived location of the breast $(\tilde{x}_p, \tilde{y}_p)$, which is shown as a red X and is assumed to be near the ground truth location (x_p, y_p). The user also can click a checkbox indicating part visibility $\tilde{v}_p$. Features $\psi_p(x, \theta_p)$ can be extracted from a box around θ_p.

be expressed in terms of user responses that pertain to part click locations $U^t_\Theta \subseteq U^t$ and user responses that pertain to attribute questions $U^t_a \subseteq U^t$. We assume a user's perception of the location of a part $\tilde{\theta}_p$ depends only on the ground truth location of that part θ_p, and a user's perception of an attribute $\tilde{a}_i$ depends only on the ground truth attribute a^c_i:

$$p(U^t|\mathbf{a}^c, \Theta, x) = \left(\prod_{p \in U^t_\Theta} p(\tilde{\theta}_p|\theta_p) \right) \left(\prod_{\tilde{a}_i \in U^t_a} p(\tilde{a}_i|a^c_i) \right). \tag{3.12}$$

We describe our methods for estimating $p(\tilde{\theta}_p|\theta_p)$ and $p(\tilde{a}_i|a^c_i)$ as follows.

Modeling User Click Responses

Our interface for collecting part locations is shown in Figure 3.4. We represent a user click response as a triplet $\tilde{\theta}_p = \{\tilde{x}_p, \tilde{y}_p, \tilde{v}_p\}$, where $(\tilde{x}_p, \tilde{y}_p)$ is a point that the user clicks with the mouse and $\tilde{v}_p \in \{visible, \ not\ visible\}$ is a binary variable indicating presence/absence of the part.

Note that the user click response $\tilde{\theta}_p$ models only part location and visibility,

whereas the true part location θ_p also includes scale and aspect. This is done in order to keep the user interface as intuitive as possible. On the other hand, incorporating scale and aspect in the true model is extremely important – the relative offsets and visibility of parts in *left side view* and *right side view* will be dramatically different. We model a distribution over user click responses as

$$p(\tilde{\theta}_p|\theta_p) = p(\tilde{x}_p, \tilde{y}_p|x_p, y_p, s_p)p(\tilde{v}_p|v_p) \tag{3.13}$$

where the relative part click locations are Gaussian distributed

$$\left(\frac{\tilde{x}_p - x_p}{s_p}, \frac{\tilde{y}_p - y_p}{s_p}\right) \sim \mathcal{N}(\tilde{\mu}_p, \tilde{\sigma}_p^2), \tag{3.14}$$

and each $p(\tilde{v}_p|v_p)$ is a separate binomial distribution for each possible value of v_p. The parameters of these distributions are estimated using a training set of pairs $(\theta_p, \tilde{\theta}_p)$. This model of user click responses results in a simple, intuitive user interface and still allows for a sophisticated and computationally efficient model of part localization (Section 3.4).

Fig 3.10(b) visualizes 1 standard deviation when we learned our model (Eq 3.13) from over 26,000 clicks per part from Mechanical Turk workers. As a reference, we also include a comparison to computer vision part predictions (Section 3.4) in Fig 3.10(c).

Attribute Question Responses

We use a model of attribute user responses similar to [14]. We estimate each $p(\tilde{a}_i|a_i)$ as a binomial distribution, with parameters learned using a training set of user attribute responses collected from MTurk. As in [14], we allow users to qualify their responses with a certainty parameter *guessing*, *probably*, or *definitely*, and we incorporate a Beta prior to improve robustness when training data is sparse.

(a) Image-Level MTurk (b) Global MTurk (c) Computer Vision

Figure 3.10. Comparing part prediction accuracy for humans and computers: In each case, a Gaussian distribution over scale-normalized offsets between predictions and ground truth is estimated, and ellipses visualize 1 standard deviation from ground truth. 3.10(a): Image-level standard deviations over 5 MTurk users who labeled this particular Black-footed Albatross image. 3.10(b): Global standard deviations over 5,794 images and 5 users per image. Ellipses are superimposed onto an unrelated picture of a bird for visualization purposes. Global standard deviations appear larger than image-level ones because occasionally MTurkers click entirely on the wrong part. 3.10(c): Standard deviations over computer vision predictions (Section 3.4) for 5,794 test images. Standard deviations of computer vision predictions are much larger because occasionally computer vision detects the bird entirely in the wrong location.

Inference

We now describe the inference procedure for estimating the per-class probabilities $p(c|U^t,x)$ (Eq 3.7), which involves evaluating $\int_\Theta p(\mathbf{a}^c,U^t,\Theta|x)d\Theta$. While this initially seems very difficult, we note that all user responses $\tilde{a}_p^i$ and $\tilde{\theta}_p$ are observed values pertaining only to a single part, and attributes a^c are deterministic when conditioned on a particular choice of class c. If we run inference separately for each class c, all components of Eqs 3.10 and 3.12 can simply be mapped into the unary potential for a particular part. Evaluating Eq 3.7 exactly is computationally similar to evaluating a separate pictorial structure inference problem for each class.

On the other hand, when C is large, running C inference problems can be inefficient. In practice, we use a faster procedure which approximates the integral in Eq 3.8 as

a sum over K strategically chosen sample points:

$$\begin{aligned}
&\int_{\Theta} p(\mathbf{a}^c, U^t, \Theta|x) d\Theta \\
&\quad \approx \sum_{k=1}^{K} p(U^t|\mathbf{a}^c, \Theta_k^t, x) p(\mathbf{a}^c|\Theta_k^t, x) p(\Theta_k^t|x) \\
&\quad = p(U_a^t|\mathbf{a}^c) \sum_{k=1}^{K} p(\mathbf{a}^c|\Theta_k^t, x) p(U_\Theta^t|\Theta_k^t, x) p(\Theta_k^t|x).
\end{aligned} \tag{3.15}$$

We select the sample set $\Theta_1^t ... \Theta_K^t$ as the set of all local maxima in the probability distribution $p(U_\Theta^t|\Theta)p(\Theta|x)$. The set of local maxima can be found using standard methods for maximum likelihood inference on pictorial structures and then running non-maximal suppression, where probabilities for each user click response $p(\tilde{\theta}_p|\theta_p)$ are first mapped into a unary potential $\psi(x;\theta_p,\tilde{\theta}_p)$ (see Eq 3.11)

$$\psi(x;\theta_p,\tilde{\theta}_p) = \psi(x;\theta_p) + \log p(\tilde{\theta}_p|\theta_p). \tag{3.16}$$

The inference step takes time linear in the number of parts and pixel locations[3] and is efficient enough to run in a fraction of a second with 13 parts, 11 aspects, and 4 scales. Inference is re-run each time we obtain a new user click response $\tilde{\theta}_p$, resulting in a new set of samples. Sampling assignments to part locations ensures that attribute detectors only have to be evaluated on K candidate assignments to part locations; this opens the door for more expensive categorization algorithms (such as kernelized methods) that do not have to be run in a sliding window fashion.

Selecting the Next Question

In this section, we introduce a common framework for predicting the informativeness of different heterogeneous types of user input (including binary questions and

[3]Maximum likelihood inference involves a bottom-up traversal of T, doing a distance transform operation [36] for each part in the tree (takes time $O(n)$ time in the number of pixels).

mouse click responses) that takes into account the expected level of human error, informativeness in a multitask setting, expected annotation time, and spatial relationships between different parts. Our method extends the expected information gain criterion described in [14].

Let $\mathrm{IG}_t(q_j)$ be the expected information gain $\mathrm{IG}(c;u_j|x,U^t)$ from asking a new question q_j:

$$\mathrm{IG}_t(q_j) = \sum_{u_j \in \mathscr{A}_j} p(u_j|x,U^t)\big(\mathrm{H}(U^t,u_j) - \mathrm{H}(U^t)\big) \tag{3.17}$$

$$\mathrm{H}(U^t) = -\sum_c p(c|x,U^t)\log p(c|x,U^t) \tag{3.18}$$

where $\mathrm{H}(U^t)$ is shorthand for the conditional class entropy $\mathrm{H}(c|x,U^t)$. Evaluating Eq 3.17 involves considering every possible user-supplied answer $u_j \in \mathscr{A}_j$ to that question, and recomputing class probabilities $p(c|x,U^t,u_j)$. For yes/no attribute questions (querying a variable $\tilde{a}_i$), this is computationally efficient because the number of possible answers is only two, and attribute response probabilities $p(U_a^t|\mathbf{a}^\mathbf{c})$ are assumed to be independent from ground truth part locations (see Eq 3.15).

Predicting Informativeness of Mouse Clicks

In contrast, for part click questions the number of possible answers to each question is equal to the number of pixel locations, and computing class probabilities requires solving a new inference problem (Section 3.4) for each such location, which quickly becomes computationally intractable.

We use a similar approximation to the random sampling method used in the inference procedure. For a given part location question q_j, we wish to compute expected

entropy:

$$E_{\tilde{\theta}_p}[\mathrm{H}(U^t, \tilde{\theta}_p)] = \sum_{\tilde{\theta}_p} p(\tilde{\theta}_p | x, U^t)\, \mathrm{H}(U^t, \tilde{\theta}_p). \tag{3.19}$$

This can be done by drawing K samples $\tilde{\theta}^t_{p1} ... \tilde{\theta}^t_{pK}$ from the distribution $p(\tilde{\theta}_p | x, U^t)$, then computing expected entropy

$$\begin{aligned} &E_{\tilde{\theta}_p}[\mathrm{H}(U^t, \tilde{\theta}_p)] \approx \\ &\quad - \sum_{k=1}^{K} p(\tilde{\theta}_p | x, U^t) \sum_{c} p(c | x, U^t, \tilde{\theta}^t_{pk}) \log p(c | x, U^t, \tilde{\theta}^t_{pk}). \end{aligned} \tag{3.20}$$

In this case, each sample $\tilde{\theta}^t_{pk}$ is extracted from a sample Θ^t_k (Section 3.4) and each $p(c | x, U^t, \tilde{\theta}^t_{pk})$ is approximated as a weighted average over samples $\Theta^t_1 ... \Theta^t_K$. The full question selection procedure is fast enough to run in a fraction of a second on a single CPU core when using 13 click questions and 312 binary questions.

Selecting Questions By Time

The expected information gain criterion (Eq 3.17) attempts to minimize the total number of questions asked. This is suboptimal as different types of questions tend to take more time to answer than others (*e.g.*, part click questions are usually faster than attribute questions). We include a simple adaptation that attempts to minimize the expected amount of human time spent. The information gain criterion $\mathrm{IG}_t(q_j)$ encodes the expected number of bits of information gained by observing the random variable u_j. We assume that there is some unknown linear relationship between bits of information and reduction in human time. The best question to ask is then the one with the largest ratio of information gain relative to the expected time to answer it:

$$q^*_{j(t+1)} = \arg\max_{q_j} \frac{\mathrm{IG}_t(q_j)}{\mathbb{E}[\mathrm{time}(u_j)]} \tag{3.21}$$

where $\mathbb{E}[\text{time}(u_j)]$ is the expected amount of time required to answer a question q_j.

3.5 Datasets and Implementation Details

In this section we provide a brief overview of the datasets we used, methods used to construct visual questions, computer vision algorithms we tested, and parameter settings.

Birds-200 Dataset

Birds-200 [120] is a dataset of 6033 images over 200 bird species, such as Myrtle Warblers, Pomarine Jaegars, and Black-footed Albatrosses – classes which cannot usually be identified by non-experts. In many cases, different bird species are nearly visually identical (see Fig. 3.15).

We assembled a set of 25 visual questions (list shown in Fig. 3.5), which encompass 288 binary attributes (*e.g.*, the question HasBellyColor can take on 15 different possible colors). The list of attributes was extracted from whatbird.com [115], a bird field guide website.

We collected "deterministic" class-attributes by parsing attributes from whatbird.com. Additionally, we collected data of how non-expert users respond to attribute questions via a Mechanical Turk interface. To minimize the effects of user subjectivity and error, our interface provides prototypical images of each possible attribute response. Screenshots of the question answering user-interface are shown in Figure 3.17.

Fig. 3.5 shows a visualization of the types of user response results we get on the Birds-200 dataset. It should be noted that the uncertainty of the user responses strongly correlates with the parts that are visible in an image as well as overall difficulty of the corresponding bird species.

When evaluating performance, test results are generated by randomly selecting a

response returned by an MTurk user for the appropriate test image.

CUB-200-2011 Dataset

In order to perform experiments on part-based models, we extended the existing CUB-200 dataset [120] to form CUB-200-2011 [113], which includes roughly $11,800$ images, nearly double the previous total. Each image is annotated with 312 binary attribute labels and 15 part labels. We obtained a list of attributes from a bird field guide website [115] and selected the parts associated with those attributes for labeling. Five different MTurk workers provided part labels for each image by clicking on the image to designate the location or denoting part absence (Figure 3.4). One MTurk worker answered attribute questions for each image, specifying response certainty with options *guessing*, *probably*, and *definitely*. They were also given the option *not visible* if the associated part with the attribute was not present. At test time, we simulated user responses in a similar manner to [14], randomly selecting a stored response for each posed question. Instead of using bounding box annotations to crop objects, we used full uncropped images, resulting in a significantly more challenging dataset than CUB-200 [120].

Animals With Attributes

We also tested performance on the Animals With Attributes (AwA) [54], a dataset of 50 animal classes and 85 binary attributes. We consider this dataset less relevant than birds (because classes are not tightly related), and therefore do not focus as much on this dataset.

Implementation Details and Parameter Settings

Attributes-Based Only Model: For Birds-200 and AwA, our computer vision algorithms are based on Andrea Vedaldi's publicly available source code [106], which combines vector-quantized geometric blur and color/gray SIFT features using spatial pyramids,

multiple kernel learning, and per-class 1-vs-all SVMs. We added additional features based on full image color histograms and vector-quantized color histograms. For each classifier we used Platt scaling [82] to learn parameters for $p(c|x)$ on a validation set. We used 15 training examples for each Birds-200 class and 30 training examples for each AwA class. Bird training and testing images are roughly cropped.

Additionally, we compare performance to a second computer vision algorithm based on attribute classifiers, which we train using the same features/training code, with positive and negative examples set using whatbird.com attribute labels. We combined attribute classifiers into per-class probabilities $p(c|x)$ using the method described in [54].

For estimating user response statistics on the Birds-200 dataset, we used $\alpha_{guess} = 64$, $\alpha_{prob} = 16$, $\alpha_{def} = 8$, and $\alpha_c = 8$ (see Section 3.3.3).

Parts and Attributes-Based Model: For attribute detectors, we used simple linear classifiers based on histograms of vector-quantized SIFT and vector-quantized RGB features (each with 128 codewords) which were extracted from windows around the location of an associated part. We believe that significant improvements in classification performance could be gained by exploring more sophisticated features or learning algorithms.

As in [36], the unary scores of our part detector are implemented using HOG templates parametrized by a vector of linear appearance weights w_{v_p} for each part and aspect. The pairwise scores are quadratic functions over the displacement between (x_p, y_p) and (x_q, y_q), parametrized by a vector of spatial weights w_{v_p,v_q} for each pose and pair of adjacent parts. For computational efficiency, we assume that the pose and scale parameters are defined on an object level, and thus inference simply involves running a separate sliding window detector for each scale and pose. The ground truth scale of each object is computed based on the size of the object's bounding box.

Because our object parts are labeled only with visibility, we clustered images using k-means on the spatial x- and y- offsets of the part locations from their parent part

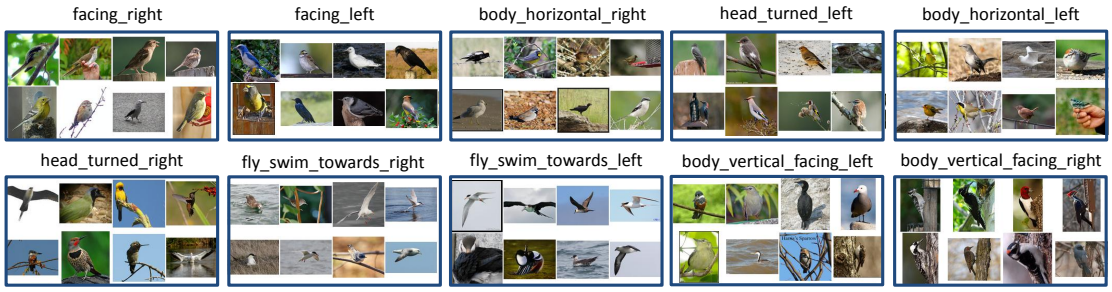


Figure 3.11. Pose clusters. Images in our dataset are clustered by k-means on the spatial offsets of part locations from parent part locations. Semantic labels of clusters were manually assigned by visual inspection. Left/right orientation is in reference to the image.

locations, normalized with respect to image dimensions; this approach handles relative part locations in a manner most similar to how we model part relationships (Section 3.4). Examples of images grouped by their pose cluster are shown in Figure 3.11. Semantic labels were assigned post hoc by visual inspection. The clustering, while noisy, reveals some underlying pose information that can be discovered by part presence and locations.

3.6 Experiments

In this section, we provide experimental results and analysis of the hybrid-human computer classification paradigm. Due to space limitations, our discussion focuses on the Birds dataset. We include results (see Fig. 3.16) from which the user can verify that trends are similar on Birds-200 and AwA.

3.6.1 Measuring Performance

We use two main methodologies for measuring performance, which correspond to two different possible user-interfaces:

- **Method 1:** We ask the user exactly T questions, predict the class with highest probability, and measure the percent of the time that we are correct.

- **Method 2:** After asking each question, we present to the user a small gallery of images of the class with highest probability and assume that the user will stop the system when presented with the correct class. In this case, we measure the average number of questions asked per test image.

For the second method, we assume that people are perfect verifiers, *e.g.*, they will stop the system if and only if they have been presented with the correct class. While this is not always possible in reality, there is some trade-off between classification accuracy and amount of human labor, and we believe that these two metrics collectively capture the most important considerations.

To evaluate performance for the localized, parts-based model, we introduce a third methodology that uses time as a measure of human effort needed to classify an object. This metric can be considered as a common quantifier for different forms of user input. Performance is determined by computing the average amount of time taken to correctly classify a test image. The computer presents images of the most likely class to the user, who will stop the system when the correct class is shown (similar to Method 2).

3.6.2 Using Binary Attribute Questions

In this section, we present our results and discuss some interesting trends toward understanding the visual 20 questions classification paradigm.

User responses are stochastic. In Fig. 3.12, we show the effects of different models of user responses without using any computer vision. When users are assumed to respond deterministically in accordance with the attributes from whatbird.com, performance rises quickly to 100% within 8 questions (roughly $\log_2(200)$). However, this assumption is not realistic; when testing with responses from Mechanical Turk, performance saturates at around 5%. Low performance caused by subjective answers are unavoidable (*e.g.*, perception of the color brown vs. the color buff), and the probability of the correct

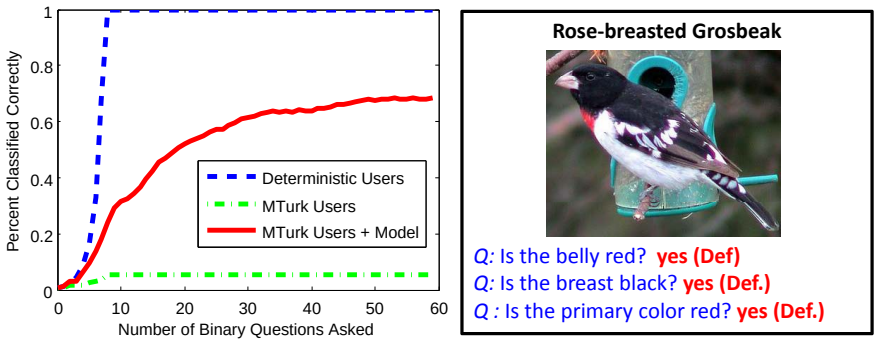


Figure 3.12. Different Models of User Responses: *Left:* Classification performance on Birds-200 (Method 1) without computer vision. Performance rises quickly (blue curve) if users respond deterministically according to whatbird.com attributes. MTurk users respond much differently, resulting in low performance (green curve). A learned model of MTurk responses is much more robust (red curve). *Right:* A test image where users answer several questions incorrectly and our learned model still classifies the image correctly.

class drops to zero after any inconsistent response. Although performance is 10 times better than random chance, it renders the system useless. This demonstrates a challenge for existing field guide websites in helping lay-people identify bird species. When our learned model of user responses (see Section 3.3.3) is incorporated, performance jumps to 70% due to the ability to tolerate a reasonable degree of error in user responses (see Fig. 3.12 for an example). Nevertheless, stochastic user responses increase the number of questions required to achieve a given accuracy level, and some images can never be classified correctly, even when asking all possible questions. In Section 3.6.2, we discuss the reasons why performance saturates at lower than 100% performance.

Computer vision reduces manual labor. The main benefit of computer vision occurs due to reduction in human labor (in terms of the number of questions a user has to answer). In Fig. 3.13, we see that computer vision reduces the average number of yes/no questions needed to identify the true bird species from 10.64 to 5.84 using responses from MTurk users. Without computer vision, the distribution of question counts is bell-shaped and centered around 7 questions. When computer vision is incorporated, the

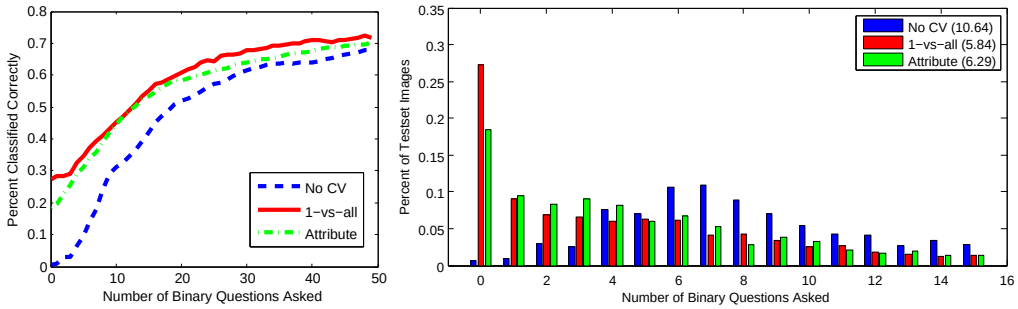


Figure 3.13. Performance on Birds-200 when using computer vision: Left Plot: comparison of classification accuracy (Method 1) with and without computer vision when using MTurk user responses. Two different computer vision algorithms are shown, one based on per-class 1-vs-all classifiers and another based on attribute classifiers. Right plot: the number of questions needed to identify the true class (Method 2) drops from 10.64 to 5.84 on average when incorporating computer vision.

distribution peaks at 0 questions but is more heavy-tailed, which suggests that computer vision algorithms are often good at recognizing the "easy" test examples (examples that are sufficiently similar to the training data), but provide diminishing returns toward classifying the harder examples that are not sufficiently similar to training data. As a result, computer vision is more effective at reducing the average amount of time necessary to classify an image than reducing the time spent on the most difficult images.

User responses drive up performance. An alternative way of interpreting the results is that user responses drive up the accuracy of computer vision algorithms. In Fig. 3.13, we see that user responses improve overall performance from $\approx 27\%$ (using 0 questions) to $\approx 72\%$.

Computer vision improves overall performance. Even when users answer all questions, performance saturates at a higher level when using computer vision ($\approx 72\%$ vs. $\approx 67\%$, see Fig. 3.13). The left image in Fig. 3.14 shows an example of an image classified correctly using computer vision, which is not classified correctly without computer vision, even after asking 60 questions. In this example, some visually salient features like the long neck are not captured in our list of visual attribute questions. The features

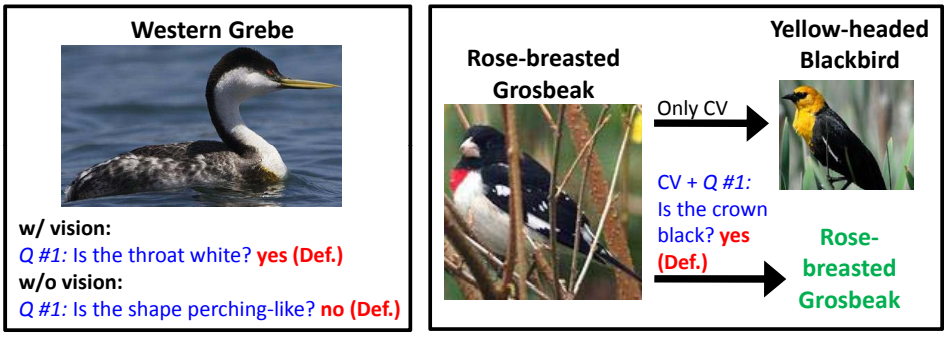


Figure 3.14. Examples where computer vision and user responses work together: *Left:* An image that is only classified correctly when computer vision is incorporated. Additionally, the computer vision based method selects the question HasThroatColor-White, a different and more relevant question than when vision is not used. In the right image, the user response to HasCrownColorBlack helps correct computer vision when its initial prediction is wrong.

used by our vision algorithms also capture other cues (such as global texture statistics) that are not well-represented in our list of attributes (which capture mostly color and part-localized patterns).

Different questions are asked with and without computer vision. In general, the information gain criterion favors questions that 1) can be answered reliably, and 2) split the set of possible classes roughly in half. Questions like HasShapePerchingLike, which divide the classes fairly evenly, and HasUnderpartsColorYellow, which tends to be answered reliably, are commonly chosen.

When computer vision is incorporated, the likelihood of classes change and different questions are selected. In the left image of Fig. 3.14, we see an example where a different question is asked with and without computer vision, which allows the computer vision based method to hone in on the correct class using one question.

Recognition is not always successful. According the the Cornell Ornithology Website [102], the four keys to bird species recognition are 1) size and shape, 2) color and pattern, 3) behavior, and 4) habitat. Bird species classification is a difficult problem and is not

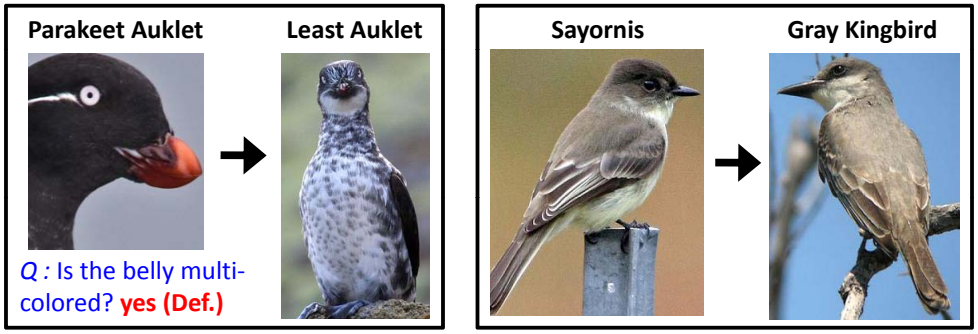


Figure 3.15. Images that are misclassified by our system: *Left:* The Parakeet Auklet image is misclassified due to a cropped image, which causes an incorrect answer to the belly pattern question (the Parakeet Auklet has a plain, white belly, see Fig. 3.3). *Right:* The Sayornis and Gray Kingbird are commonly confused due to visual similarity.

always possible using a single image. One potential advantage of the visual 20 questions paradigm is that other contextual sources of information such as behavior and habitat can easily be incorporated as additional questions.

Fig. 3.15 illustrates some example failures. The most common failure conditions occur due to 1) classes that are nearly visually identical, 2) images of a poor viewpoint or low resolution where some parts are not visible, 3) significant mistakes made by MTurkers, or 4) limitations in the particular set of attributes we selected.

3.6.3 1-vs-all Vs. Attribute-Based Classification

In general, 1-vs-all classifiers slightly outperform attribute-based classifiers; however, they converge to similar performance as the number of question increases, as shown in Fig. 3.13 and 3.16. The features we use (kernelized and based on bag-of-words) may not be well suited to the types of attributes we are using, which tend to be localized and associated with a particular part. One potential advantage of attribute-based methods is computational scalability when the number of classes increases; whereas 1-vs-all methods always require C classifiers, the number of attribute classifiers can be varied in order to trade-off accuracy and computation time. The table below displays the average

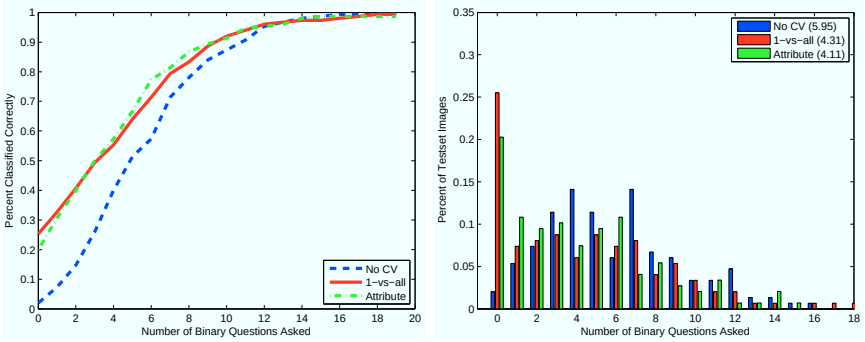


Figure 3.16. Performance on Animals With Attributes: Left Plot: Classification performance (Method 1), simulating user responses using soft class-attributes (see [54]). Right Plot: The required number of questions needed to identify the true class (Method 2) drops from 5.94 to 4.11 on average when incorporating computer vision.

number of questions needed (Method 1) on the Birds dataset using different number of attribute classifiers (which were selected randomly):

200 (1-vs-all)	288 attr.	100 attr.	50 attr.	20 attr.	10 attr.
5.84	6.29	6.49	7.37	8.76	9.40

3.6.4 Using Part and Attribute Questions

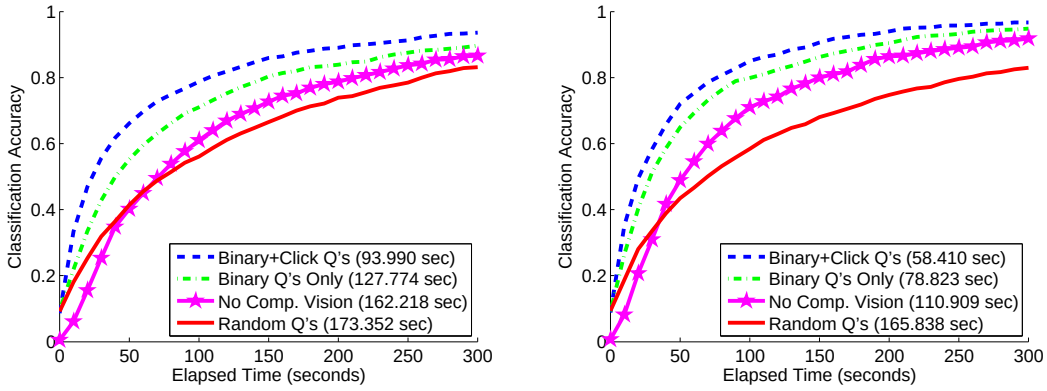
Using our criteria for question selection (Section 3.4) and our performance metric based on time-to-classification, we examine the average classification accuracy for: (1) our integrated approach combining localization/classification algorithms and part click and binary attribute questions; (2) using binary questions only with non-localized computer vision algorithms and expected information gain to select questions (representative of [14]); (3) using no computer vision; and (4) selecting questions at random. We follow with observations on how the addition of click questions affects performance and human effort required. Examples of attribute and part questions that are posed to the user at test-time are shown in Figure 3.17.

Question selection by time reduces human effort. By minimizing human effort with the time criterion, we are trading off between the expected information gain from a

(a) Attribute question

(b) Part question

Figure 3.17. Attribute and Part Questions. 3.17(a): for the attribute question *what is the wing color* the user selects both *black* and *white* and qualifies her answer with a certainty *definitely*. 3.17(b): for the part click question *click on the tail*, the user provides an (x, y) mouse location.



(a) Information gain criterion (b) Time criterion

Figure 3.18. Interactive Classification Using Part and Attribute Questions. Classification accuracy as a function of time when 3.18(a) maximizing expected information gain; and 3.18(b) minimizing amount of human labor, measured in time. Performance is measured as the average number of seconds to correctly classify an image (described in Section 3.6.1).

question response and the expected time to answer that question. Subsequently, we are able to classify images in 36.6 seconds less on average using both binary and click questions than if we only take into account expected information gain; however, the margin in performance gain between using and not using click questions is reduced.

We note that the average time to answer a part click question is 3.01 ± 0.26 seconds, compared to 7.64 ± 5.38 seconds for an attribute question; in this respect, part questions are more likely to be asked first.

Part localization improves performance. In Figure 3.18(a), we observe that by selecting the next question using our expected information gain criterion, average classification time using both types of user input versus only binary questions is reduced by 33.8 seconds on average. Compared to using no computer vision, we note an average reduction in human effort of over 40% (68.2 seconds).

Using the time criterion for selecting questions, the average classification time for a single image using both binary and click questions is 58.4 seconds. Asking binary

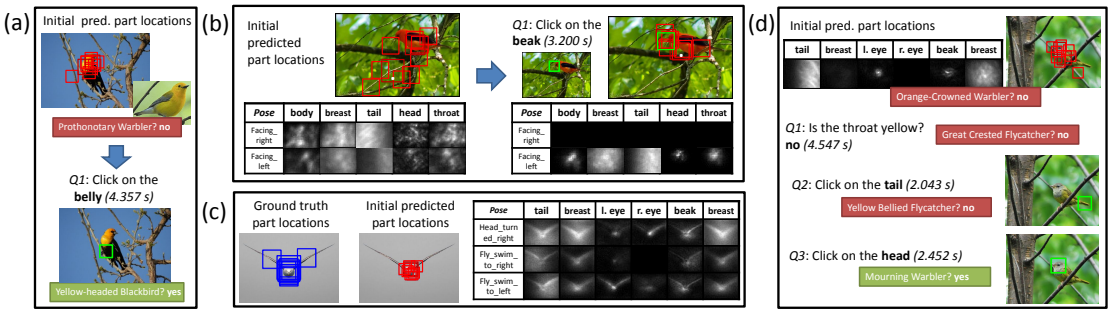


Figure 3.19. Examples of the behavior of our system. 3.19(a): The system estimates the bird pose incorrectly but is able to localize the head and upper body region well, and the initial class prediction captures the color of the localized parts. The user's response to the first system-selected part click question helps correct computer vision. 3.19(b): The bird is incorrectly detected, as shown in the probability maps displaying the likelihood of individual part locations for a subset of the possible poses (not visible to the user). The system selects "*Click on the beak*" as the first question to the user. After the user's click, the other part location probabilities are updated and exhibit a shift towards improved localization and pose estimation. 3.19(c): Certain infrequent poses (*e.g.* frontal views) were not discovered by the initial off-line clustering (see Figure 3.11). The initial probability distributions of part locations over the image demonstrate the uncertainty in fitting the pose models. The system tends to fail on these unfamiliar poses. 3.19(d): The system will at times select both part click and binary questions to correctly classify images.

questions only, the system takes an additional 20.4 seconds on average to correctly classify an image (Figure 3.18(b)). Using computer vision algorithms, we are able to consistently achieve higher average classification accuracy than using no computer vision at all, in the same period of time.

User responses drive up performance. There is a disparity in classification accuracy between evaluating attribute classifiers on ground truth locations (17.3%) versus predicted locations (10.3%); by using user responses to part click questions, we are able to overcome initial erroneous part detections and guide the system to the correct class. Figure 3.19(a) presents an example in which the bird's pose is estimated incorrectly. After posing one question and re-evaluating attribute detectors for updated part probability

distributions, our model is able to correctly predict the class.

In Figure 3.19(b), we visualize the question-asking sequence and how the probability distribution of part locations over the image changes with user clicks. We note in Figure 3.19(c) that our pose clusters did not discover certain poses, especially frontal views, and the system is unable to estimate the pose with high certainty.

As previously discussed, part click questions take on average less time to answer. We observe that the system will tend to ask 2 or 3 part click questions near the beginning and then continue with primarily binary questions (*e.g.* Figure 3.19(d)). At this point, the remaining parts can often be inferred reliably through reasoning over the spatial model, and thus binary questions become more advantageous.

3.7 Conclusion

We have proposed a novel approach to object recognition of fine-grained categories that efficiently combines class attribute and part models and selects questions to pose to the user in an intelligent manner. Our experiments, carried out on a challenging dataset including 200 bird species, show that our system is accurate and quick. In addition to demonstrating our approach on a diverse set of basic-level categories, future work can include introducing more advanced image features in order to improve attribute classification performance. Furthermore, we used simple mouse clicks to designate part locations, and it would be of interest to investigate whether asking the user to provide more detailed part and pose annotations would further speed up recognition.

Acknowledgements

Chapter 3 is primarily based on material from “Visual Recognition with Humans in the Loop” by S. Branson, C. Wah, B. Babenko, F. Schroff, P. Welinder, P. Perona, and

S. Belongie [14]. The dissertation author implemented parts of the code and experiments, and contributed to the writing of the paper.

Parts of Section 3.1 and 3.4 are based on material from "The Ignorant Led by the Blind: A Hybrid Human-Machine Vision System for Fine-Grained Categorization" by S. Branson, G. Van Horn, C. Wah, P. Perona, and S. Belongie [12]. The dissertation author contributed to algorithm development, implemented code and experiments, and contributed to the writing of the paper.

Section 3.4 is primarily based on material from "Multiclass Recognition and Localization with Humans in the Loop" by C. Wah, S. Branson, P. Perona, and S. Belongie [112]. The dissertation author contributed to algorithm development, implemented code and experiments, and contributed to the writing of the paper.

Chapter 4

Interactive Categorization with Similarity Learning

4.1 Introduction

Within the realm of visual categorization in computer vision, humans can play multiple roles, as we have observed. As experts, they can define a comprehensive set of semantic parts and attributes to describe and differentiate categories, as well as provide ground truth attribute values, such as for a field guide. As non-expert users of interactive classification systems [14, 112], they can also supply these attribute and part annotations.

These attribute-based methods have several weaknesses, especially within fine-grained visual categorization. Fine-grained categories comprise the set of classes (*e.g.* Pembroke Welsh Corgi, Shiba Inu) within a basic-level category (*e.g.* dogs); each basic-level category requires its own unique, discriminative part and attribute vocabulary. Acquiring this vocabulary involves identifying an expert resource (*e.g.* a field guide) for that basic-level category. For certain categories, such as chairs or paintings, it may be difficult to produce an adequate vocabulary. Furthermore, one must obtain image- or class-level annotations for these attributes. Even if the labels were crowdsourced, each basic-level category would require a custom set of annotation tools, and building these tools is a nontrivial task.

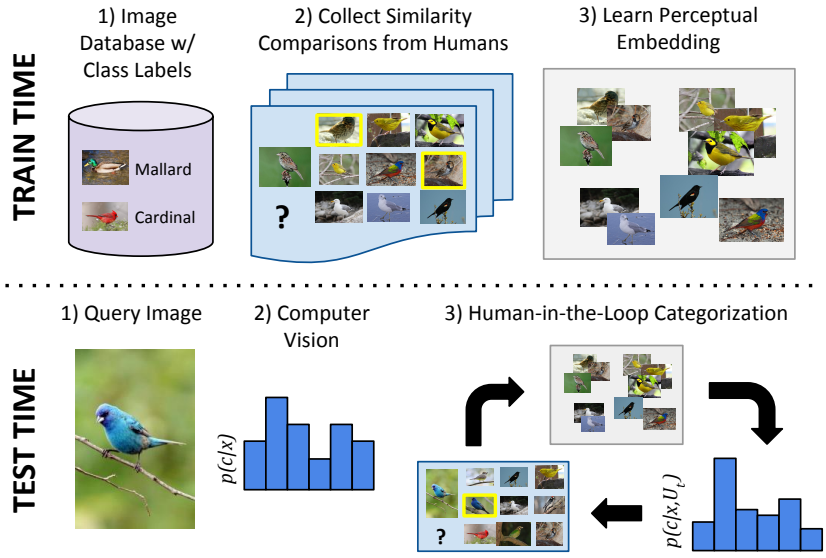


Figure 4.1. Similarity metrics for interactive categorization. Our interactive categorization system learns a perceptual similarity metric from human similarity comparisons on a fixed training set of images and class labels. At test time, our system leverages this learned metric, along with similarity comparisons provided by the user, to classify out-of-sample query images.

In addition, users may have difficulty understanding the domain-specific jargon used to articulate the semantic attribute vocabulary. The fixed-size vocabulary may also lack sufficient discriminative attributes for recognition. Thus, the cost in obtaining attribute vocabularies is high, making it expensive to extend an existing system to new categories.

In this chapter, we present an approach to visual categorization (Fig. 4.1) that is based on perceptual similarity rather than an attribute vocabulary. We assume that we are provided with a fine-grained dataset of images that are annotated with only class labels. In an offline stage, we collect relative similarity comparisons between images in the dataset, and then leverage these human-provided comparisons to perform visual categorization.

This similarity-based approach to interactive classification has several compelling

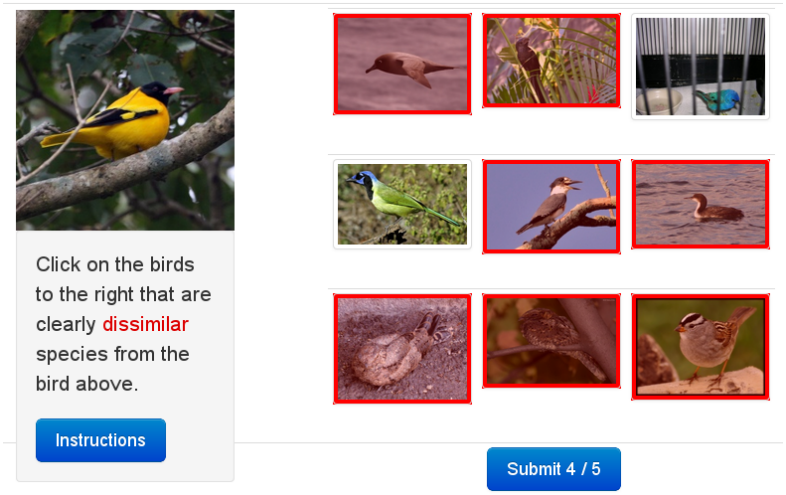


Figure 4.2. Interface for Collecting Similarity Comparisons. An example of the interface used for offline collection of similarity comparisons, from which we learn a similarity metric.

advantages. First, we no longer require part and attribute vocabularies, which can be expensive to obtain. By eliminating the need for experts to predefine these vocabularies, we no longer constrain users by expert-defined terminology. Moreover, the continuous embedded similarity space is a richer and vastly more powerful representation than these typically fixed-size vocabularies. These factors facilitate the adaptation of an existing similarity-based system to other basic-level categories.

This similarity-based paradigm enables us to incrementally improve our computer vision models and algorithms while providing a useful service to users. Each user response collected at test time can further refine the learned similarity metrics and consequently improve performance. In addition, our flexible framework supports a variety of off-the-shelf computer vision algorithms, such as SVMs, logistic regression, and distance learning algorithms, all of which can be easily mapped into the system.

The psychology literature [101] informs us that humans judge similarity subjectively based on various universal factors that may differ from person to person; in

evaluating similarity between objects in images, these factors could be based on category, pose, background, illumination, *etc*. Because of this, we also study how multiple general-purpose similarity metrics, with respect to universal factors such as color and shape, can be used to perform categorization.

4.2 Related Work

Recently, the computer vision community has seen a burst of interest in interactive classification systems [14, 112, 52], several of which build on attribute-based classification methods. Some works harvest attributes through various means [54, 33, 53, 32, 73], while others discover attributes in an automatic or interactive manner [84, 6, 25], relying on users to identify and name attributes [74, 50, 62, 63] or to provide feedback in order to improve classification [22, 78].

In contrast to these attribute-centric methods, we focus on similarity. Some recent works use similarity in feature space [53, 4]; others rely on human judgment to quantify similarity for classifying attributes or clustering categories [75, 22, 43, 56]. We instead learn a metric of perceptual similarity for categorization from relative comparisons [90, 1, 65, 97], specifically employing stochastic triplet embedding [104] in this work. Some works capture multiple modalities of similarity rather than use a single metric [103, 66, 18], while we focus on learning independent metrics of perceptual similarity.

Another related area is relevance feedback-based image retrieval [89, 2, 19, 46, 127]. Some works, *e.g.*, [88], have focused on identifying nonlinear manifolds that better align with human perception; however, they do not adequately bridge the semantic gap or capture perceptual measures of similarity. In particular, our work bears similarities to the relevance feedback system presented in [39] but differs in several important ways. First, the motivating assumption in [39] is that the user possesses only a mental image or concept of a semantic category. We instead assume existence of the query image, such

that we are able to incorporate computer vision at test time. Second, [39] uses a single similarity metric derived from visual features (*i.e.* GIST) rather than human perception; we conduct human experiments to generate a perceptual embedding of the data. We combine this perceptual similarity metric along with computer vision as part of a unified framework for recognition. Our system supports multiple similarity metrics and is able to trade off between these metrics at test time. To our knowledge, no other existing system combines perceptual and visual information for categorization in this integrated manner.

4.3 Perceptual Similarity Metrics for Interactive Categorization

We present in this section an efficient, flexible, and scalable system for fine-grained visual categorization that is based on perceptual similarity and combines different types of similarity metrics and computer vision methods in a unified framework. Additionally, we demonstrate the value in using a perceptual similarity metric over relevance feedback-based image retrieval methods and vocabulary-dependent attribute-based approaches.

Our visual categorization system is similar to the system in [39], with several important distinctions. While our system shares aspects of [39]'s user and display models, it uses similarity metrics that are derived from human perception of similarity rather than computer vision features, which allow us to bridge the "semantic gap" of many content-based image retrieval systems [19], including [39]. This semantic gap references the disparity between information extracted from visual data and how the user perceives and interprets that data [19]. Second, we assume that a query image is available at test time, enabling us to incorporate computer vision algorithms that are evaluated on the test image in order to initialize per-class probabilities [14]. Our system reduces human effort (as measured by the average number of questions posed to the user) by 43%, compared

to an implementation of [39] that has been initialized using computer vision.

In Section 4.2, we discuss relevant work. In Section 4.3.1, we introduce our method for learning similarity metrics and describe how we integrate those metrics in our framework. In Section 4.4, we present an extension to multiple localized region-based similarity metrics. We discuss implementation details in Section 4.5 and present our experimental results in Section 4.6.

4.3.1 Methods and Framework

We formulate the problem as follows. Given an image x, we wish to predict the object class from C possible classes that fall within a common basic-level category, where $\mathscr{C}$ is the set of images belonging in the true object class. We do so using a combination of computer vision and a series of questions that are interactively posed to a user. Each question contains a display D of images, and the user is asked to make a subjective judgment regarding the similarity of images in D to the target image x, providing a response u.

An image x in pixel space can also be represented as a vector $\mathbf{z}$ in human-perceptual space. At train time, we are given a set of N images and their class labels $\{(x_i, c_i)\}_{i=1}^{N}$. We ask similarity questions to human users to learn a perceptual embedding $\{(x_i, \mathbf{z}_i, c_i)\}_{i=1}^{N}$ of the training data. At test time, we observe an image x and pose questions to a human user, and we obtain probabilistic estimates of $\mathbf{z}$ and c that are incrementally refined as the user answers more questions.

Learning Similarity Metrics from Triplet Constraints

In this section, we describe how we use similarity comparisons collected from humans to learn a perceptual embedding of similarity (Sec. 4.3.1). We begin by obtaining a set of K user similarity comparisons in an offline data collection stage; more details

regarding this step are discussed in Section 4.5. Each collected user response is interpreted as follows.

A user is asked to judge the similarity between a target image x and a display D that comprises a set $\mathscr{I}$ of G images. From each user response u_k, $k = 1 \ldots K$, we obtain two disjoint sets: one set $\{x_{S_1}, x_{S_2}, \ldots, x_{S_n}\} \in \mathscr{I}_S$ represents the images judged as similar to the query image; and $\{x_{D_1}, x_{D_2}, \ldots, x_{D_m}\} \in \mathscr{I}_D$ includes all other images, such that $\mathscr{I}_D \cup \mathscr{I}_S = \mathscr{I}$. Recall that a user response for a given query image x yields two sets $\mathscr{I}_D$ and $\mathscr{I}_S$. We broadcast this to an equivalent set of (noisy) triplet constraints $\mathscr{T}^k$:

$$\mathscr{T}^k = \{(i, j, l) | x_i \text{ is more similar to } x_j \text{ than } x_l\}, \tag{4.1}$$

where i is the target image, represented as x_i; j is from set $\mathscr{I}_S$; and l is drawn from set $\mathscr{I}_D$. Therefore, for each user response, we obtain nm triplet constraints in $\mathscr{T}^k$. For a display size $G = 9$, this value can range from 8 to 20 triplet constraints per user response. Constraints from each user response are then added to a comprehensive set $\mathscr{T}$ of similarity triplets.

Generating a Perceptual Embedding

Let $s(i, j)$ denote the perceptual similarity between two images x_i and x_j. Using $\mathscr{T}$, we wish to find an embedding $\mathbf{Z}$ of N training images $\{\mathbf{z}_1, \ldots, \mathbf{z}_N\} \in \mathbb{R}^r$ for some $r \leq N$, in which triplet comparisons based on Euclidean distances are consistent with $s(\cdot, \cdot)$. In other words, we want the following to occur with high probability:

$$\|\mathbf{z}_i - \mathbf{z}_j\|_2 < \|\mathbf{z}_i - \mathbf{z}_l\|_2 \Longleftrightarrow s(i, j) > s(i, l). \tag{4.2}$$

The dimensionality r is empirically chosen based on minimizing generalization error (see Sec. 4.6.1). We use the metric learning approach described in [104] and optimize for the

embedding $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_N]$, such that for each triplet (i, j, l) the similarity of $\mathbf{z}_i$ and $\mathbf{z}_j$ is large in comparison to the similarity of $\mathbf{z}_i$ and $\mathbf{z}_l$ according to a Student-t kernel; we refer the reader to [104] for additional details. From the learned embedding $\mathbf{Z}$, we generate a similarity matrix $S \in N \times N$ with entries:

$$S_{ij} = \exp\left(-\frac{\|\mathbf{z}_i - \mathbf{z}_j\|^2}{2\sigma^2}\right), \tag{4.3}$$

which can be directly used in our classification system. The scaling parameter σ is learned jointly with the user response model parameters (see Sec. 4.3.1). In practice, this matrix can be reduced to $S \in C \times C$, where C is the number of classes, by pooling over images in each class (see Sec. 4.6.1).

Human-in-the-Loop Classification

Given a test image x, the goal of our human-in-the-loop classification system is to identify the true class c as quickly as possible using a combination of computer vision and user responses to similarity questions. At each timestep t, the system intelligently chooses a display D_t of G images to show. The user provides a response u_t, selecting the image perceived to be most similar to the test image x. Let $U_t = u_1 \ldots u_t$ be the set of responses obtained within timestep t. Our goal is to predict class probabilities $p(c|x, U_t)$ while exploiting the visual content of the image x and user responses U_t. We compute class probabilities by marginalizing over all possible locations $\mathbf{z}$ of image x in perceptual space:

$$p(c, U_t|x) \quad = \quad \int_z p(c, \mathbf{z}, U_t|x) dz \tag{4.4}$$

where $p(c|x, U_t) \propto p(c, U_t|x)$. Our probabilistic prediction of the location $\mathbf{z}$ and the class c becomes increasingly refined as the user answers more questions. We can further

decompose $p(c, \mathbf{z}, U_t|x)$ into terms:

$$p(c, \mathbf{z}, U_t|x) \quad = \quad p(U_t|c, \mathbf{z}, x)p(c, \mathbf{z}|x) \tag{4.5}$$

where $p(U_t|c, \mathbf{z}, x)$ is a model of how users respond to similarity questions, and $p(c, \mathbf{z}|x)$ is a computer vision estimate.

User Response Model

We describe our probabilistic model of how users answer similarity questions as follows. We decompose user response probabilities $p(U_t|c, \mathbf{z}, x)$ as such:

$$p(U_t|c, \mathbf{z}, x) = p(U_t|\mathbf{z}) = \prod_{b=1}^{t} p(u_b|\mathbf{z}). \tag{4.6}$$

Here, we assume that a user's response to similarity questions depends only on the true location $\mathbf{z}$ in perceptual space and that answers to each question are independent. Recall that each similarity question comprises a display D of G images, and the user responds by selecting the index $i \in D$ of an image that is perceived to be most similar to the test image. A perfect user would deterministically choose the image x_i for which the perceptual similarity $s(\mathbf{z}, \mathbf{z}_i)$ is highest, such that:

$$p(u|\mathbf{z}) = 1[s(\mathbf{z}, \mathbf{z}_i) = \max_{j \in D} s(\mathbf{z}, \mathbf{z}_j)]. \tag{4.7}$$

However, real users may respond differently due to subjective differences and user error. We thus model noisy responses probabilistically, assuming that the probability that the user selects i is proportional to its similarity $s(\mathbf{z}, \mathbf{z}_i)$ to the test image x:

$$p(u|\mathbf{z}) = \frac{\phi(s(\mathbf{z}, \mathbf{z}_i))}{\sum_{j \in D} \phi(s(\mathbf{z}, \mathbf{z}_j))} \tag{4.8}$$

where $\phi(\cdot)$ is some customizable, monotonically increasing function. In practice, we use

$$\phi(s) = \max(\theta, (1-\theta)s) \tag{4.9}$$

where θ is a learnable parameter. This model of $p(u|\mathbf{z})$ can be understood as a mixture of two distributions: with probability θ a user selects an image at random (*e.g.*, due to user error); otherwise, a user selects an image with probability proportional to its perceptual similarity. Recall from Eq 4.3 that $s(\mathbf{z}, \mathbf{z}_j)$ contains an additional parameter σ. Similar to [38], the parameters σ and θ are learned by maximizing the log-likelihood of a validation set of 200 non-Turker human user responses.

Efficient Computation

Recall that the user sequentially answers a series of similarity questions $U_t = u_1 \ldots u_t$. In this section, we derive an efficient algorithm for updating class probability estimates $p(c|x, U_t)$ in each timestep t.

Let w_k^t be shorthand for the probability $p(c_k, \mathbf{z}_k, U_t|x)$:

$$w_k^t = \left(\prod_{r=1}^{t} p(u_r|\mathbf{z}_k) \right) p(c_k, \mathbf{z}_k|x) \tag{4.10}$$

where k enumerates images in the training set. Each weight w_k captures how likely location $\mathbf{z}_k$ is the true location $\mathbf{z}$. Note that w_k^{t+1} can be efficiently computed from w_k^t as:

$$w_k^{t+1} = p(u_{t+1}|\mathbf{z}_k) w_k^t = \frac{\phi(S_{ik})}{\sum_{j \in D} \phi(S_{jk})} w_k^t \tag{4.11}$$

where i is the selected image at $t+1$, S_{ij} is an entry of the similarity matrix (Sec. 4.3.1), and $w_k^0 = p(c_k, \mathbf{z}_k|x)$. To estimate class probabilities, we approximate the integral in

Eq 4.4 as the sum over training examples:

$$p(c, U_t|x) \approx \frac{1}{N} \sum_{\substack{k=1\ldots n, \\ c_k=c}} p(c_k, \mathbf{z}_k, U_t|x). \tag{4.12}$$

By the definition of w_k^t and normalizing probabilities, it follows that $p(c|x, U_t)$ is the sum of the weights of training examples of class c:

$$p(c|x, U_t) = \frac{\sum_{k, c_k=c} w_k^t}{\sum_k w_k^t}, \tag{4.13}$$

resulting in an efficient algorithm where we maintain weights w_k^t for each training example: (1) we initialize weights $w_k^0 = p(c_k, \mathbf{z}_k|x)$ (estimated using computer vision; see Sec. 4.3.2); (2) we update weights when the user answers a similarity question (Eq 4.11); and (3) we update per-class probabilities (Eq 4.13).

Choosing Which Images to Display

Recall that at each timestep, our system intelligently poses a similarity question by selecting a display D of G images. We wish to choose the set of images that maximizes expected information gain. We follow the procedure used by Ferecatu and Geman [39], which defines an efficient approximate solution for populating this display. We group the images into equal-weight clusters, where each image possesses mass w_k^t. This ensures that each image in the display is equally likely to be clicked, maximizing the information gain in terms of the entropy of $p(c, \mathbf{z}_k, U_t|x)$. Given the clustering of images, we pick the image within the cluster with the highest mass for the display using an approximate solution. We refer the reader to [31, 39] for additional details. A similar procedure can be used to instead pick a set of G classes to display, assigning each class a mass $\sum_{k, c_k=c} w_k^t$, maximizing the information gain in terms of the entropy of $p(c|x, U_t)$.

4.3.2 Incorporating Computer Vision

Recall from Eq 4.5 that we would like to train an estimator for $p(c, \mathbf{z}|x)$, the probability that an observed image x belongs to a particular class c and location $\mathbf{z}$ in perceptual space. In practice, our human-in-the-loop classification algorithm (as described in Sec. 4.3.1) only requires us to estimate $w_k^0 = p(c_k, \mathbf{z}_k|x)$ for training examples $k = 1...N$ rather than for all possible values of $\mathbf{z}$. In this section, we show how off-the-shelf computer vision algorithms such as SVMs, boosting, logistic regression, and distance learning algorithms can be mapped into this framework. We also discuss novel extensions for designing new algorithms that are more customized to the form of $p(c, \mathbf{z}|x)$. For each such method, we describe the resulting computation of w_k^0.

No Computer Vision

If no computer vision algorithm is available, then we have no information toward predicting c or $\mathbf{z}$ based on observed image pixels x. As such, we assume each location $\mathbf{z}_k$ is equally likely:

$$w_k^0 = p(c_k, \mathbf{z}_k|x) = \frac{1}{N}. \tag{4.14}$$

Classification Algorithms

Classification algorithms such as SVMs, boosting, and logistic regression produce a classification score that can be adapted to produce a probabilistic output $p(c|x)$. They are otherwise agnostic to the prediction of $\mathbf{z}$. We thus assume that $\mathbf{z}_i$ and $\mathbf{z}_j$ are equally likely for examples of the same class $c_i = c_j$:

$$w_k^0 = p(c_k, \mathbf{z}_k|x) = \frac{1}{N_{c_k}} p(c_k|x) \tag{4.15}$$

where N_c is the number of training images of class c. We learn parameters for $p(c|x)$ on a validation set [82].

Distance-Based Algorithms

Non-parametric methods (e.g., nearest neighbor and distance-learning methods) can be adapted to produce a similarity $s(x_k,x)$ between x and the k_{th} training example (computed using low-level image features) but are otherwise agnostic to class:

$$w_k^0 = p(c_k,\mathbf{z}_k|x) \propto s(x_k,x). \tag{4.16}$$

A Gaussian kernel $s(x_k,x) = \exp\{-d(x_k,x)/\sigma\}$ is commonly used, where $d(x_k,x)$ is a distance function and σ is estimated on a validation set. Note that due to normalization in Eq 4.13, using an unnormalized probability does not affect correctness.

Pose-Based Classification Algorithms

Note that the above classification and distance-based algorithms are sub-optimal due to not exploiting information in $\mathbf{z}_k$ and c, respectively. We consider a simple extension to help remedy this. We obtain a perceptual pose embedding $\mathbf{Z}^o$ of the training data using pose similarity questions (see Sec. 4.3.1), then cluster training examples $\mathbf{z}_1^o...\mathbf{z}_N^o$ using k-means into K discrete poses. Let o_i be the pose index of the i_{th} example. We train a separate multiclass classifier for each pose o, obtaining a pose-conditioned class estimator for $p(c|x,o)$. We similarly train a multiclass pose classifier that estimates pose probabilities $p(o|x)$. We assume our classifiers give us information about $\mathbf{z}$ through pose labels o but are otherwise agnostic to the prediction of $\mathbf{z}$:

$$w_k^0 = p(c_k,\mathbf{z}_k|x) = \frac{1}{N_{c_k,o_k}} p(c_k|x,o_k)p(o_k|x) \tag{4.17}$$

where N_{co} is the number of training examples of class c and pose o. At test time, we have the option of asking a mixture of class and pose similarity questions. In practice, we found that pose-conditioned classification accuracy did not match that of a multiclass classifier. This may be due to lack of (positive) training examples and lack of object localization. However, we include this section of an example of how one might formulate novel computer vision algorithms that are customized to a similarity-based human-in-the-loop interface.

4.4 Extension to Multiple Localized Perceptual Metrics

Current similarity-based approaches to interactive fine-grained categorization rely on learning metrics from holistic perceptual measurements of similarity between objects or images. However, making a single judgment of similarity at the object level can be a difficult or overwhelming task for the human user to perform. Secondly, a single general metric of similarity may not be able to adequately capture the minute differences that discriminate fine-grained categories. In this work, we propose a novel approach to interactive categorization that leverages multiple perceptual similarity metrics learned from localized and roughly aligned regions across images, reporting state-of-the-art results and outperforming methods that use a single nonlocalized similarity metric.

While similarity can be holistic in nature (*e.g.*, object utility or function, or overall shape), it can also be highly localized, for instance, when specific corresponding regions or parts of the object differ from one other. Especially at the fine-grained category level in which classes tend to be visually coherent, it is likely that the small yet important characteristics that distinguish subcategories are localizable. In these scenarios, a single metric of perceptual similarity that is observed at the object level can be overly general, and asking a user to make holistic nonlocalized similarity comparisons can be difficult.

By using localized similarity comparisons and constraining the user's view to a

portion of the image, we are able to highlight certain aspects of similarity; these localized judgments tend to be easier for humans to perform than holistic similarity judgments (see Figure 4.3). Moreover, we can potentially reduce the effect of nuisance factors such as background noise and differing object poses. For each common region or part, we learn a separate perceptual space that captures local visual information.

In order to compare common local regions between images, we must first identify the set of relevant regions to consider, and second, we must determine spatial correspondences between regions across images and objects. For many basic-level categories, there exist field guides that specify part vocabularies for describing or discriminating categories, but these share the same weaknesses as semantic attribute vocabularies. The regions that are most useful for discrimination may not align with part semantics, and moreover, additional annotation is required to localize all the regions in the images.

We propose using an unsupervised approach to discovering discriminative, visually coherent and roughly aligned regions [92, 21] in the dataset, which can be used to localize the similarity comparisons. This method has multiple advantages: first, we can determine spatial correspondences between images by using the discovered patches as detectors; second, the regions are by nature common in gradient appearance; and lastly, the discovered regions may provide implicit (albeit noisy) pose alignment.

First, we present an approach to interactive classification that leverages localized similarity comparisons and does not rely on part or attribute vocabularies. We discover a set of discriminative, localized and roughly aligned regions for this fine-grained visual categorization task. Second, we provide a quantitative analysis of how human users respond differently to nonlocalized versus localized perceptual similarity comparisons. Finally, we demonstrate that localized similarity comparisons are more intuitive for users to perform, and that by using independent localized metrics we can improve categorization accuracy over using a single nonlocalized metric.

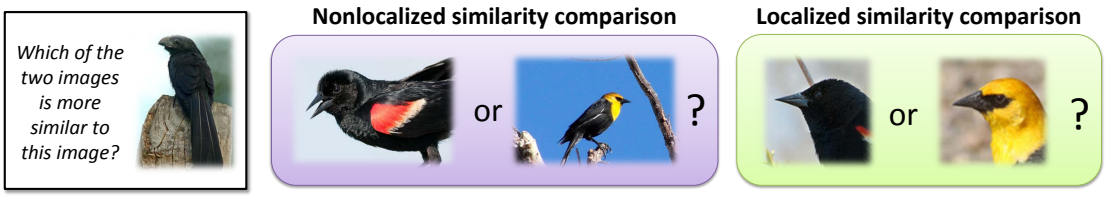


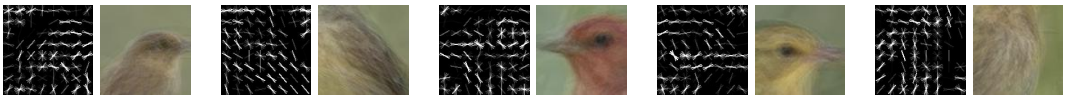
Figure 4.3. Localized Similarity Comparisons for Interactive Categorization. We use perceptual similarity metrics learned from localized comparisons to perform interactive categorization. By directing the user's attention to localized and roughly aligned regions, we aim to reduce both overall human effort required for categorization as well as improve performance over using nonlocalized comparisons and metrics.

Incorporating Multiple Similarity Metrics

We consider an extension in which similarity can be decomposed into multiple similarity metrics. These metrics can represent different *visual traits*; it is intended for these traits to be broadly applicable to a wide range of basic-level categories, such as similarity in terms of color, shape, or texture. Specifically, we consider the case where these traits correspond to localized discriminative regions.

Our system can be modified to support two types of similarity comparisons: nonlocalized and localized (see Figure 4.5). In the former, the images in the display each show the whole uncropped object. For the latter, users are asked to make a localized judgment of similarity, and all images in the grid are localized with respect to a *region* r, drawn from a set of discriminative regions $\mathscr{R}$. We define a region as a visually discriminative and recurring object part that does not have to be semantically defined or meaningful. In practice, it is a spatially localized and roughly aligned template derived from an associated descriptor (see Figure 4.4).

Our system can support the use of multiple similarity metrics S^r, $r \in 1 \ldots R$ that are represented at test time as different questions, where we direct the user's attention to specific visual traits (or localized regions). At train time, we obtain a separate embedding $\mathbf{Z}^1 \ldots \mathbf{Z}^R$ for each trait (using similarity questions that are targeted toward a specific trait),



Region 1 Region 13 Region 21 Region 23 Region 39

Figure 4.4. Discovering Discriminative Regions. We discover a set of discriminative regions (Section 4.4) and select a subset to use in our experiments, each visualized above as a HOG template alongside the averaged image of the highest confidence positive detections for that corresponding detector.

yielding multiple similarity matrices $S^1 \dots S^R$.

At test time at each timestep t, we pick both a trait r and display of images D that is likely to provide the most information gain. This amounts to finding the trait that can produce the most balanced clustering according to the current weights w_k^t. Computation of updated class probabilities occurs identically to the procedure described in Section 4.3.1, with a slightly modified update rule that replaces Eq 4.11:

$$w_k^{t+1} = p(u_{t+1}|\mathbf{z}_k^r) w_k^t = \frac{\phi(S_{ik}^r)}{\sum_{j \in D} \phi(S_{jk}^r)} w_k^t. \tag{4.18}$$

Here, we update weights w_k^{t+1} according to the similarity matrix S^r of the selected trait r.

Incorporating Localization Information

In order to utilize multiple similarity metrics with localization, we must handle instance-level variations, specifically the presence or visibility of certain pose-aligned parts in the image. In this section, we describe how we automatically obtain the set of discriminative regions to localize similarity comparisons (Section 4.4) and how we choose which images and regions to show in the display (Section 4.4).

Discovering Discriminative Regions

In order to highlight the same localized region across images for performing localized similarity comparisons, we require instance-level region correspondences.

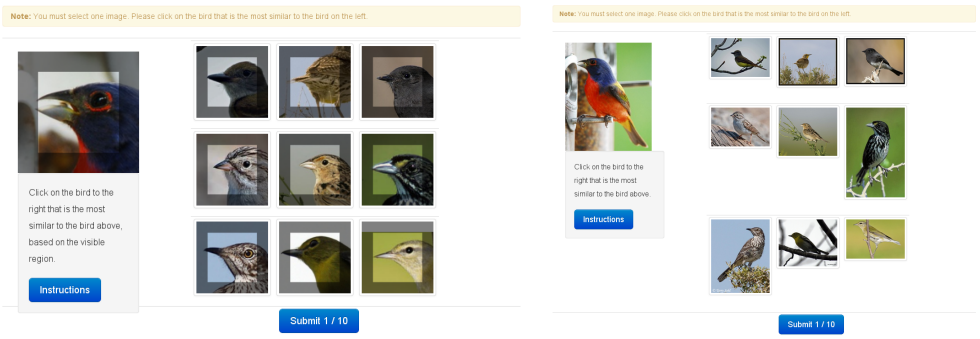
We use the unsupervised approach of Singh *et al*. [92] to discover a set of mid-level discriminative visual representations that are localized and roughly pose aligned. At test time, we can use these templates as part detectors that are evaluated on input images in a sliding window manner.

The initial candidate regions are extracted at random from uncropped images across multiple categories. To iteratively train the discriminative classifiers, we assign the positive set to consist of training examples belonging to a single basic-level category, while the negative set consists of images from all other categories, drawn from the PASCAL VOC dataset [29].

Display Model

It is likely that the localized regions discovered in Section 4.4 may not be present in certain images; this corresponds to a low detection score for a particular region detector. As such, we modify the display model of [114] to take part presence into account. Intuitively, for a particular region r, we wish to include images in the display that are highly likely to contain that localized region. Recall that we have a set $\mathscr{R}$ of discriminative regions. For a given image x_k in the training set, we model the probability the region $r \in \mathscr{R}$ is present in x_k as $p(v_k|r,x_k)$. In practice, this is determined by applying a sigmoid function to the output of the region detector. The γ parameter is learned on a validation set [82].

In selecting images for the display, we employ the approximate solution described in [114, 31, 39], which groups the images into clusters to ensure that each image in the display is equally likely to be selected, maximizing the information gain in terms of the entropy of $p(c, \mathbf{z}_k^r, U_t|x)$. For the display, we thus pick the image within the cluster with the highest mass as weighted by the region presence probability $w_k^t p(v_k|r,x_k)$.



(a) Localized comparison (b) Nonlocalized comparison

Figure 4.5. Comparing Localized and Nonlocalized Comparisons. Interfaces for collecting 4.5(a) localized and 4.5(b) nonlocalized comparisons.

4.5 Dataset and Implementation Details

We perform experiments on *CUB-200-2011* [113], which contains 200 bird classes with roughly 60 images per class. We maintain the training/testing split—only training images are seen in the data collection phase and are used to generate the embedding. Test images are considered as out-of-sample input to the interactive categorization system.

Collecting Nonlocalized Comparisons

To collect the similarity comparisons, we created an interface (Fig. 4.2) that displays a reference image along with a grid of 3×3 images. Amazon Mechanical Turk workers are asked to select all the images in the grid that clearly belong to a different species, as compared to the reference image. Images for each task are sampled at the category level without replacement, such that no two images belong to the same category. Additional observations regarding how the collected data impacts the embedding generation are discussed in Section 4.6.1.

Collecting Localized Comparisons

For learning the perceptual metrics, we collect additional localized similarity comparisons using Amazon Mechanical Turk. Collecting similarity judgments densely over all training images for each region would be an expensive and costly process; instead, we sample images for the displays from the distribution $p(v_k|r,x_k)$, such that noisy detections with low $p(v_k|r,x_k)$ are less likely to be selected for annotation. For each region r, we collect localized similarity comparisons using the GUI in Figure 4.5(a), in which the display consists of a grid of $G = 9$ images. Some context around each region is included.

Discriminative Region Vocabulary

In generating the set of discriminative regions, we assume that we are provided with ground truth object bounding boxes in both training and testing. We only keep discovered patches that have sufficient overlap (50%) with the ground truth object bounding box [58]. This eliminates many noisy detections that fire in the image background, resulting in 106 localized and roughly aligned regions. We also wish to ensure sufficient diversity in the regions used; consequently, we apply agglomerative clustering to reduce the set of 106 discovered regions to 23 region clusters, and we manually select 5 diverse and representative regions from different clusters to comprise $\mathscr{R}$ and to use in our experiments (see Figure 4.4). In practice, one could double the size of $\mathscr{R}$ by mirroring the regions to ensure left/right aspect coverage.

In Figure 4.6, we visualize the discriminative regions as the averaged image of the highest confidence positive detections for that corresponding region detector.

Figure 4.6. Discriminative Regions. The 106 discovered discriminative regions. We select 5 to use in our experiments.

Computer Vision Features and Learning

We use multiclass classifiers to initialize $p(c, \mathbf{z}|x)$, extracting color/grayscale SIFT features and color histograms with VLFEAT [105] that were combined with spatial pyramids. We trained 1-vs-all SVMs using LIBLINEAR [30].The classification scores are used to update w_k^0 according to Eq 4.15. At test time, we display a ranked list of classes based on the posterior probabilities, from which users can verify the class of the input image.

We also compare to a method that uses Fisher vector encodings (FVs) with features extracted from the object bounding boxes, which has been demonstrated to improve FGVC accuracy over other computer vision algorithms [40]. We extract SIFT descriptors and use a 256-visual words GMM, applying an $L2$-normalization on the Fisher vectors and learning a linear SVM with VLFEAT [105], yielding an average classification accuracy on the testset of 34.76%, compared to 19.4% with SIFT/color features.

4.6 Experiments

In Section 4.6.1, we describe how the embeddings are generated. In Section 4.7, we observe how human perception differs between localized and nonlocalized similarity judgments. In Section 4.6.3, we present our results on interactive classification.

4.6.1 Embedding Generation

Nonlocalized Metric: Using a set of triplets generated from our collected similarity comparisons, we are able to learn an embedding (Fig. 4.8) of N nodes, where N=200 is the number of classes. To better understand the tradeoff between dimensionality r and embedding accuracy, we compute the generalization error as we sweep over the number

of dimensions. The generalization error measures the percentage of held-out similarity triplets satisfied in three-fold cross validation. With this method, we empirically estimate r=10 as sufficient for minimizing generalization error (see Figure 4.7.

In Figure 4.8, various clusters of classes are highlighted. We observe that visually similar classes tend to belong to coherent clusters within the embedding, for example, the gulls, large black birds, and small brown striped birds. However, we also note that certain species that are dissimilar to the other birds tend to fall in their own cluster, towards the upper left portion of the embedding.

An embedding at the category level does not characterize intraclass variation, which can be high due to differences in gender, age, season, *etc*. Instead, this is handled through the noisy user model (Eq 4.8). While our method does not inherently require it, learning a similarity metric at the category level requires much fewer annotations and still gives a reasonable metric of similarity. In our experiments, we used roughly $93,000$ triplets out of a possible 8 million to generate a category-level embedding. At the instance level, this would be equivalent to collecting over 2 billion triplets.

Localized Metrics: or each localized region (Figure 4.4), we generate triplets from similarity comparisons (Section 4.5) to learn an independent localized embedding of N nodes and of dimensionality d for each region r. The comparisons are collected at the instance level; to learn each embedding, we pool over instances in each class, such that we obtain an embedding of $N = C = 200$ and $d = 10$ [114]. This enables us to generate a similarity matrix $S^r \in C \times C$ for each region r. The metric is learned independently from all other regions; see Figure 4.10 and the supplemental material for visualizations of the embeddings.

This pooling step helps to mitigate the effects of noise in both user similarity responses and region detection, and we find that we do not need to filter any noisy user responses from training in order to learn the embeddings. By pooling over classes, we

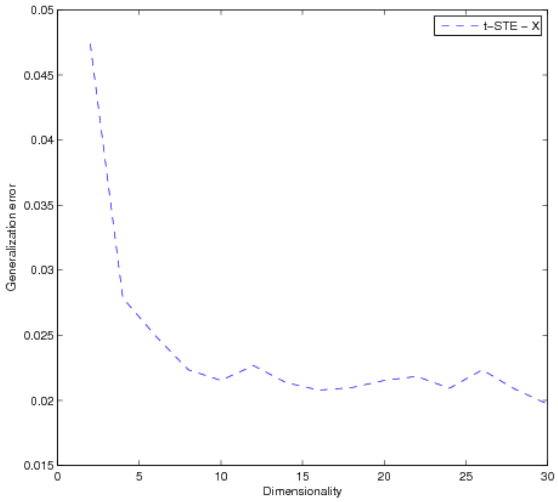


Figure 4.7. Embedding Generalization Error. We observe the generalization error for an embedding as we sweep over the number of dimensions. The generalization error is the percentage of triplet constraints that are not satisfied in three-fold cross validation. At dimensionality $r = 10$, the reduction in error stabilizes at roughly 2%.

assume that the visual appearance of parts are coherent within a subcategory; however in reality, there is intraclass variation due to differences in gender, age, season, *etc*. While we do not directly address this, our user response model is able to account for noise in user responses.

In Figures 4.9, 4.10, 4.11, 4.12, and 4.13, we visualize the first two dimensions of the embeddings for the 5 localized discriminative regions used in our experiments.

4.6.2 Using Nonlocalized Similarity Metrics

We present our results for interactive classification using the learned perceptual metric for class similarity in Figures 4.15 and 4.16. Qualitative examples of results are presented in Figures 4.17(a) and 4.17(b). At test time, a user is shown a display of 3×3 images and asked to select the bird that is most similar to the input class (see Figure 4.14). The input image is drawn from the test set, and the display images are drawn strictly from the pool of training images. As such, the system does not possess prior knowledge

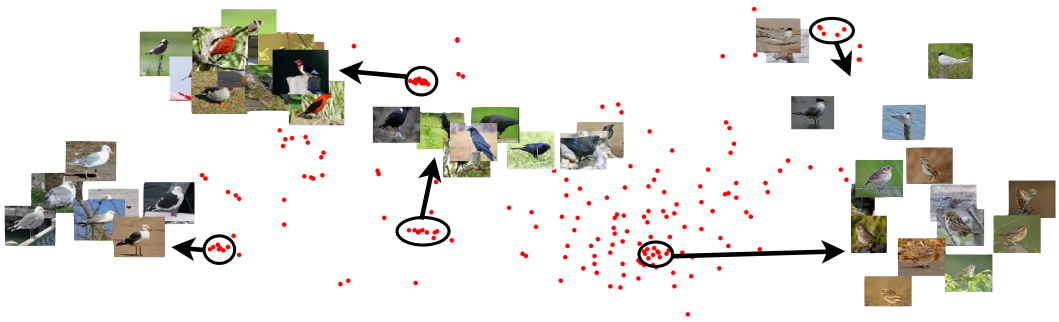


Figure 4.8. Nonlocalized Similarity Embedding. A visualization of the first two dimensions of the 200-node category-level similarity embedding. Visually similar classes tend to belong to coherent clusters (circled and shown with selected representative images).

of perceptual similarity between a given input image and any possible display of images. We use simulated user responses, which facilitates comparison to previous work as well as allows us greater flexibility in running experiments. Playback simulations based on real human responses are common in human-in-the-loop work [14, 112, 74, 75, 78] as they allow algorithmic and parameter setting choices to be explored without rerunning human experiments.

In our experiments, we measure classification accuracy as a function of the number of questions or displays the user has seen. We use the same experimental setup and evaluation criteria as [112], assuming that humans can verify the highest probability class perfectly and can stop the system early. Performance is measured as the average number of questions that a user must answer per test image to classify it correctly. Different types of questions (similarity, attribute, or part-based) may incur varying amounts of cognitive effort on the user's part, which may be reflected in differing amounts of time to answer a single question. As our test-time user responses are simulated, we compare performance based on the number of questions posed.

Similarity comparisons are advantageous compared to attribute questions. In Figures 4.15(a) and 4.15(b), we show the effects of not using and using computer vision,

Figure 4.9. Localized Similarity Embedding for Region 1.

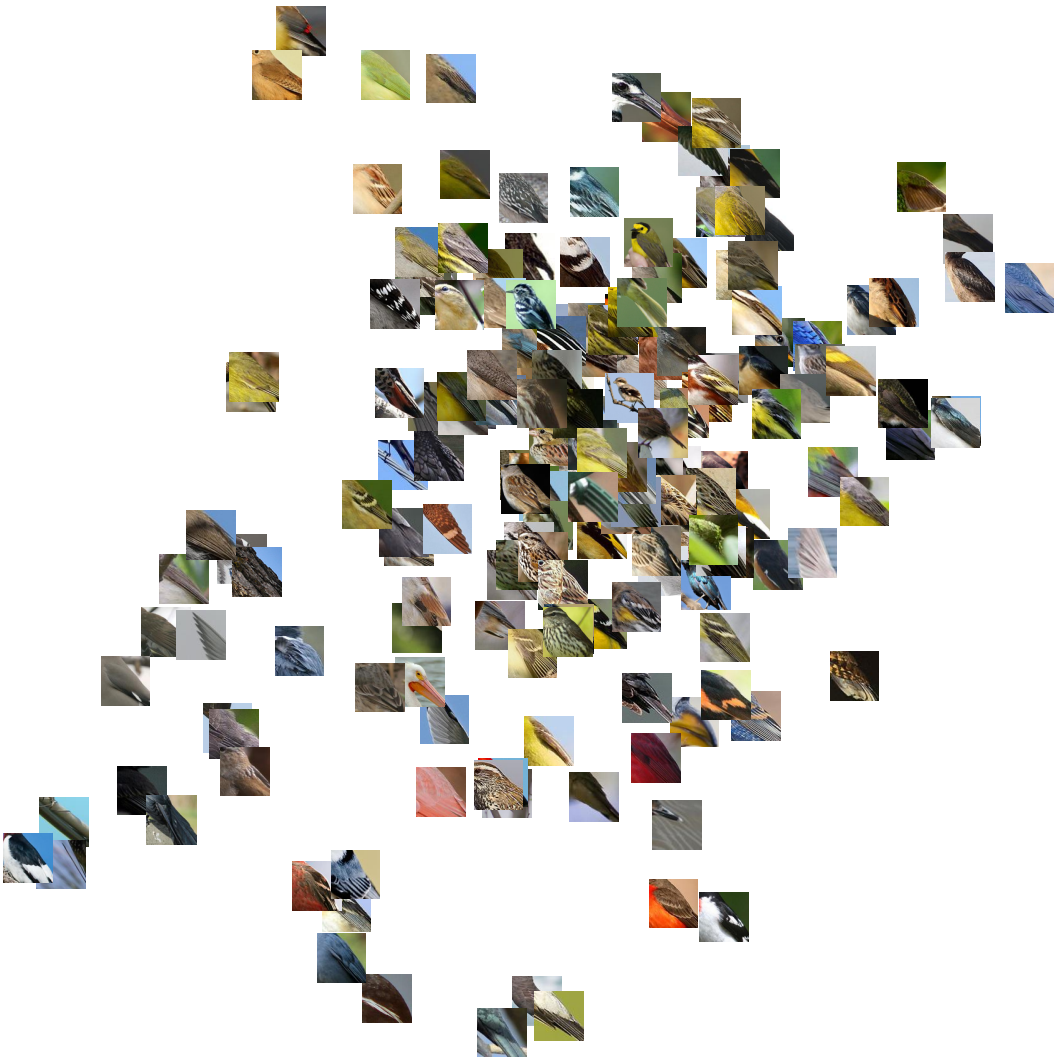


Figure 4.10. Localized Similarity Embedding for Region 13.

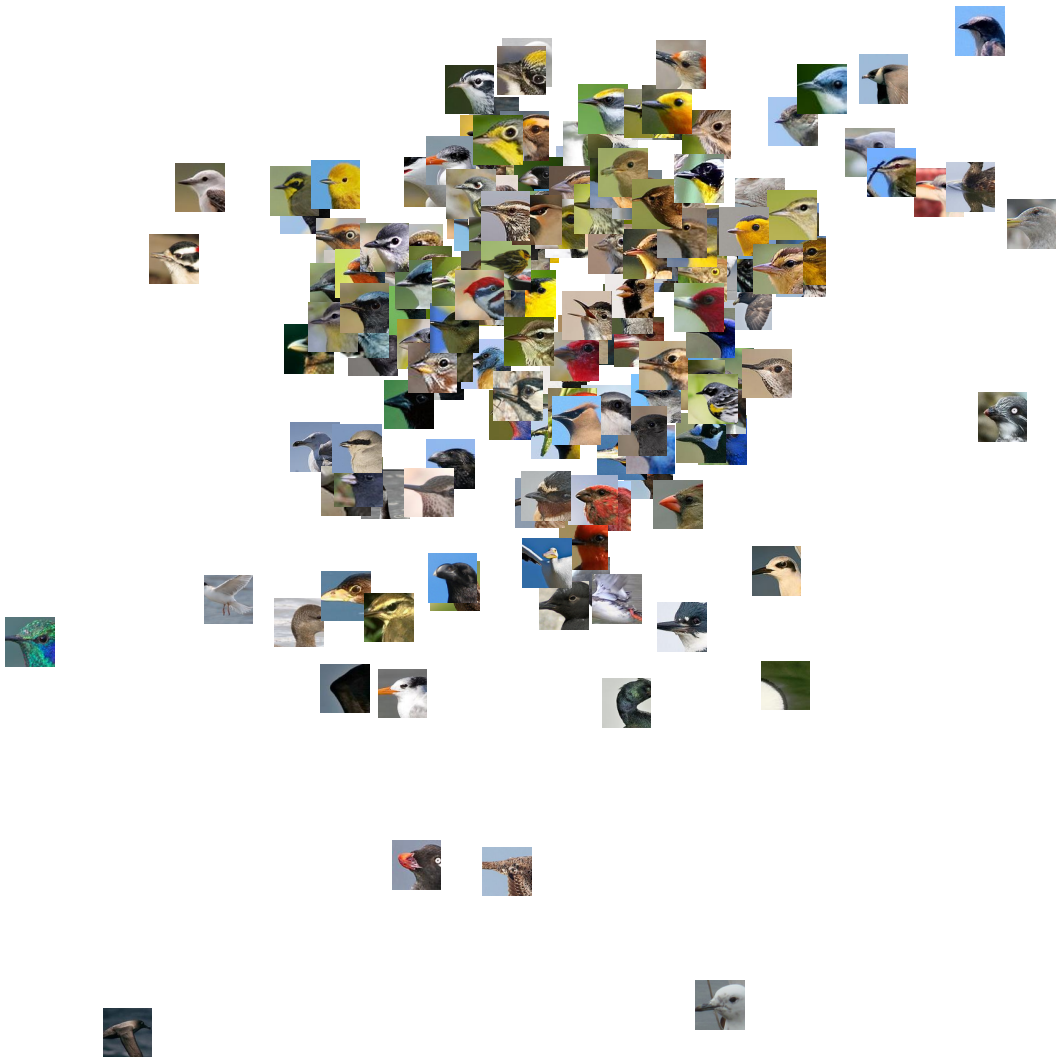


Figure 4.11. Localized similarity embedding for Region 21.

Figure 4.12. Localized similarity embedding for Region 23.

Figure 4.13. Localized similarity embedding for Region 39.

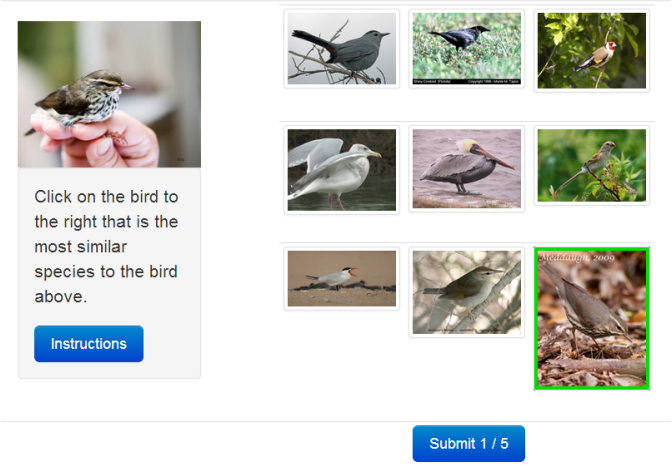
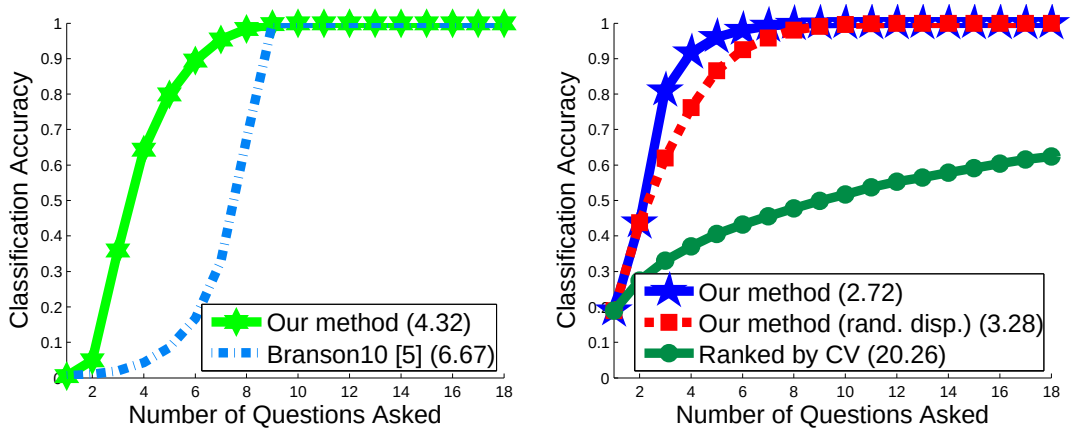


Figure 4.14. Test-Time Interface. An example of a test-time user interface for our interactive classification system.



(a) No computer vision (b) With computer vision

Figure 4.15. Observing Deterministic Users. We report the average number of questions asked per test image in parentheses for each method. 4.15(a): Our similarity-based approach requires fewer questions (4.32 vs. 6.67) than [14], which uses attributes. 4.15(b): Our display mechanism reduces user effort, as compared to randomly generated grids of images and a baseline based on the ranked classification scores.

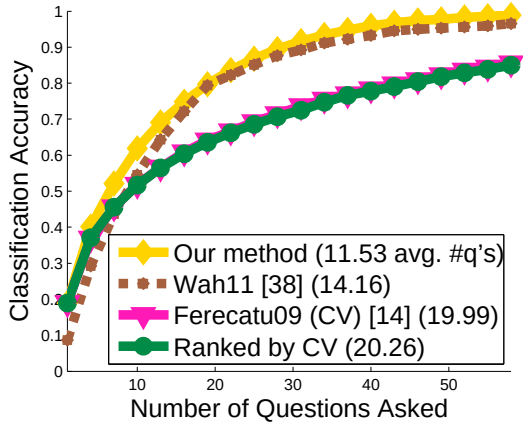


Figure 4.16. Observing Simulated Noisy Users. Our method outperforms a part and attribute-based interactive classification system [112] as well as the relevance feedback-based image retrieval system described in [39], which has been modified to utilize computer vision in initializing per-class probabilities for fairness of comparison.

respectively. We observe performance using deterministic (perfect) users (Eq 4.7) who are assumed to respond in accordance with the learned similarity metric. For a direct comparison to attribute-based approaches, we compare our method to the setting in which users answer attribute questions deterministically in accordance with expert-defined class-attribute values, as reported in [14]. We are able to reduce the average number of questions needed by 2.4.

Computer vision reduces the burden on the user. We note a similar trend when computer vision is incorporated at test time (Fig. 4.15(b)), in which users take an average of 2.7 questions per image. The addition of computer vision (Sec. 4.3.2) reduces the number of questions a user must answer in order to classify an image by 1.6 (Fig. 4.15(a)).

Intelligently selecting image displays reduces effort. We compare performance for two versions of our method: the first intelligently populates each display (Sec. 4.3.1) and the second randomly generates a display of images at each question. Using our display model, we observe that 2.7 questions are required on average, compared to 3.3 questions using a random display. We also compare to a baseline derived from classification scores

[Ranked by CV], in which the user moves down the ranked list of classes one at a time to verify the correct class. With our model, we reduce the average number of questions from 20.3 to 2.7.

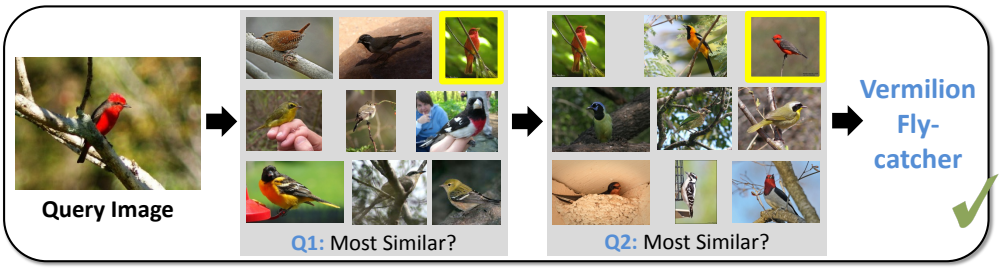
Our system is robust to user noise. In reality, assuming deterministic users is impractical, as users are likely to have subjective differences in their perceptions of similarity. To account for this, we incorporate a user response model that accounts for real human behavior (see Sec. 4.3.1). Using a validation set of query images, we pose similarity questions to real human users and estimate the parameters of a noisy user response $p(u|\mathbf{z})$ with the collected responses.

In our experiments, we simulate noisy user behavior at test time by randomly selecting answers according to the distribution $p(u|\mathbf{z})$. We compare performance directly to the results presented in [112], a system that uses part-localized computer vision algorithms as well as user feedback via attribute and part-click questions, obtaining a reduction of 2.6 questions on average (Fig. 4.16).

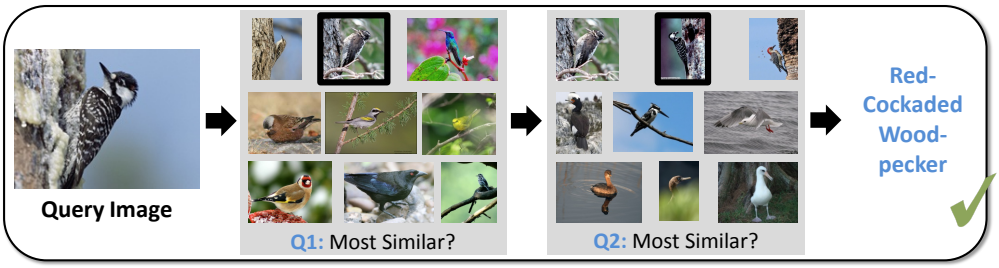
We also improve performance significantly over an implementation of [39] that uses a similarity metric generated from the L1 distances of concatenated feature vectors (see Sec. 4.5). For a fair comparison, the system in [39] is modified to use computer vision in initializing the per-class probabilities, as the query image is provided. We note that the use of the L1 distance-based metric is unable to adequately capture perceptual similarity, resulting in a high average number of questions needed for categorization.

4.6.3 Using Multiple Localized Similarity Metrics

We show our results on interactive classification in Figure 4.19; qualitative examples are presented in Figure 4.18. At test time, we use an interface similar to that used in training (Figure 4.5(a)), with the primary difference being how the reference image is displayed. The region detections in test images can be noisy, and we wish to



(a) Category similarity metric

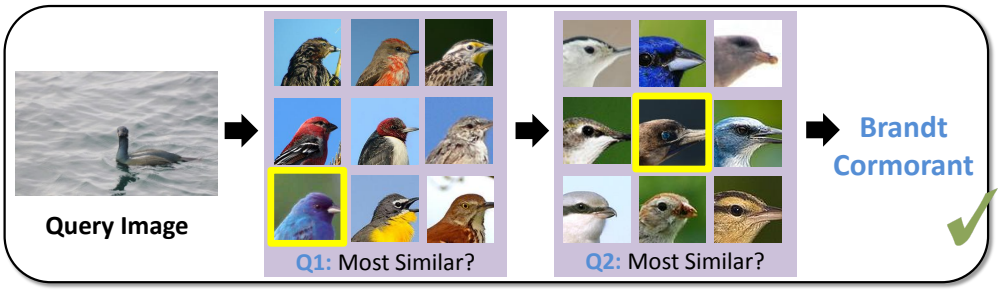


(b) Category similarity metric

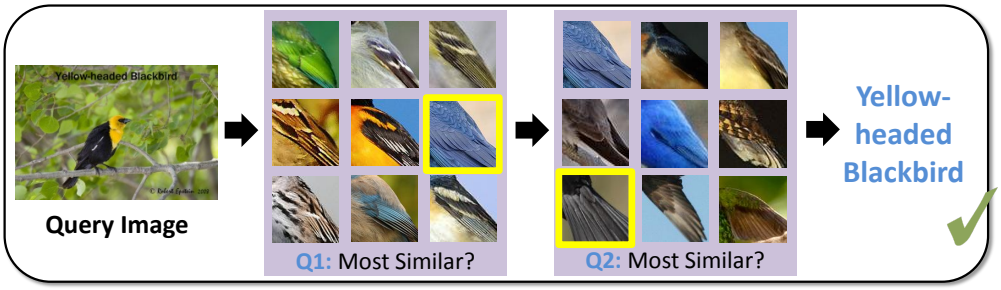
Figure 4.17. Qualitative Results for Nonlocalized Metrics learned from AMT workers (Figures 4.17(a) and 4.17(b)).

avoid highlighting an erroneous detection to the user. Instead, we show the nonlocalized reference image, and we assume that, with some cost in human effort, the user is able to mentally localize and align the corresponding region, based on the localized region highlighted in the grid images.

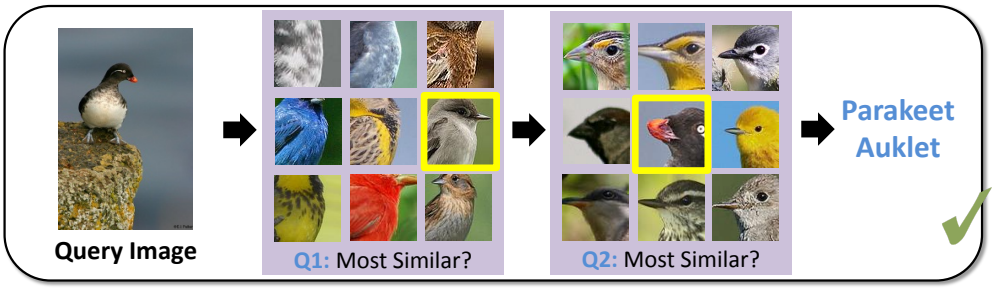
Similar to our experiments with nonlocalized metrics as described in Section 4.6.2, we use simulated user responses that allow us to compare to previous work more readily as well as explore different parameter choices. We use a model for user behavior that accounts for noisy responses, estimating parameters on a validation set of real human responses. We refer the user to Section 4.3.1 and [114] for details on the user model. Our experimental setup and performance metrics are the same as [112, 114], in which the user can verify perfectly the highest probability class, and we evaluate our system based on the average number of questions a user must answer per test image to classify it correctly.



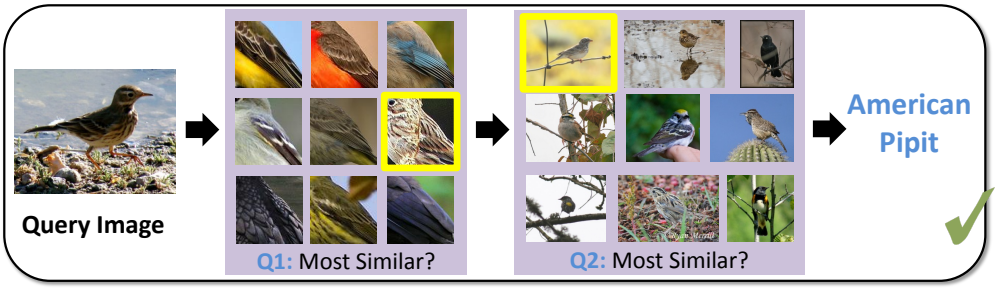
(a)



(b)

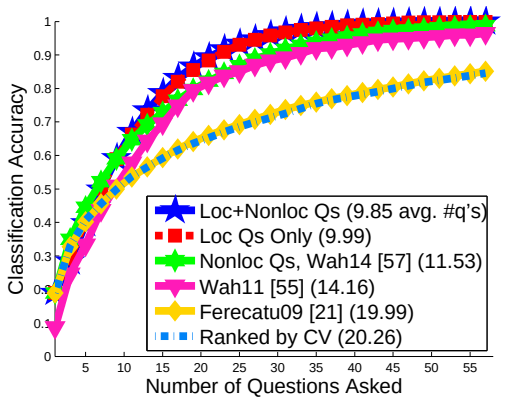


(c)

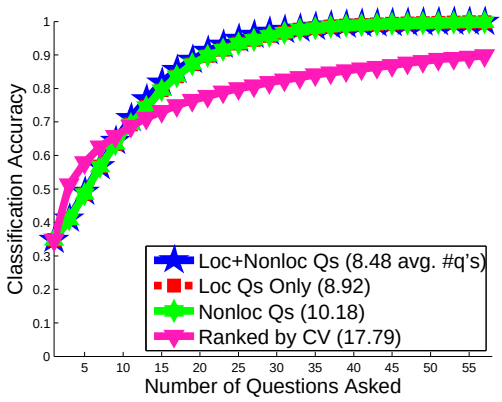


(d)

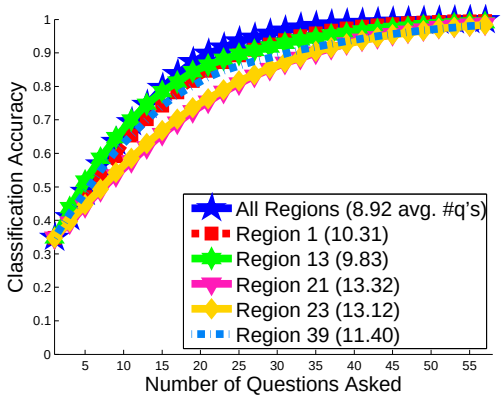
Figure 4.18. Qualitative Results for Localized Metrics. We present qualitative results for interactive categorization using our system using only the 5 localized similarity metrics in 4.18(a), 4.18(b), and 4.18(c), as well as using the localized metrics along with a nonlocalized metric 4.18(d).



(a) SIFT/color hist.



(b) Fisher vectors (FVs)



(c) Local metrics (FVs)

Figure 4.19. Interactive Categorization Results. 4.19(a): Using both localized and nonlocalized metrics outperforms using either type of metric alone. We compare to prior baselines from [114]. 4.19(b) We observe performance when the initial class probability estimates are improved by using Fisher vectors. 4.19(c): We compare performance using each localized metric separately.

It is advantageous to use localized and nonlocalized metrics together. We compare performance with simulated noisy users to [114], which uses a single nonlocalized class similarity metric, as well as to previous baselines from [114]: an interactive classification system that uses part-localized computer vision algorithms and poses semantic part click and binary attribute questions [112]; an implementation of a relevance feedback system that uses a feature-based $L1$-distance metric [39]; and a baseline derived from classification scores alone, in which the user moves down the ranked list of classes to verify the correct class. Class probabilities are initialized using the CV algorithms based on SIFT/color histograms. Results are shown in Figure 4.19(a).

By combining localized and nonlocalized metrics, we are able to classify the test images with 9.85 questions on average, compared to 9.99 by using localized metrics only and 11.53 from using the nonlocalized metric. In Figure 4.19(b), we observe a similar trend when we use FV-based computer vision estimates for initializing per-class probabilities; using both types of metrics results in 0.44 less questions on average than using only localized metrics. This performance gain is further exaggerated when we take into consideration that localized comparisons take on average 5.01 sec less time to perform than nonlocalized comparisons (Section 4.7). We also compare to the Ranked by CV baseline using the Fisher vector encoding. This baseline outperforms our system initially but fails on more difficult images, whereas our similarity-based approach is able to ultimately identify the correct class.

Localized comparisons are more informative than nonlocalized comparisons. In general, our interactive categorization system will tend to ask users to make localized comparisons in the beginning, as these questions provide the most expected information gain. As the per-class probability estimates are refined, the system will ask more nonlocalized similarity questions. We present the distribution of questions asked in the supplemental material.

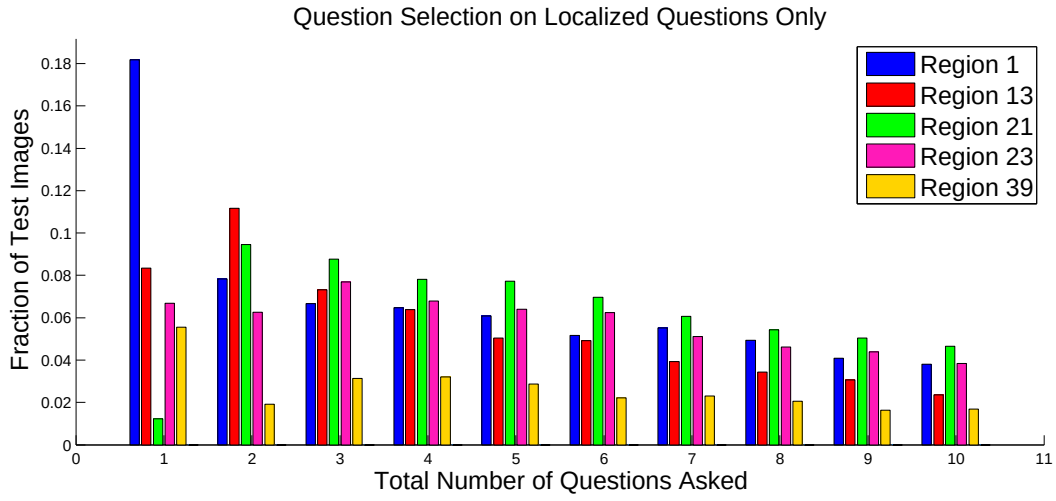
Some localized regions are more useful for categorization than others. In Figure 4.19(c) we present categorization results using the localized metrics separately. We note that using a single localized metric for Region 13 outperforms the other 4 metrics. This may suggest that the visual representation captured by Region 13, visualized in Figure 4.10, is particularly useful for discriminating bird species. Nevertheless, it can still be beneficial to use a combination of regions, as not all regions will be present in all the images. For example, using Region 13 alone produces a boost in average categorization accuracy initially for the first 15 questions; after that point, other localized metrics become more informative.

In Figure 4.20, we examine the distribution of questions asked with the interactive categorization system. Fisher vector encodings are used to initialize the computer vision estimate. We also compute the breakdown of localized similarity and nonlocalized similarity comparisons queried when both types of metrics are used.

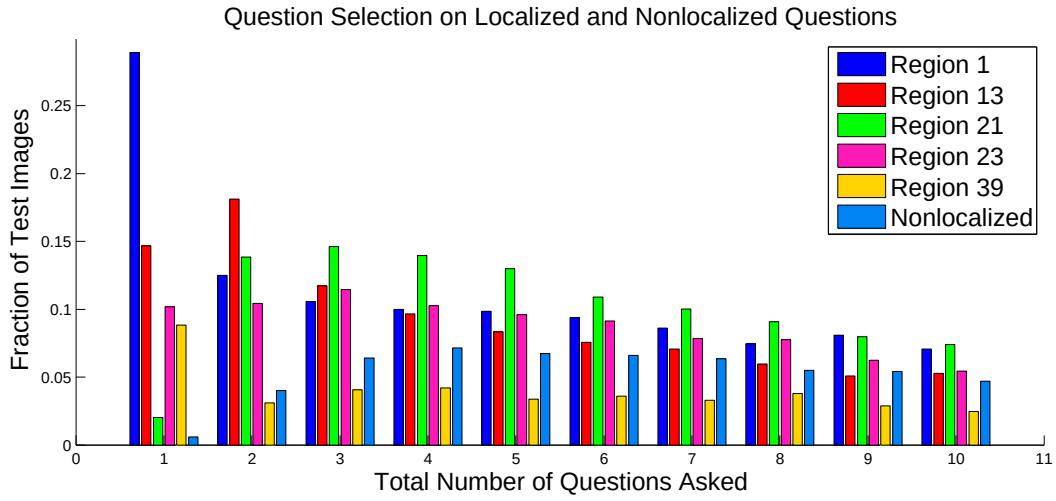
4.7 Human Perception of Similarity

We first observe empirically how users respond differently to localized compared to nonlocalized similarity questions. We generate 20 unique questions, each of which consists of 10 images total: a reference image and a grid of $G = 9$ images. Each question is seen by up to 10 AMT workers. The 200 images in the questions are selected from the top-scoring detections across the dataset for Region 1, Region 21, and Region 39 (see Figure 4.4).

We create two experiments from the set of 20 questions. One consists of localized questions only, in which the detected region is highlighted in the image (Figure 4.5(a)). The second experiment consists of the same set of questions, but the images are shown to the user as the full uncropped version (Figure 4.5(b)). Across both experiments, the images in the grid appear in the same position, and the user is asked to select a single



(a)



(b)

Figure 4.20. Question Distribution. We present the distribution of questions asked, computed as a fraction of all the test images. We simulate noisy user responses with our user response model. Figure 4.20(a): We note that localized similarity for Region 1 is the most common first question to pose to the user. Figure 4.20(b): Initially, the questions asked tend to be localized in nature, and as the per-class estimate is refined, the system will choose with greater frequency to query the user for nonlocalized similarity comparisons.

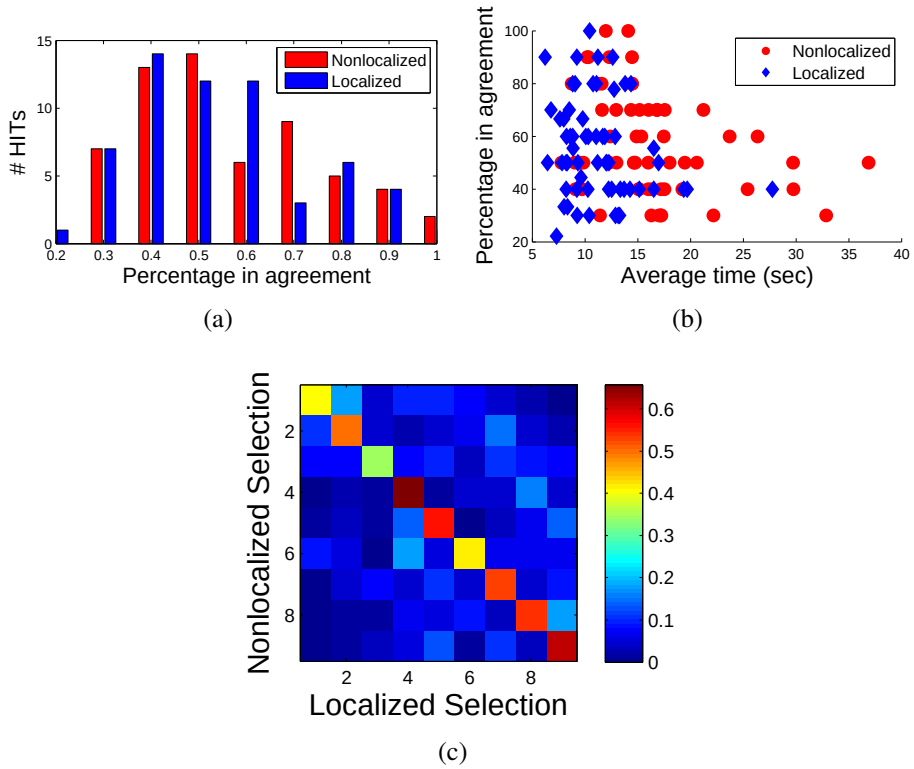


Figure 4.21. Comparing human perception of nonlocalized vs. localized similarity. 4.21(a) Histogram of HITs with a certain maximum percentage agreement (MPA) among 10 AMT worker responses; 4.21(b) MPA vs. average worker time per question; 4.21(c) co-occurrence rate of user-selected image location in the 3×3 grid, enumerated $1-9$.

image in the grid that is most similar to the reference image. The only variable that changes between experiments is how the images are displayed to the user. Due to how the discriminative regions are discovered, both experiments show images that are roughly pose aligned. We present our results in Figure 4.21.

Localized similarity comparisons require less human effort. We observe in Figure 4.21(b) the relationship between user consistency and response time. Each point corresponds to a single question. We plot the maximum percentage agreement (MPA) of the 10 users who agree on a single image in the grid, versus the average response time over those users. On average, it takes a human user less time (and with lower variance) to answer a localized comparison (11.35 ± 10.17 sec), compared to 16.36 ± 14.31 sec

for a nonlocalized comparison.

Users answer both localized and nonlocalized questions with similar consistency. In Figure 4.21(a), we plot the distribution of tasks according to the MPA. Both localized and nonlocalized similarity questions have comparable average MPA across questions (0.54 vs. 0.56, respectively), suggesting that users answer the two types of questions with a similar level of consistency. With localized regions, there still exist multiple dimensions upon which to judge similarity, such as color, shape, and pattern. As such, localization does not remove ambiguities, but does make the comparison task easier to perform.

Responses on localized questions yield different information about similarity. We present in Figure 4.21(c) the co-occurrence rates of selected image location in the 3×3 grid (1 to 9, enumerated in left-to-right, top-to-bottom order) for corresponding nonlocalized and localized questions. Selections are normalized by row. Users select the same image as the most similar for both nonlocalized and localized questions only 50.73% of the time on average, indicating that a localized similarity response provides different visual similarity information to the system. We do note that worker noise and bias can affect their responses [60]; for example, workers have a tendency to click on the lower-left portion of the grid, as it is closer to the button to advance to the next question (see Figure 4.22(c)).

In Figure 4.22, we observe AMT worker behavior on the localized similarity comparison tasks. The statistics are generated on all data collected to learn the localized similarity embeddings. A total of 79 workers provided responses.

4.8 Conclusion

We have presented an efficient approach to interactive fine-grained categorization that does not rely on experts for part and attribute vocabularies and is cost-effective to

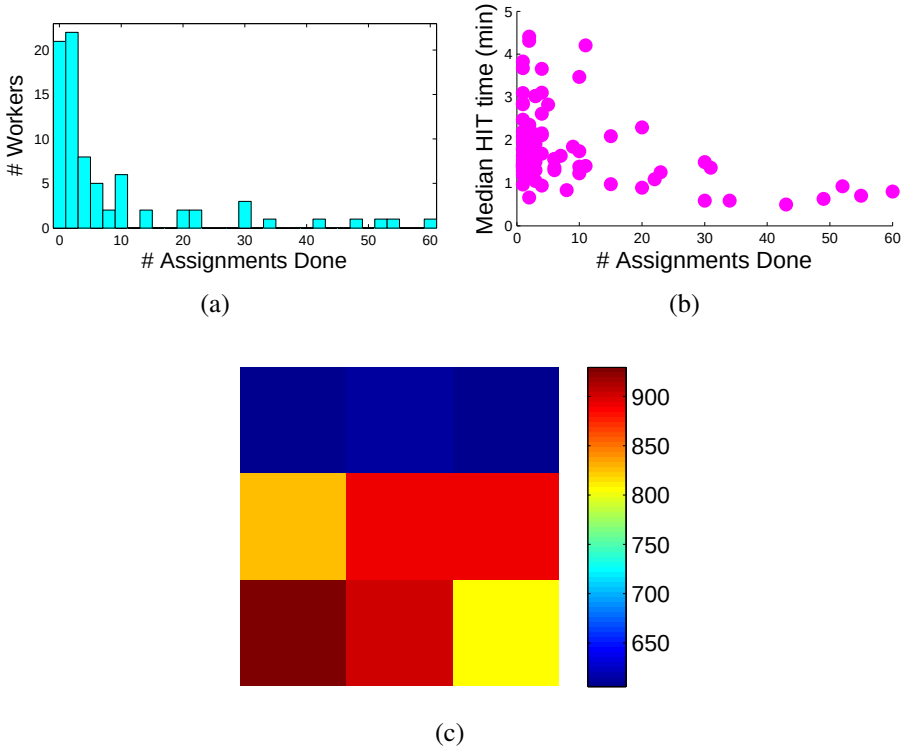


Figure 4.22. Observing AMT worker behavior. 4.22(a): We observe the number of workers who have completed a certain number of assignments. 4.22(b): Each point represents a worker, and we plot the worker's median HIT completion time versus the total number of assignments completed. 4.22(c): The distribution in the 3×3 grid of where AMT workers tend to click. The workers have a bias towards clicking at the bottom left corner of the grid; the button to advance to the next screen is located below the last row of the grid.

deploy for new basic-level categories. As users answer similarity questions for new query images, we can augment the training set and regenerate the perceptual similarity metric, enabling the system to iteratively improve as more responses are collected.

Future work could involve using these perceptual embeddings to induce attributes, parts, taxonomies, *etc*., which may be of educational value to a user. In addition, as often there exists no ground truth relative similarity judgment, it would be of interest to the computer vision community to determine best practices of eliciting consistent user similarity comparisons.

Performance of the system is affected significantly by noisy detections, which subsequently impact how accurately a user can judge localized similarity. To alleviate this, we can consider using humans to automate portions of our categorization pipeline, for instance, to clean up poor detections or select a sufficient set of useful regions to use for the system.

Acknowledgements

This chapter is primarily based on material from "Similarity Comparisons for Interactive Fine-Grained Categorization" by C. Wah, G. Van Horn, S. Branson, S. Maji, P. Perona, S. Belongie [114]. The dissertation author developed the algorithm and experiments, and wrote most of the paper.

Sections 4.4 and 4.7 are based on material from "Learning Localized Perceptual Similarity Metrics for Interactive Categorization" by C. Wah, S. Maji, and S. Belongie that has been submitted for publication. The dissertation author developed the algorithm and experiments, and wrote most of the paper.

Chapter 5

Conclusion

5.1 Final Thoughts

Fine-grained visual categorization has made incredible progress in the last few years. Three years ago, an initial baseline for classification accuracy on the CUB-200-2011 benchmark [113] was reported as 10.3% using uncropped images, and 17.3% using ground truth part locations. As of last year, recent works [40, 16, 8, 126, 7, 44] have demonstrated significantly improved accuracy; the highest of which reports 57.84% [44] mean accuracy using ground truth part annotations. Furthermore, deep convolutional features [51, 23] have reported impressive performance for various computer vision problems; the deep convolutional model DeCAF [23] sets the current bar for mean classification accuracy on the dataset at 58.75% using object bounding boxes and 64.96% using part locations.

These results are highly encouraging and demonstrate that computer vision algorithms have come a long way in terms of automatic fine-grained categorization. As research in fine-grained visual categorization continues to advance, the task will become increasingly automated, until eventually we will no longer need humans in the loop.

For now, humans can still be a key part of the categorization loop. For example, we must obtain annotated training data in order to extend existing categorization systems

to support new basic-level categories. Through interactivity, we are able to simultaneously collect additional training data and incrementally improve our models and algorithms, while providing a useful service to users. Moreover, interaction can be beneficial and desired for the user, for either educational (*e.g.* learning about bird species) or exploratory purposes (*e.g.* in searching for visually similar shoes). Until we have pushed performance of automatic classification algorithms to acceptable levels for deployment to real-life applications, we can still identify a role for interactivity in the classification process.

5.2 Future Directions

We enumerate several possible directions for future work.

- *Extensions to other basic-level categories.* We are collaborating with various organizations and institutions to collect additional datasets and explore using similarity comparisons for other basic-level categories.

- *Incorporating human-computer interaction models for collecting user input.* We hope to apply methods from the human-computer interaction community to the data annotation process. This involves investigating best practices for creating annotation tasks and querying users for different forms of input in a cost-effective manner.

- *Automating a pipeline for deploying interactive categorization paradigms.* Deploying our paradigms to new basic-level categories requires some manual intervention in discriminative region selection, cleaning up annotated data, etc. It would be advantageous to develop an automatic pipeline to use humans for these tasks, facilitating the deployment process.

Appendix A

CUB-200-2011 Dataset

CUB-200-2011 is an extended version of CUB-200 [120], a challenging dataset of 200 bird species. The extended version roughly doubles the number of images per category and adds new part localization annotations. All images are annotated with bounding boxes, part locations, and attribute labels. Images and annotations were filtered by multiple users of Mechanical Turk. We introduce benchmarks and baseline experiments for multi-class categorization and part localization.

A.1 Introduction

Bird species classification is a difficult problem that pushes the limits of the visual abilities for both humans and computers. Although different bird species share the same basic set of parts, different bird species can vary dramatically in shape and appearance (*e.g.*, consider pelicans vs. sparrows). At the same time, other pairs of bird species are nearly visually indistinguishable, even for expert bird watchers (*e.g.*, many sparrow species are visually similar). Intraclass variance is high due to variation in lighting and background and extreme variation in pose (*e.g.*, flying birds, swimming birds, and perched birds that are partially occluded by branches).

It is our hope that CUB-200-2011 will facilitate research in subordinate categorization by providing a comprehensive set of benchmarks and annotation types for one

particular domain (birds). We would like to cultivate a level of research depth that has thus far been reserved for a few select categories such as pedestrians and faces. Focusing on birds will help keep research more tractable from a logistical and computational perspective. At the same time, we believe that many of the lessons learned (in terms of annotation procedures, localization models, feature representations, and learning algorithms) will generalize to other domains such as different types of animals, plants, or objects.

A.2 Dataset Specification and Collection

Bird Species: The dataset contains 11,788 images of 200 bird species. Each species is associated with a Wikipedia article and organized by scientific classification (*order, family, genus, species*). The list of species names was obtained using an online field guide[1]. Images were harvested using Flickr image search and then filtered by showing each image to multiple users of Mechanical Turk [119]. Each image is annotated with bounding box, part location, and attribute labels. See Fig A.1 for example images and Fig A.6 for more detailed dataset statistics.

Bounding Boxes: Bounding boxes were obtained using the interface in Fig. A.4.

Attributes: A vocabulary of 28 attribute groupings (see Fig A.2(b)) and 312 binary attributes (*e.g.*, the attribute group *belly color* contains 15 different color choices) was selected based on an online tool for bird species identification[2]. All attributes are visual in nature, with most pertaining to a color, pattern, or shape of a particular part. Attribute annotations were obtained for each image using the interface in Fig. A.5.

Part Locations: A total of 15 parts (see Fig A.2(a)) were annotated by pixel location and visibility in each image using the GUI shown in Fig A.3(a). The "ground

[1] http://www.birdfieldguide.com
[2] http://www.whatbird.com

truth“ part locations were obtained as the median over locations for 5 different Mechanical Turk users per image.

A.3 Applications

CUB-200-2011 has a number of unique properties that we believe are of interest to the research community:

Subordinate category recognition: Methods that are widely used on datasets such as Caltech-101 [45] (*e.g.*, lossy representations based on histogramming and bag-of-words) are often less successful on subordinate categories, due to higher visual similarity of categories. Research in subordinate categorization may help encourage development of features or localization models that retain a greater level of discriminative power.

Multi-class object detection and part-based methods: Part-based methods have recently experienced renewed interest and success [36]. Unfortunately, availability of datasets with comprehensive part localization information is still fairly limited. Additionally, whereas datasets for image categorization often contain hundreds or thousands of categories [45, 20], popular datasets for object detection rarely contain more than 20 or so categories [29] (mostly due to computational challenges). Methods that employ shared part models offer great promise toward scaling object detection to a larger number of categories. CUB-200-2011 contains a collection of 200 different bird species that are annotated using the same basic set of parts, thus making it uniquely suited toward research in shared part models.

Attribute-based methods: Attribute-based recognition is another form of model sharing that has recently become popular. Most existing datasets for attribute-based recognition (*e.g.* Animals With Attributes [54]) do not contain localization information. This is an obstacle to research in attributed-based recognition, because visual attributes are often naturally associated with a particular part or object (*e.g.* blue belly or cone-shaped

beak).

Crowdsourcing and user studies: Annotations such as part locations and attributes open the door for new research opportunities, but are also subject to a larger degree of annotation error and user subjectivity as compared to object class labels. By releasing annotations from multiple MTurk users per training image, we hope to encourage research in crowdsourcing techniques for combining annotations from multiple users, and facilitate user studies evaluating the reliability and relative merit of different types of annotation.

A.4 Benchmarks and Baseline Experiments

We introduce a set of benchmarks and baseline experiments for studying bird species categorization, detection, and part localization:

1. **Localized Species Categorization:** *Given the ground truth part locations, assign each image to one of 200 bird classes*. This benchmark is intended to facilitate studies of different localization models (*e.g.*, to what extent does localization information improve classification accuracy?), and also provide greater accessibility to existing categorization algorithms. Using RGB color histograms and histograms of vector-quantized SIFT descriptors with a linear SVM, we obtained a classification accuracy of 17.3% (see Fig A.7(d)).

2. **Part Localization:** *Given the full, uncropped bird images, predict the location and visibility of each bird part*. We measured the distance between predicted part locations and ground truth, normalized on a per-part basis by the standard deviation over part click locations for multiple MTurk users. The maximum error per part was bounded at 5 standard deviations. This was also the error associated with misclassification of part visibility. Using HOG-based part-detectors and a

mixture of tree-structured pictorial structures, we obtained an average error of 2.47 standard deviations (by contrast, an average MTurk user should be off by 1 standard deviation). See Fig A.8 for example part localization results and their associated loss.

3. **Species Categorization/Detection:** *Using only the full, uncropped bird images, assign each image to one of 200 bird classes.* For this benchmark, one can use the method of his/her choice (*e.g.*, image categorization, object detection, segmentation, or part-based detection techniques); however, since the images are uncropped, we anticipate that the problem cannot be solved with high accuracy without obtaining some degree of localization. Detecting the most likely part configuration using a universal bird detector (as for benchmark 2) and then applying a localized species classifier (as for benchmark 1), we obtained a classification accuracy of 10.3% (see Fig A.7(b)).

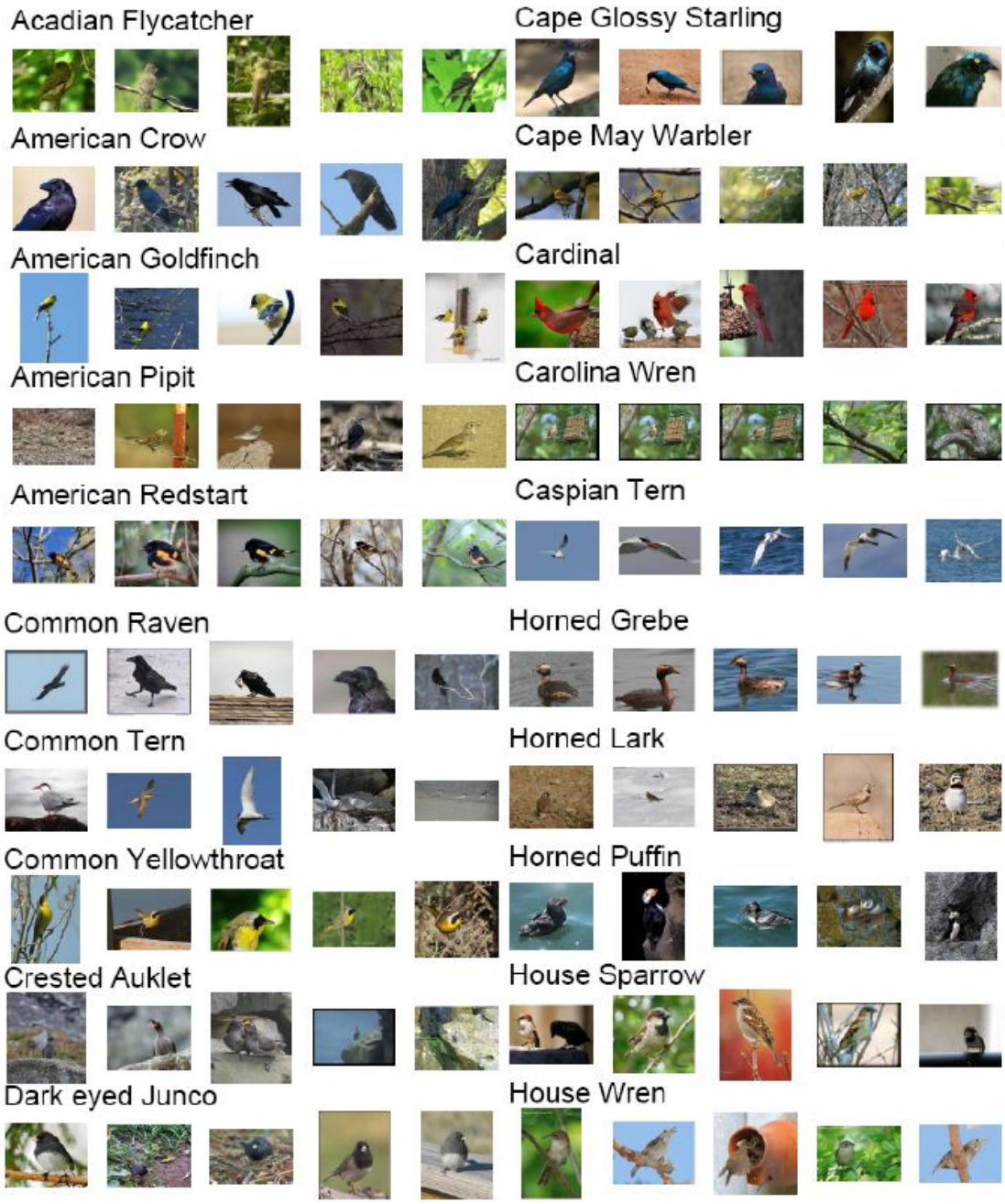
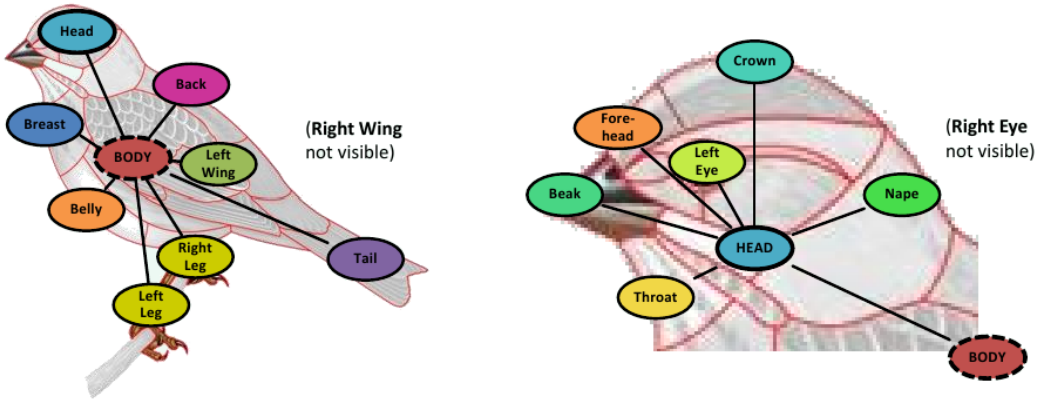


Figure A.1. CUB-200-2011 Example Images



(a) Collected Parts

Part	Attributes	Part	Attributes	Part	Attributes
Beak	*HasBillShape, HasBillColor , HasBillLength*	**Back**	*HasBackColor, HasBackPattern*	**Breast**	*HasBreastPattern, HasBreastColor*
Belly	*HasBellyPattern, HasBellyColor*	**Fore-head**	*HasForehead Color*	**Bird** (all parts)	*HasSize, HasShape*
Throat	*HasThroatColor*	**Nape**	*HasNapeColor*	**Head**	*HasHeadPattern*
Crown	*HasCrownColor*	**Eye**	*HasEyeColor*	**Leg**	*HasLegColor*
Tail	*HasUpperTailColor, HasUnderTailColor, HasTailPattern, HasTailShape*	**Wing**	*HasWingPattern, Has WingColor, HasWingShape*	**Body**	*HasUnderpartsColor, HasUpperPartsColor, HasPrimaryColor*

(b) Attribute Part Associations

Figure A.2. Collected Parts and Attributes. (a) The 15 part location labels collected for each image. (b) The 28 attribute-groupings that were collected for each image, and the associated part for localized attribute detectors.

(a) Part GUI

Figure A.3. MTurk GUI for collecting part location labels, deployed on 11,788 images for 15 different parts and 5 workers per image.

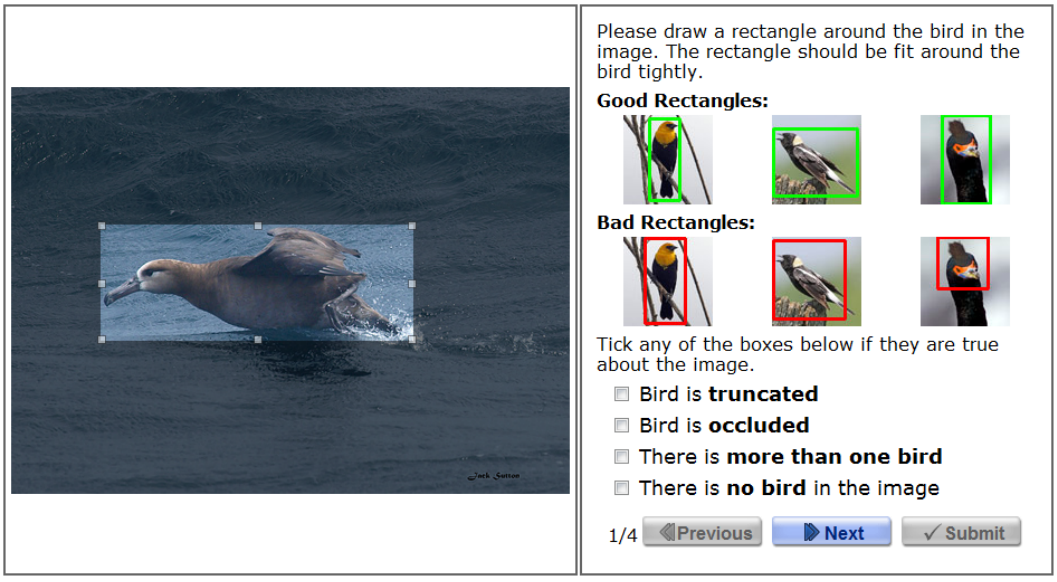


Figure A.4. MTurk GUI for collecting bounding box labels, deployed on 11,788 images.

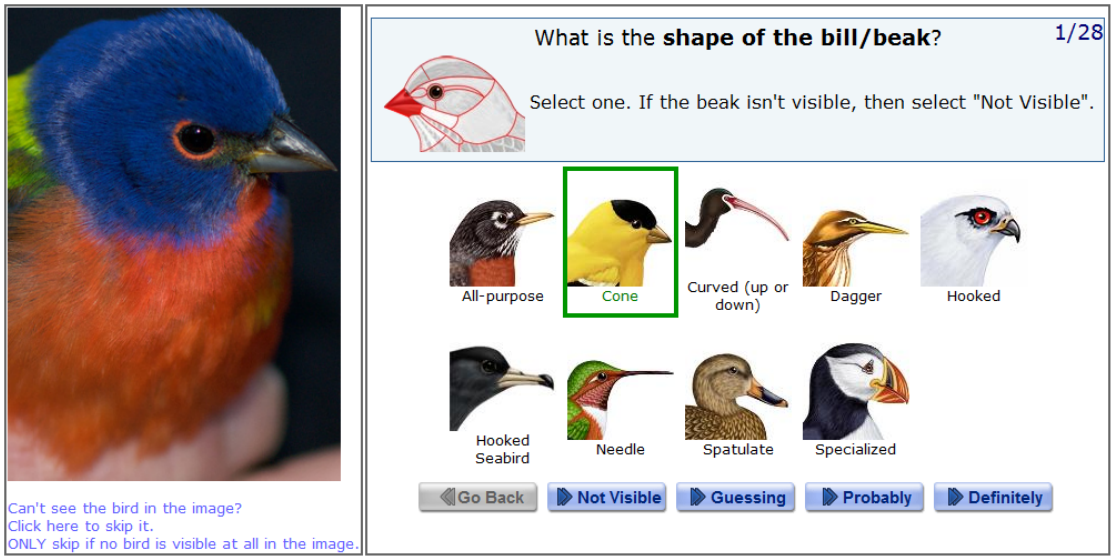
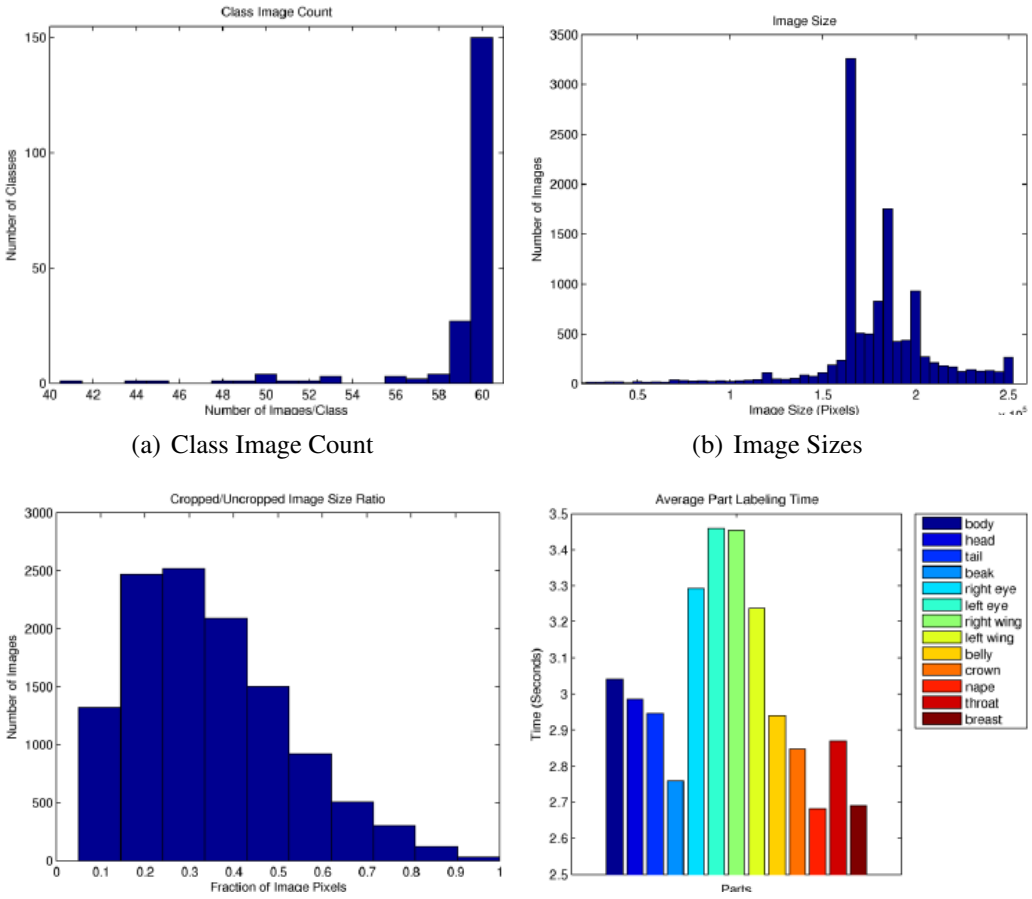


Figure A.5. MTurk GUI for collecting attribute labels, deployed on 11,788 images for 28 different questions and 312 binary attributes.



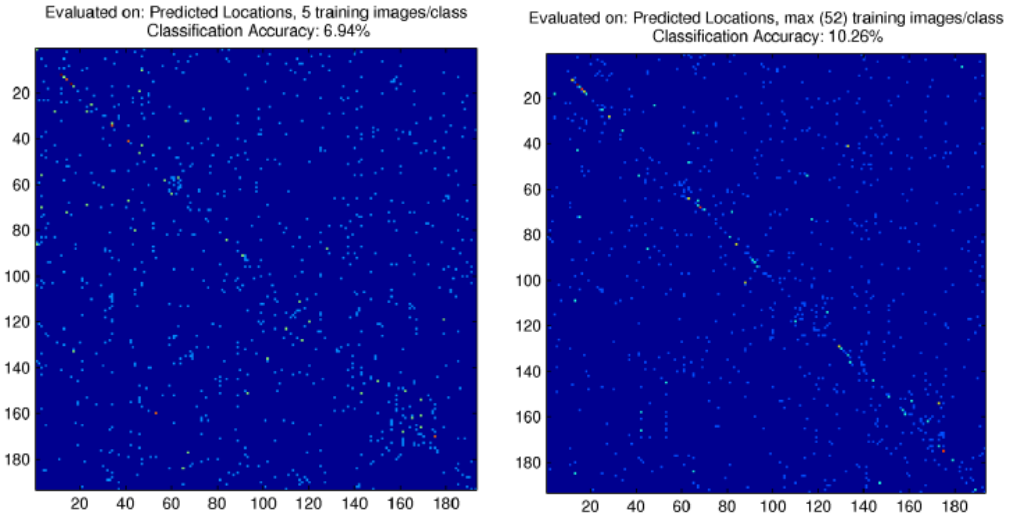
(a) Class Image Count

(b) Image Sizes

(c) Cropped/Uncropped Image Size Ratio

(d) Average Part Labeling Time

Figure A.6. Dataset Statistics. (a) Distribution of the number of images per class (most classes have 60 images). (b) Distribution of the size of each image in pixels (most images are roughly 500X500). (c) Distribution of the ratio of the area of the bird's bounding box to the area of the entire image. (d) The average amount of time it took MTurkers to label each part.



(a) Predicted Locations, 5 Images/Class

(b) Predicted Locations, 52 Images/Class

(c) Ground Truth Locations, 5 Images/Class

(d) Ground Truth Locations, 52 Images/Class

Figure A.7. Categorization Results for 200-way bird species classification. The top 2 images show confusion matrices when using a universal bird detector to detect the most likely location of all parts and then evaluating a multiclass classifier. The bottom 2 images show confusion matrices when evaluating a multiclass classifier on the ground truth part locations. The 2 images on the left show results with 5 training images per class, and the images on the right show results with 52 training images per class.

Figure A.8. Example Part Detection Results, with good detection results on the left and bad detection results on the right. A loss of 1.0 indicates that the predicted part locations are about as good as the average MTurk labeler.

Bibliography

[1] Sameer Agarwal, Jongwoo Lim, Lihi Zelnik-Manor, Pietro Perona, David J. Kriegman, and Serge Belongie. Beyond pairwise clustering. In *CVPR*, 2005.

[2] Charu C. Aggarwal. Towards meaningful high-dimensional nearest neighbor search by hci. In *ICDE*, 2002.

[3] P. Frazier B. Jedynak and R. Sznitman. Twenty questions with noise: Bayes optimal policies for entropy loss. *Journal of Applied Probability*, 49(1):114–136, 2012.

[4] Boris Babenko, Steven Branson, and Serge Belongie. Similarity metrics for categorization. In *CVPR*, 2009.

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

[6] T. Berg, A. Berg, and J. Shih. Automatic attribute discovery and characterization from noisy web data. In *ECCV*, 2010.

[7] T. Berg, J. Liu, S. Lee, M. Alexander, D. Jacobs, and P. Belhumeur. Birdsnap : Large-scale fine-grained visual categorization of birds. In *CVPR*, 2014.

[8] Thomas Berg and Peter N Belhumeur. POOF : Part-based one-vs-one features for fine-grained categorization, face verification, and attribute estimation. In *CVPR*, 2013.

[9] M. Beynon, D. Cosker, and D. Marshall. An expert system for multi-criteria decision making using Dempster Shafer theory. *Expert Systems with Applications*, 20(4), 2001.

[10] Irving Biederman, Suresh Subramaniam, Moshe Bar, Peter Kalocsai, and Jozsef Fiser. Subordinate-level object classification reexamined. *Psychological research*, 1999.

[11] L. Bourdev and J. Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *ICCV*, 2009.

[12] Steve Branson, Grant Van Horn, Catherine Wah, Pietro Perona, and Serge Belongie. The ignorant led by the blind: A hybrid human-machine vision system for fine-grained categorization. *International Journal of Computer Vision (IJCV)*, February 2014.

[13] Steve Branson, Pietro Perona, and Serge Belongie. Strong supervision from weak annotation. In *ICCV*, 2011.

[14] Steve Branson, Catherine Wah, Boris Babenko, Florian Schroff, Peter Welinder, Pietro Perona, and Serge Belongie. Visual recognition with humans in the loop. In *ECCV*, 2010.

[15] Yuning Chai, Victor Lempitsky, and Andrew Zisserman. Bicos: A bi-level co-segmentation method. In *ICCV*, 2011.

[16] Yuning Chai, Victor Lempitsky, and Andrew Zisserman. Symbiotic segmentation and part localization for fine-grained categorization. In *ICCV*, 2013.

[17] Yuning Chai, Esa Rahtu, Victor Lempitsky, Luc Van Gool, and Andrew Zisserman. Tricos. In *ECCV*, 2012.

[18] S. Changpinyo, K. Liu, and F. Sha. Similarity component analysis. In *NIPS*, 2013.

[19] Ritendra Datta, Dhiraj Joshi, Jia Li, and James Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys*, 40(2), 2008.

[20] J. Deng, W. Dong, R. Socher, L.-J. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009.

[21] Carl Doersch, Saurabh Singh, Abhinav Gupta, Josef Sivic, and Alexei a. Efros. What makes Paris look like Paris? *ACM Transactions on Graphics*, 31(4):1–9, July 2012.

[22] J. Donahue and K. Grauman. Annotator rationales for visual recognition. In *ICCV*, 2011.

[23] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. DeCAF: A deep convolutional activation feature for generic visual recognition. In *ICML*, 2014.

[24] Jeff Donahue and Kristen Grauman. Annotator rationales for visual recognition. In *ICCV*, 2011.

[25] Kun Duan, Devi Parikh, David Crandall, and Kristen Grauman. Discovering Localized Attributes for FGVC. In *CVPR*, 2012.

[26] Z. Elouedi, K. Mellouli, and P. Smets. Belief decision trees: theoretical foundations. *International Journal of Approximate Reasoning*, 28(2-3):91–124, 2001.

[27] Ian Endres, Ali Farhadi, Derek Hoiem, and David A. Forsyth. The benefits and challenges of collecting richer object annotations. In *ACVHL*, 2010.

[28] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL VOC Challenge 2009 Results. http://www.pascal-network.org/challenges/VOC/voc2009/workshop/index.html.

[29] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html, 2012.

[30] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR. *JMLR*, 2008.

[31] Y. Fang and D. Geman. Experiments in mental face retrieval. In *AVBPA*, 2005.

[32] A. Farhadi, I. Endres, and D. Hoiem. Attribute-centric recognition for cross-category generalization. In *CVPR*, 2010.

[33] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *CVPR*, 2009.

[34] Ryan Farrell, Om Oza, Vlad I. Morariu, Trevor Darrell, and Larry S. Davis. Birdlets: Subordinate categorization using volumetric primitives and pose-normalized appearance. In *ICCV*, 2011.

[35] Li Fei-Fei, R. Fergus, and P. Perona. A bayesian approach to unsupervised one-shot learning of object categories. In *ICCV*, volume 2, page 1134, 2003.

[36] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *CVPR*, 2008.

[37] P.F. Felzenszwalb and D.P. Huttenlocher. Efficient matching of pictorial structures. In *CVPR*, 2000.

[38] Marin Ferecatu and Donald Geman. Interactive search for image categories by mental matching. In *ICCV*, 2007.

[39] Marin Ferecatu and Donald Geman. A statistical framework for category search from a mental picture. *TPAMI*, 2009.

[40] E. Gavves, B. Fernando, C. Snoek, A. Smeulders, and T. Tuytelaars. Fine-grained categorization by alignments. In *ICCV*, 2013.

[41] D. Geman and B. Jedynak. An active testing model for tracking roads in satellite images. *PAMI*, 1996.

[42] Donald Geman and Bruno Jedynak. Shape recognition and twenty questions. Technical report, IN PROC. RECONNAISSANCE DES FORMES ET INTELLIGENCE ARTIFICIELLE (RFIA), 1993.

[43] R. Gomes, P. Welinder, A. Krause, and P. Perona. Crowdclustering. In *NIPS*, 2011.

[44] C. Göring, E. Rodner, A. Freytag, and J. Denzler. Nonparametric part transfer for fine-grained recognition. In *CVPR*, 2014.

[45] Gregory Griffin, Alex Holub, and Pietro Perona. Caltech-256 object category dataset. Technical Report CNS-TR-2007-001, California Institute of Technology, 2007.

[46] Jonathon Hare, Patrick Sinclair, Paul Lewis, Kirk Martinez, Peter Enser, and Christine Sandom. Bridging the semantic gap in multimedia information retrieval. In *ESWC*, 2006.

[47] Alex D. Holub, Max Welling, and Pietro Perona. Hybrid generative-discriminative visual categorization. *IJCV*, 77(1-3):239–258, 2008.

[48] J. Jeon and R. Manmatha. Using Maximum Entropy for Automatic Image Annotation. In *International Conference on Image and Video Retrieval (CIVR)*, 2004.

[49] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Fei-Fei Li. Novel dataset for fine-grained image categorization: Stanford dogs. *CVPR Workshop on Fine-Grained Visual Categorization (FGVC)*, 2011.

[50] A. Kovashka, S. Vijayanarasimhan, and K. Grauman. Actively selecting annotations among objects and attributes. In *ICCV*, 2011.

[51] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classication with deep convolutional neural networks. In *NIPS*, 2012.

[52] 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 *ECCV*, 2012.

[53] Neeraj Kumar, Alexander C. Berg, Peter N. Belhumeur, and Shree K. Nayar. Attribute and simile classifiers for face verification. In *ICCV*, 2009.

[54] C.H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, 2009.

[55] Natalia Larios, Bilge Soran, Linda G Shapiro, G Martinez-Munoz, Junyuan Lin, and Thomas G Dietterich. Haar random forest features and svm spatial matching kernel for stonefly species identification. In *ICPR*, 2010.

[56] Edith Law, Burr Settles, Aaron Snook, Harshit Surana, Luis von Ahn, and Tom Mitchell. Human computation for attributes and attribute values acquisition. In *FGVC Workshop*, 2011.

[57] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. A maximum entropy framework for part-based texture and object recognition. In *ICCV*, 2005.

[58] Y. J. Lee, A. A. Efros, and M. Hebert. Style-aware mid-level representation for discovering visual connections in space and time. In *CVPR*, 2013.

[59] A. Levin, D. Lischinski, and Y. Weiss. A closed-form solution to natural image matting. *PAMI*, 2007.

[60] Greg Little. Top, middle, or bottom? http://groups.csail.mit.edu/uid/deneme/?p=27, 2009.

[61] Jiongxin Liu, Angjoo Kanazawa, David Jacobs, and Peter Belhumeur. Dog breed classification using part localization. *ECCV*, 2012.

[62] Subhransu Maji. Discovering a lexicon of parts and attributes. In *ECCV Workshop on Parts and Attributes*, 2012.

[63] Subhransu Maji and Greg Shakhnarovich. Part annotations via pairwise correspondence. In *Human Computation Workshop*, 2012.

[64] G. Martınez-Munoz, W. Zhang, N. Payet, S. Todorovic, N. Larios, A. Yamamuro, D. Lytle, A. Moldenke, E. Mortensen, R. Paasch, et al. Dictionary-free categorization of very similar objects via stacked evidence trees. In *CVPR*, 2009.

[65] B. McFee and G. R. G. Lanckriet. Metric learning to rank. In *ICML*, June 2010.

[66] B. McFee and G. R. G. Lanckriet. Learning multi-modal similarity. *JMLR*, 2011.

[67] Thomas Mensink, Jakob Verbeek, and Gabriela Csurka. Learning structured prediction models for interactive image labeling. In *CVPR*, 2011.

[68] Carolyn B Mervis and Maria A Crisafi. Order of acquisition of subordinate-, basic-, and superordinate-level categories. *Child Development*, 1982.

[69] Richard E. Neapolitan. *Probabilistic reasoning in expert systems: theory and algorithms*. John Wiley & Sons, Inc., New York, NY, USA, 1990.

[70] M-E Nilsback and Andrew Zisserman. A visual vocabulary for flower classification. In *CVPR*, 2006.

[71] M.E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *ICCVGIP*, 2008.

[72] P. Ott and M. Everingham. Shared parts for deformable part-based models. In *CVPR*, 2011.

[73] M. Palatucci, D. Pomerleau, G. Hinton, and T. Mitchell. Zero-Shot Learning with Semantic Output Codes. In *NIPS*, 2009.

[74] D. Parikh and K. Grauman. Interactively building a discriminative vocabulary of nameable attributes. In *CVPR*, 2011.

[75] D. Parikh and K. Grauman. Relative attributes. In *ICCV*, 2011.

[76] Devi Parikh and C Lawrence Zitnick. Finding the weakest link in person detectors. In *CVPR*, 2011.

[77] Devi Parikh and C Lawrence Zitnick. Human-debugging of machines. In *NIPS Wisdom of Crowds*, 2011.

[78] Amar Parkash and Devi Parikh. Attributes for classifier feedback. In *ECCV*, 2012.

[79] Omkar Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *CVPR*, 2012.

[80] Omkar M Parkhi, Andrea Vedaldi, CV Jawahar, and Andrew Zisserman. The truth about cats and dogs. In *ICCV*, 2011.

[81] P. Perona. Vision of a visipedia. *Proceedings of the IEEE*, 98(8):1526 –1534, aug. 2010.

[82] J.C. Platt. Probabilities for SV machines. In *NIPS*, 1999.

[83] J. R. Quinlan. *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[84] M. Rohrbach et al. What helps where – and why? semantic relatedness for knowledge transfer. In *CVPR*, 2010.

[85] Eleanor Rosch. Principles of categorization. *Concepts: core readings*, 1999.

[86] Eleanor Rosch, Carolyn B Mervis, Wayne D Gray, David M Johnson, and Penny Boyes-Braem. Basic objects in natural categories. *Cognitive psychology*, 1976.

[87] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction. *TOG*, 2004.

[88] Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. The earth mover's distance as a metric for image retrieval. In *IJCV*, volume 40, 2000.

[89] Yong Rui, Thomas Huang, Michael Ortega, and Sharad Mehrotra. Relevance feedback: A power tool for interactive content-based image retrieval. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(5), 1998.

[90] Matthew Schultz and Thorsten Joachims. Learning a distance metric from relative comparisons. In *NIPS*, 2003.

[91] B. Settles. *Curious machines: active learning with structured instances*. 2008.

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

[93] M. Stark, M. Goesele, and B. Schiele. A shape-based object class model for knowledge transfer. In *ICCV*, 2010.

[94] Michael Stark, Jonathan Krause, Bojan Pepik, David Meger, James J. Little, Bernt Schiele, and Daphne Koller. Fine-grained categorization for 3d scene understanding. In *BMVC*, 2012.

[95] Raphael Sznitman, Anasuya Basu, Rogerio Richa, James Handa, Peter Gehlbach, Bruno Jedynak, Russell H. Taylor, and Gregory D. Hager. Unified detection and tracking in retinal microsurgery. In *MICCAI*, pages 1–8, 2011.

[96] Raphael Sznitman and Bruno Jedynak. Active testing for face detection and localization. *Transactions on Pattern Analysis and Machine Intelligence*, 32(10):1914–1920, 07 2010.

[97] O. Tamuz, C. Liu, S. Belongie, O. Shamir, and A. Kalai. Adaptively learning the crowd kernel. In *ICML*, 2011.

[98] S. Tsang, B. Kao, K.Y. Yip, W.S. Ho, and S.D. Lee. Decision trees for uncertain data. In *International Conference on Data Engineering (ICDE)*, 2009.

[99] Theodoros Tsiligkaridis, Brian M. Sadler, and Alfred O. Hero. A collaborative 20 questions model for target search with human-machine interaction. *ICASSP*, pages 6516–6520, 2013.

[100] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *JMLR*, 6(2):1453, 2006.

[101] Amos Tversky. Features of similarity. *Psychological rev*, 1977.

[102] Cornell University. www.allaboutbirds.org. http://www.allaboutbirds.org/NetCommunity/page.aspx?pid=1053.

[103] L.J.P van der Maaten and G.E. Hinton. Visualizing non-metric similarities in multiple maps. *ML*, 2012.

[104] L.J.P van der Maaten and K.Q Weinberger. Stochastic triplet embedding. In *MLSP*, 2012.

[105] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library. http://www.vlfeat.org/, 2008.

[106] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman. Multiple kernels for object detection. In *ICCV*, 2009.

[107] S. Vijayanarasimhan and K. Grauman. What's It Going to Cost You?: Predicting Effort vs. Informativeness for Multi-Label Image Annotations. In *CVPR*, 2009.

[108] S. Vijayanarasimhan and K. Grauman. Large-scale live active learning. In *CVPR*, 2011.

[109] L. Von Ahn, B. Maurer, C. McMillen, D. Abraham, and M. Blum. Recaptcha: Human-based character recognition via web security measures. *Science*, 321(5895), 2008.

[110] Carl Vondrick and Deva Ramanan. Video Annotation and Tracking with Active Learning. In *NIPS*, 2011.

[111] Carl Vondrick, Deva Ramanan, and Donald Patterson. Efficiently Scaling Up Video Annotation with Crowdsourced Marketplaces. In *Proceedings of European Conference on Computer Vision*, 2010.

[112] C. Wah, S. Branson, P. Perona, and S. Belongie. Multiclass recognition and localization with humans in the loop. In *ICCV*, 2011.

[113] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. Caltech-UCSD Birds-200-2011. Technical Report CNS-TR-2011-001, Caltech, 2011.

[114] Catherine Wah, Grant Van Horn, Steve Branson, Subhransu Maji, Pietro Perona, and Serge Belongie. Similarity comparisons for interactive fine-grained categorization. In *CVPR*, 2014.

[115] Mitchell Waite. whatbird.com. http://www.whatbird.com/.

[116] G. Wang and D. Forsyth. Joint learning of visual attributes, object classes and visual saliency. In *ICCV*, 2009.

[117] Josiah Wang, Katja Markert, and Mark Everingham. Learning models for object recognition from natural language descriptions. In *BMVC*, 2009.

[118] M. Weber, M. Welling, and P. Perona. Unsupervised learning of models for recognition. In *ECCV*, Dublin, Ireland, 2000.

[119] P. Welinder, S. Branson, S. Belongie, and P. Perona. The Multidimensional Wisdom of Crowds. In *NIPS*, 2010.

[120] 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.

[121] W. Wu and J. Yang. SmartLabel: an object labeling tool. In *Multimedia*, 2006.

[122] Y. Yang and D. Ramanan. Articulated Pose Estimation using Flexible Mixtures of Parts. In *CVPR*, 2011.

[123] Bangpeng Yao, Gray Bradski, and Li Fei-Fei. A codebook and annotation-free approach for fgvc. In *CVPR*, 2012.

[124] Bangpeng Yao, Aditya Khosla, and Li Fei-Fei. Combining randomization and discrimination for fgvc. In *CVPR*, 2011.

[125] N. Zhang, R. Farrell, and T. Darrell. Pose pooling kernels for sub-category recognition. In *CVPR*, 2012.

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

[127] X. Zhou and Thomas Huang. Relevance feedback in image retrieval: A comprehensive review. *Multimedia systems*, 2003.