

Abstract of

“Collective Insight: Crowd-driven Image Understanding”

by Genevieve Patterson, Ph.D., Brown University, May 2016.

Crowdsourced training data has become a mainstay in computer vision. Some of the most significant discoveries of the last few years were made possible by crowd annotated datasets. How can researchers best exploit the vast untapped human intelligence available in the crowd? Beyond naive annotation, we explore several distinct types of crowd interaction. We poll the crowd to discover a taxonomy of visual attributes, leverage intelligent annotation protocols to label a massive dataset, and use the crowd to build detectors with minimal supervision.

This dissertation comprises three sections. First, we present a pipeline for automatically generating a large set of discriminative visual attributes. We demonstrate that the crowd can generate attributes useful for scene classification and create the SUN Attribute Dataset, the largest set of attributes for scenes. The next section introduces the largest dataset of attributes for objects, the MS COCO Attributes Dataset. Using our MS COCO Attributes Dataset, a fine-tuned classification system can do more than recognize object categories – for example, rendering multi-label classifications such as “sleeping spotted curled-up cat” instead of simply “cat”. To overcome the expense of annotating thousands of MS COCO object instances with hundreds of attributes, we present an Economic Labeling Algorithm (ELA) which intelligently generates crowd labeling tasks based on correlations between attributes. The final part of this dissertation describes the Tropel system, an active learning pipeline that takes one user-provided example and bootstraps a detector using active query responses from the crowd. We use Tropel to create hundreds of detectors on-demand from unlabeled images from three domains – ornithological images, fashion images, and street-level images of Paris.

Collective Insight

Crowd-driven Image Understanding

by

Genevieve Patterson

Department of Computer Science, Brown University

B. S. Electrical Engineering, University of Arizona, 2007

B. S. Mathematics, University of Arizona, 2007

M. S. Electrical Engineering, University of Tokyo, 2009

A dissertation submitted in partial fulfillment of the
requirements for the Degree of Doctor of Philosophy
in the Department of Computer Science at Brown University

Providence, Rhode Island

May 2016

© Copyright 2016 by Genevieve Patterson

Department of Computer Science, Brown University

This dissertation by Genevieve Patterson

*Department of Computer Science, Brown University is accepted in its present form by
the Department of Computer Science as satisfying the dissertation requirement
for the degree of Doctor of Philosophy.*

Date _____

James Hays, Director

Recommended to the Graduate Council

Date _____

Erik Sudderth, Reader
Brown University

Date _____

Stefanie Tellex, Reader
Brown University

Date _____

Serge Belongie, Reader
Cornell University and Cornell Tech

Approved by the Graduate Council

Date _____

Peter M. Weber
Dean of the Graduate School

Acknowledgments



Figure 1: *Acknowledgments.*

Grad school friends who stayed true, thick or thin
Advisors backed me up for a win
Salad days were so merry
(Though possibly from sherry)
Thank you deeply to my kith and kin

Genevieve Patterson was supported by the Department of Defense (DoD) through the National Defense Science & Engineering Graduate Fellowship (NDSEG) Program. This was a wonderful program. I truly am grateful.

Contents

List of Tables	vii
List of Figures	viii
1 Introduction	1
1.1 Discovering and Labeling Attributes	1
1.2 Decreasing Cost	3
1.3 Crowd-in-the-loop Detection	4
1.4 Evaluation	6
2 Crowd Labeling	8
2.1 Attribute-based Representations of Scenes	8
2.2 Building a Taxonomy of Scene Attributes from Human Descriptions	13
2.3 Building the SUN Attribute Database	15
2.4 Exploring Scenes in Attribute Space	18
2.5 Recognizing Scene Attributes	20
2.6 Predicting Scene Categories from Attributes	29
2.7 Predicting Human Confusions	35
2.8 Conclusions from building the SUN Attribute Dataset	36
3 Scalable Dataset Creation	37
3.1 Introduction	37
3.2 Related Work	39
3.3 Attribute Discovery	41
3.4 Exhaustive Annotation	43
3.5 Economic Labeling	44
3.6 Attribute Classification	49
3.7 Worker Performance	52
3.8 Conclusions from building the COCO Attributes Dataset	56

4	Using the crowd to discover features and train detectors	57
4.1	Crowd-enabled Mid-Level Feature Discovery	57
4.2	Scene Classification with Discriminative Patches	60
4.3	Tropel: Crowdsourcing Detectors with Minimal Training	63
4.4	Related Work	65
4.5	Bootstrapping Classifiers	67
4.6	Experimental Evaluation - CUB dataset	70
4.7	Experimental Evaluation - Fashion dataset	82
4.8	Detecting difficult to name concepts	86
4.9	Omitting the Crowd: Detectors created by End-Users Only	88
4.10	Assessing Worker Performance	92
4.11	Conclusions from building the Tropel System	96
5	Conclusion	97
A	Nationalities of the Mechanical Turk Workforce	98
B	Extended Results	103
C	Choosing a Title	107

★ Parts of this thesis have previously appeared in the following conference proceedings and journal publications:

Genevieve Patterson and James Hays. Sun attribute database: Discovering, annotating, and recognizing scene attributes. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2751–2758. IEEE, 2012

Genevieve Patterson, Chen Xu, Hang Su, and James Hays. The sun attribute database: Beyond categories for deeper scene understanding. *IJCV*, 108(1-2):59–81, 2014

Genevieve Patterson, Tsung-Yi Lin, and James Hays. Using humans to build mid-level features. In *Computer Vision and Pattern Recognition (CVPR), Scene Understanding Workshop*, 2013

Genevieve Patterson, Grant Van Horn, Serge Belong, Pietro Perona, and James Hays. Crowd in the loop active learning. In *Neural Information Processing Systems (NIPS), Crowd Workshop*, 2013

Genevieve Patterson, Grant Van Horn, Serge Belongie, Pietro Perona, and James Hays. Tropel: Crowdsourcing detectors with minimal training. In *Third AAAI Conference on Human Computation and Crowdsourcing*, 2015

Jianxiong Xiao, James Hays, Bryan C Russell, Genevieve Patterson, Krista A Ehinger, Antonio Torralba, and Aude Oliva. Basic level scene understanding: categories, attributes and structures. *Frontiers in psychology*, 4, 2013

List of Tables

2.1	Most Correlated Attributes.	30
4.1	Crowd Training with Negative Examples	77
4.2	Comparison to Active Learning Baseline	77
4.3	Hierarchical Similarity of Top Detections	78
4.4	Weighted Response Strategies	81
A.1	COCO Attributes Worker Nationality	100
A.2	Tropel Worker Nationality	102
C.1	Titles Contributed on Facebook	108
C.2	Education Level of AMT Workers	108
C.3	All Titles Submitted via AMT	109

List of Figures

1	Acknowledgments.	iv
1.1	Example Scene Attributes.	2
1.2	Example Multilabel Annotations from MS COCO Attributes.	3
1.3	Output of a collaborative classifier	5
2.1	Visualization of a hypothetical space of scenes	9
2.2	Scenes partitioned by attributes rather than categories	10
2.3	Attribute Collection UIs.	13
2.4	Scene Attributes word cloud	14
2.5	Annotation interface for AMT workers	16
2.6	AMT Worker Statistics	16
2.7	Examples of Attribute Labels	18
2.8	Distributions of scenes with the given attribute	19
2.9	2D visualization of the SUN Attribute dataset	21
2.10	2D visualization of 15 scene categories.	22
2.11	Average Precision for Attribute Classifiers, Balanced Training Set	24
2.12	Average Precision for Attribute Classifiers, Natural Population Training Set	25
2.13	Examples of Scene Attribute Detection	27
2.14	Top 5 Most Confident detections in Test Set	28
2.15	Scene Category Recognition using Ground Truth Attribute Labels	31
2.16	Scene Category Recognition using Estimated Attribute Labels	32
2.17	Scene Category Recognition without Visual Examples	34
2.18	Comparison to Human Confusions	36
3.1	AMT User Interfaces used in the creation of MS COCO Attributes	42
3.2	Mean Recall Comparison of Alternative ELA methods	47
3.3	Mean Recall Across all Categories for 50 Attributes	48
3.4	MS COCO Attribute Dataset show in t-Stochastic Nearest Neighbor Embedding	50
3.5	t-SNE Visualization of 4 Object Categories in Attribute Space	51
3.6	Table of Example Annotations from MS COCO Attributes	51
3.7	Multilabel Attribute Classification	53

3.8	Example Attribute Classifications - SVM	54
3.9	Example Attribute Classifications - Multilabel CNN	55
3.10	Worker Agreement during MS COCO Attributes Annotation	55
3.11	Consensus Agreement on both Positive and Negative Labels.	56
4.1	AMT patch cluster refinement interface.	59
4.2	Popularity of images in the corresponding positions on the UI.	60
4.3	Most confident detections by Discriminative Patches	61
4.4	Scene Classification Performance of Human and Automatic Patches	61
4.5	Comparison of Scene Category Confusions made by Automatic and Crowd Patch Classifiers	62
4.6	Top detections from One Seed Classifier	63
4.7	Active Query User Interface	69
4.8	Example detections at different iterations of the Tropel pipeline	70
4.9	Tropel Detector Average Precision per Iteration	71
4.10	Tropel Detector Average Precision for 200 Bird Species	73
4.11	User Interface with Zero-Shot learning instructions	75
4.12	Difference in Detector Average Precision	76
4.13	Classifier Drift	79
4.14	Classifier Drift Comparison	79
4.15	Example detections at different iterations of the pipeline: Fashion	83
4.16	Example detections from 5 fashion concepts	84
4.17	Comparison of Worker selected training examples and detector output.	85
4.18	Top 20 detections of classifiers trained on two different datasets	86
4.19	Seed Patch Selection	87
4.20	Example detections at all iterations of the pipeline: Architecture	88
4.21	Example detections for 5 architectural concepts	89
4.22	Multi-Detector Search: Cathedral	89
4.23	Multi-Detector Search: Bridge	90
4.24	Multi-Detector Search: Fruit Stand	91
4.25	End-User versus Crowd Detector AP comparison	92
4.26	Bird Annotator Accuracy	93
4.27	Fashion Annotator Accuracy	94
4.28	Annotator Recall	95
A.1	MS COCO Attributes Worker Information	99
A.2	Tropel Worker Information	101
B.1	Attribute Population	104
B.2	Mean Recall Across all Categories for all Attributes	105
B.3	Multilabel Attribute Classification	106

Chapter 1

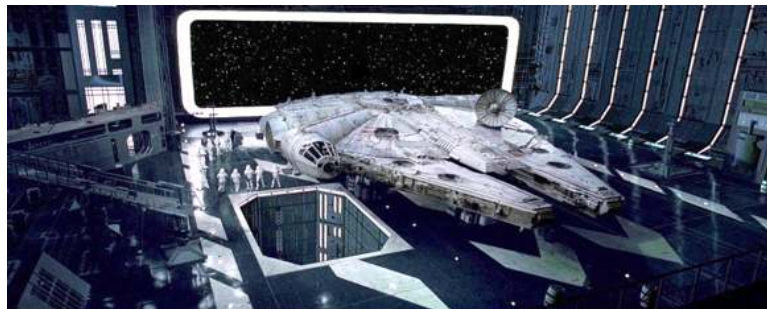
Introduction

In recent years, the computer vision community has witnessed an explosive increase in the power of recognition systems. State-of-the-art detectors require hundreds of thousands or even millions of labeled training images to achieve high performance. This dissertation explores how to create training data accurately and cheaply from large amounts of unlabeled images. The following chapters introduce several novel systems for discovering interesting visual phenomena, labeling visual events economically, and eliminating the need for dataset annotation altogether.

Our systems employ crowd workforces to enable large-scale data collection and detector creation, exploring a new type of crowd-computation. Each of the following chapters delivers a system that collaborates with the crowd to create a labeled dataset or train object detectors. We identify what is visually important by intelligently exploiting the crowd. We demonstrate that a crowd workforce can enable cost effective, on-demand creation of datasets and visual classifiers.

1.1 Discovering and Labeling Attributes

Traditionally, computer vision algorithms describe visual phenomena (e.g. objects, faces, actions, scenes, etc.) by giving each instance a categorical label (e.g. cat, Halle Berry, drinking, downtown street, etc.). Identifying only objects is a limited approximation of the human understanding of natural images. For scenes in particular, this model has several significant issues: (1) The extent of scene understanding achievable is quite shallow – there is no way to express interesting *intra*-category variations. (2) The space of scenes is continuous, so hard partitioning creates numerous ambiguous boundary cases. (3) Images often simultaneously exhibit characteristics of multiple distinct scene categories. (4) A categorical



Spatial Envelope	large, enclosed
Affordances / Functions	can fly, park, walk
Materials	shiny, black, hard
Object Presence	has storm troopers, ships
Emotion	scary, intimidating

Figure 1.1: *Example Scene Attributes*. The scene shown above would be poorly described using only a single scene category label, e.g. ‘landing bay’ [49]. Instead, this figure displays a variety of attributes that give a more detailed description of what is happening in the scene.

representation can not generalize to types of scenes which were not seen during training. To overcome these issues, we employ attribute-based scene understanding. In Chapter 2, we collect a large dataset of scene attributes in order to reveal the power of scene understanding using attributes.

In the domain of scenes, an attribute-based representation might describe an image with ‘concrete’, ‘shopping’, ‘natural lighting’, ‘glossy’, and ‘stressful’ in contrast to a categorical label such as ‘store’. Attributes do not follow scene category boundaries, allowing them to describe intra-class variation (e.g. a canyon might have water or it might not) and inter-class relationships (e.g. both a canyon and a beach could have water).

Scene representations are vital to enabling many data-driven graphics and vision applications. There is important research on *low-level* representations of scenes (i.e. visual features) such as the gist descriptor [51] or spatial pyramid [43], but there has been little investigation into *high-level* representations of scenes (e.g. attributes or categories), which are directly interpretable by end-users, leaving the standard category-based recognition paradigm largely unchallenged. In Chapter 2, we explore a new, attribute-based representation of scenes. Chapter 2 describes the process of discovering discriminative visual attributes via crowd experiments as well as annotating a large dataset with the discovered attributes.

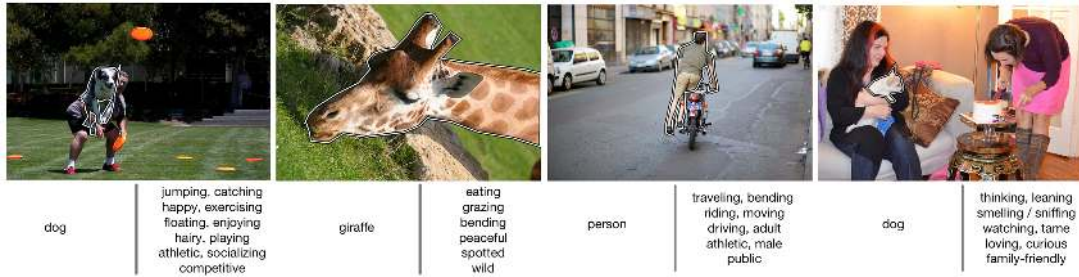


Figure 1.2: *Examples from MS COCO Attributes.* In the figure above, images from the MS COCO dataset are shown with one object outlined in white. Under the image, the MS COCO object label is listed on the left, and the MS COCO Attribute labels are listed on the right. The MS COCO Attributes labels give a deep and broad description of the context of the object.

1.2 Decreasing Cost

In Chapter 3, we expand our attribute labeling system to a large object dataset. We discover and annotate meaningful attributes for the 80 object categories of the Microsoft Common Objects in Context (MS COCO) dataset to deliver the largest attribute dataset to date. Our object attributes are identified using text-mining as well as crowd experiments. Using our MS COCO Attributes dataset, a fine-tuned classification system can do more than recognize object categories – for example, rendering multi-label classifications such as “sleeping spotted curled-up cat” instead of simply “cat”.

In order to overcome the expense of exhaustively annotating each object instances with each attribute label, we present an economic labeling algorithm (ELA) which intelligently generates crowd labeling tasks based on correlations between attributes. The ELA offers a large reduction in labeling cost while maintaining reasonable attribute recall and variety. We create a baseline for object-attribute classification on test images from the MS COCO dataset to demonstrate that object-attribute recognition is a challenging problem.

MS COCO Attributes contains 196 attribute labels covering 29 object categories. In total, we collected 3.5 million object-attribute pair annotations. Our dataset includes 180 thousand different object instances. To demonstrate the utility and cost savings of our ELA method, we conduct classification experiments which show that our efficiently labeled training data can be used to produce classifiers of similar discriminative ability as classifiers created using exhaustively labeled ground truth.

1.3 Crowd-in-the-loop Detection

Discovering discriminative mid-level features

Without the help of humans or the crowd, identifying the contextually distinctive parts of an image is a challenging task for computer vision. The first section of Chapter 4 investigates putting the crowd-in-the-loop of the CV pipeline by using human participants to discover mid-level discriminative features. Amazon Mechanical Turk (AMT) workers filter groups of cropped image patches to identify clusters that have strong visual and semantic similarity. We show that SVMs trained from human-defined discriminative patches outperform the patch classifiers discovered without user participation by Singh et al. and Doersch et al. [16, 72].

The first part of Chapter 4 shows that the crowd is capable of training classifiers for coherent localized visual events. We use these crowd-created local element classifiers as mid-level features in a scene classification pipeline. In the second part of Chapter 4, we use the crowd to create fine-grained object and part classifiers from unlabeled image datasets.

Crowd-driven Active Learning

In the second part of Chapter 4, we examine a typical problem for users of computer vision. An end user, John, has access to a large set of images. John wants to search these images for a specific visual event, and begins his search from a positive example image. John's use case is difficult problem for state of the art recognition and detection systems because it requires contextual affordances and attributes. Obtaining large training sets for a variety of these semantic attributes is challenging. There are an endless supply of possible attributes to chose from. This dissertation addresses how researchers can narrow their efforts to the attributes that are most important to humans.

Michelle, a casual bird spotter, presents another unsolved problem to state of the art systems. Michelle wants to use computer vision to identify a bird that is in an unusual pose, is a juvenile or female, or is a new species to existing classification systems. Classifying bird species is a well studied problem in the literature due to the high-quality annotations available for 200 bird species in the CUB dataset [80]. However, systems trained on this type of dataset cannot recognize new species on the fly and have well known issues with identifying juvenile, female, and atypically posed birds which may not look like the more distinctive and archetypal male birds. This dissertation also addresses incorporating end-users like Michelle directly in the process of creating novel event detectors.



Figure 1.3: Output of a collaborative classifier initialized with a single user selected example.

As demonstrated in Chapter 2, a major obstacle in human-CV interaction is the overhead of collecting a trustworthy set of annotations for a set of images. Chapter 2 shows that crowd workforces can successfully label large datasets, significantly reducing the labeling burden on the researcher. Unfortunately, this process severely restricts the possibility for end-user involvement. The objects and categories to annotate must be defined in advance. A quality control system needs to be constructed to train and monitor the crowd workers. There is no doubt that this process is the gold standard for creating a ground truth dataset necessary for benchmarking new vision technologies. The SUN Attribute dataset we create in Chapter 2 attains high label reliability by filtering and training the crowd workers [56, 61]. However, we claim that a system can make approximations in this process in order to enable dynamic, flexible end-user interaction with the vision system.

Chapter 4 introduces the Tropel system which enables non-technical users to create arbitrary visual detectors without first annotating a training set. These non-technical users are experts at identifying what they are searching for but may not have the technical skill to implement a CV system. The primary contribution of this chapter is a crowd active learning pipeline that is seeded with only a single positive example and an unlabeled set of training images. We examine the crowd’s ability to train visual detectors given severely limited training themselves.

Chapter 4 presents a series of experiments that reveal the relationship between worker training, worker consensus and the average precision of detectors trained by crowd-in-the-loop active learning. In order to verify the efficacy of our system, we train detectors for bird species that work nearly as well as those trained on the exhaustively labeled CUB 200 dataset at significantly lower cost and with little effort from the end user. To further illustrate the usefulness of our pipeline, we demonstrate qualitative results on unlabeled datasets containing fashion images and street-level photographs of Paris. By making

a small compromise on the perfection of the supervised annotations, our system produces an unlimited range of visual classifiers from wholly unlabeled images for far less cost than annotating a gold standard dataset for only predefined categories.

1.4 Evaluation

The claims we set forth in the abstract and earlier in this chapter require evaluation against traditional benchmarks and demonstration in novel contexts. Below we outline the evaluation procedures used in this dissertation.

Scene Classification

In Chapter 2, we argue that the crowd can be used to discover discriminative visual attributes from unlabeled scene images. To validate that claim, we present the first large-scale scene attribute database. First, we perform crowdsourced human studies to find a taxonomy of 102 discriminative attributes. We discover attributes related to materials, surface properties, lighting, affordances, and spatial layout.

Next, we build the SUN Attribute Database on top of the diverse SUN categorical database [83]. We employ AMT workers to annotate attributes for 14,140 images from 707 scene categories. We perform numerous experiments to study the interplay between scene attributes and scene categories. We train and evaluate attribute classifiers and then study the feasibility of attributes as an intermediate scene representation for scene classification and zero shot learning. We show that when used as features for these tasks, low dimensional scene attributes can compete with or improve on the state of the art performance.

In Chapter 4, we use human participants to discover mid-level discriminative features. To verify the discriminativeness of the crowd-created mid-level features, we also use a scene classification benchmark. We show that SVMs trained from human-defined discriminative patches outperform the patch classifiers discovered by Singh et al. and Doersch et al. [16, 72] when used as features for classification on the 15 scene dataset [44].

Attribute Classification

In both Chapters 2 and 3, we show that attribute classification is a challenging computer vision problem. In both chapters we present attribute classification results, establishing baselines for both scene and object attribute recognition. Because the SUN Attribute dataset was created using exhaustive labeling

and the MS COCO Attribute dataset was labeled using the ELA, there is the concern that the object-attribute pairs of the second dataset will be less varied and interesting than the first. To prove otherwise, in Chapter 3 we demonstrate that our ELA labeled dataset is similarly diverse and surprising as the exhaustively labeled attribute dataset. Experiments in Chapter 3 show that a multilabel CNN trained on an ELA-created dataset has similar performance to classifiers trained on exhaustively labeled data. Furthermore, experiments in Chapter 3 show that our attribute classifiers generalize well across object classes.

Cost Comparison with Canonical Methods

An added advantage of ELA method of Chapter 3 and the crowd-based active learning system of Chapter 4 is the economic efficiency of these pipelines. Chapter 3 presents extensive analysis of the trade-offs between label coverage and cost savings for the ELA. Chapter 4 has a side by side comparison of the cost of creating bird classifiers using our pipeline versus the cost for annotating the CUB 200 dataset.

In Chapter 3 we provide our heuristics for determining how and when to spend money on label annotation. The MS COCO Attributes dataset is created without making ‘visual’ approximations, e.g. using active learning for dataset creation, as we do in Chapter 4. This choice may limit our cost savings, but we believe our dataset will be useful for training novel algorithms in a way datasets collected with active learning cannot be.

Fine-Grained Object Detection

The primary contribution of Chapter 4 is a crowd active learning method that is seeded with only a single positive example and an unlabeled set of training images. In a sense, a user trains the crowd with a single example and the crowd, in an active learning setting, then provides the hundreds of positive and negative training examples necessary for an accurate detector. We envision this approach being useful for end-users who want to detect arbitrary objects, parts, attributes (or combinations thereof) for which there is no existing labeled database. In order to verify the efficacy of our approach, we use this crowd-based active learning pipeline to train detectors for 200 bird species. We show that these detectors work nearly as well as those trained on the exhaustively labeled CUB 200 dataset. We also demonstrate qualitative results on a new unlabeled fashion dataset and street-level photographs of Paris.

Chapter 2

Crowd Labeling

One of the core challenges of computer vision is understanding the content of a scene. Often, scene understanding is demonstrated in terms of object recognition, 3d layout estimation from multiple views, or scene categorization. In this chapter we instead reason about scene *attributes* – high-level properties of scenes related to affordances (‘shopping’, ‘studying’), materials (‘rock’, ‘carpet’), surface properties (‘dirty’, ‘dry’), spatial layout (‘symmetrical’, ‘enclosed’), lighting (‘direct sun’, ‘electric lighting’), and more (‘scary’, ‘cold’). We describe crowd experiments to first determine a taxonomy of 102 interesting attributes and then to annotate binary attributes for 14,140 scenes. These scenes are sampled from 707 categories of the SUN database and this lets us study the interplay between scene attributes and scene categories. We evaluate attribute recognition with several existing scene descriptors. Our experiments suggest that scene attributes are an efficient feature for capturing high-level semantics in scenes.

2.1 Attribute-based Representations of Scenes

Scene representations are vital to enabling many data-driven graphics and vision applications. There is important research on *low-level* representations of scenes (i.e. visual features) such as the gist descriptor [51] or spatial pyramids [43]. Typically, low-level features are used to classify scenes into a single scene category. For example, a scene could be described by the category label ‘village’ or ‘mountain’. Category labels can be a useful way to briefly describe the context of a scene. However, there are limitations to using a single category label to try to describe everything that is happening in a scene. In this chapter, we explore a different approach, attribute-based representation of scenes.

Scene attributes shake up the standard category-based recognition paradigm. Figure 2.1 illustrates



Figure 2.1: Visualization of a hypothetical space of scenes embedded in 2D and partitioned by categories.

the limitations of a strictly category-based description of scenes. Categorical scene representations have several potential shortcomings: (1) Important intra-class variations such as the dramatic differences between four ‘village’ scenes can not be captured, (2) hard partitions break up the continuous transitions between many scene types such as ‘forest’ and ‘savanna’, (3) an image can depict multiple, independent categories such as ‘beach’ and ‘village’, and (4) it is difficult to reason about unseen categories, whereas attribute-based representations lend themselves towards zero-shot learning [40, 41].

An attribute-based representation of scenes addresses these problems by expressing variation within a scene category. Using attributes, we can describe scenes using many attribute labels instead of simple binary category membership. We can also use attributes to describe new scene categories not seen at training time (zero-shot learning), which would be impossible with a category-based representation.

It is worth noting that the presence of a particular attribute can be ambiguous in a scene, just like category membership can be ambiguous. Scenes only have one category label, though, and with hundreds of categories (as with the SUN database) the ground truth category is often unclear. But with a large taxonomy of attributes, most tend to be unambiguous for a particular scene. In this work we largely treat attributes as binary (either present or not), but when annotators disagree (see Figure 2.7) it tends to be because the attribute is partially present (e.g. a slightly ‘dirty’ room or a partly ‘indoors’ patio). This real-valued notion of attribute presence is natural and in contrast to categorical representations where

The Space of Scenes: Attributes



Figure 2.2: Hypothetical space of scenes partitioned by attributes rather than categories. In reality, this space is much higher dimensional and there are not clean boundaries between attribute presence and absence.

membership is usually strict.

Our work is inspired by *attribute-based* representations of objects [3, 20, 22–24, 40, 66, 74], faces [39], and actions [48, 85] as an alternative or complement to category-based representations. Attribute-based representations are especially well suited for scenes because *scenes are uniquely poorly served by categorical representations*. For example, an object usually has unambiguous membership in one category. One rarely observes *issue 2* (e.g. this object is on the boundary between sheep and horse) or *issue 3* (e.g. this object is both a potted plant and a television).

In the domain of scenes, an attribute-based representation might describe an image with ‘concrete’, ‘shopping’, ‘natural lighting’, ‘glossy’, and ‘stressful’ in contrast to a categorical label such as ‘store’. Figure 2.2 visualizes the space of scenes partitioned by attributes rather than categories. Note, the attributes do not follow category boundaries. Indeed, that is one of the appeals of attributes – they can describe intra-class variation (e.g. a canyon might have water or it might not) and inter-class relationships (e.g. both a canyon and a beach could have water). As stated by Ferrari and Zisserman, “recognition of attributes can complement category-level recognition and therefore improve the degree to which machines perceive visual objects,” [24].

In order to explore the use of scene attributes, we build a dataset of scene images labeled with a

large vocabulary of scene attributes. Later sections in this chapter describe the creation and verification of the SUN attribute database in the spirit of analogous database creation efforts such as ImageNet [10], LabelMe [68], and Tiny Images [75].

A small set of scene attributes was explored in Oliva and Torralba’s seminal ‘gist’ paper [51] and follow-up work [52]. Eight ‘spatial envelope’ attributes were found by having participants manually partition a database of eight scene categories. These attributes such as openness, perspective, and depth were predicted using the gist scene representation. Greene and Oliva show that these global scene attributes are predictive of human performance on a rapid basic-level scene categorization task. They argue that global attributes of the type we examine here are important for human perception, saying, “rapid categorization of natural scenes may not be mediated primarily through objects and parts, but also through global properties of structure and affordance,” [29]. In this context ‘affordance’ is used to mean the capacity of a scene to enable an activity. For example, a restaurant affords dining and an empty field affords playing football.

Russakovsky and Fei-Fei identify the need to discover visual attributes that generalize between categories in [66]. Using a subset of the categories from ImageNet, Russakovsky and Fei-Fei show that attributes can both discriminate between unique examples of a category and allow sets of categories to be grouped by common attributes. In [66] attributes were mined from the WordNet definitions of categories. The attribute discovery method described in this chapter instead identifies attributes directly with human experiments. In the end we discover a larger set of attributes, including attributes that would be either too common or too rare to be typically included in the definition of categories.

More recently, Parikh and Grauman [54] argue for ‘relative’ rather than binary attributes. They demonstrate results on the eight category outdoor scene database, but their training data is limited – they do not have per-scene attribute labels and instead provide attribute labels at the category level (e.g. highway scenes should be more ‘natural’ than street scenes). This undermines one of the potential advantages of attribute-based representations – the ability to describe intra-class variation. In this chapter we discover, annotate, and recognize 15 times as many attributes using a database spanning 90 times as many categories where *every scene* has independent attribute labels.

Lampert et al. demonstrate how attributes can be used to classify unseen categories [40]. Lampert et al. show that attribute classifiers can be learned independent of category, then later test images can be classified as part of an unseen category with the simple knowledge of the expected attributes of the unseen category. This opens the door for classification of new categories without using visual training

examples to learn those unseen categories. In Section 2.6 we examine the performance of our scene attributes for zero-shot learning by recognizing test images from categories in our dataset without seeing visual examples for those scene categories.

Our scene attribute investigation is organized as follows. First, we derive a taxonomy of 102 scene attributes from crowd-sourced experiments (Section 2.2). Next, we use crowdsourcing to construct our attribute-labeled dataset on top of a significant subset of the SUN database [83] spanning 707 categories and 14,140 images (Section 2.3). We visualize the distribution of scenes in attribute space (Section 2.4).

We train and test classifiers for predicting attributes (Section 2.5). Furthermore, in Section 2.6 we explore the use of scene attributes for scene classification and the zero-shot learning of scene categories. We compare how scene classifiers derived using scene attributes confuse scene categories similar to how human respondents confuse categories. This chapter is based on research originally presented in a CVPR conference publication [57] and a longer, more detailed IJCV journal publication [61].

Since the original release of the SUN Attribute database there have been several interesting studies which use it. Zhou et. al demonstrate state of the art performance for scene attribute recognition and scene classification with the Places database [86]. In their paper, Zhou et. al introduce a very large scene dataset containing over 7 million scene images. This dataset enables the authors to train new Convolutional Neural Network (CNN) features that outperform earlier systems on scene-centric recognition tasks including scene attribute recognition.

The SceneAtt dataset expands the SUN Attribute dataset by adding more outdoor scene attributes [81]. The scene attributes discovered in later sections of this chapter are also used to support several different kinds of in-the-wild recognition systems. Kovashka et al. use the SUN attributes in their pipeline to improve personalized image search [35]. Zhou et al. use the SUN attributes as features for identifying the city in which an image was taken in [87]. Mason et al. and our own IJCV paper on the SUN attributes use the attributes as input to novel image captioning pipelines [50, 61].

These are some of the largest and most successful projects that build on or take inspiration from the SUN Attribute dataset. In the next sections, we will introduce readers to the scene attributes that helped to push forward research in scene and attribute understanding.

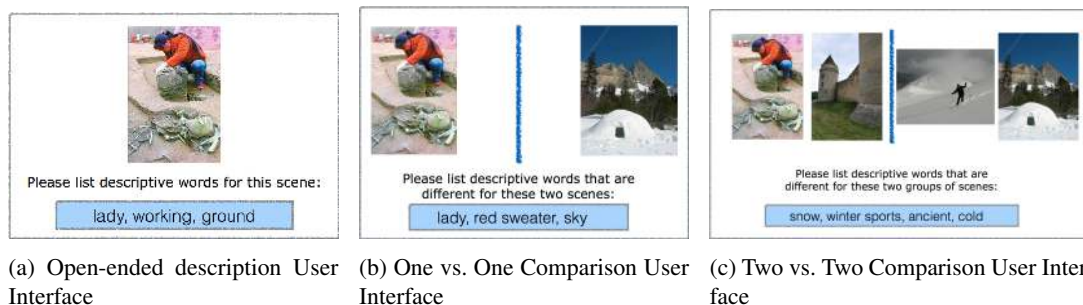


Figure 2.3: *Attribute Collection UIs*. These are examples of the Mechanical Turk user interfaces used to collect scene attributes.

2.2 Building a Taxonomy of Scene Attributes from Human Descriptions

Our first task is to establish a taxonomy of scene attributes for further study. The space of attributes is effectively infinite but the majority of possible attributes (e.g., “Was this photo taken on a Tuesday”, “Does this scene contain air?”) are not interesting. We are interested in finding discriminative attributes which are likely to distinguish scenes from each other (not necessarily along categorical boundaries). We limit ourselves to *global, binary* attributes. This limitation is primarily economic – we collect millions of labels and annotating binary attributes is more efficient than annotating real-valued or relative attributes. None-the-less, by averaging the binary labels from multiple annotators we produce a real-valued confidence for each attribute.

To determine which attributes are most relevant for describing scenes we perform open-ended image description tasks on Amazon Mechanical Turk (AMT). First we establish a set of ‘probe’ images for which we will collect descriptions. There is one probe image for every category, selected for its canonical appearance. We want a set of images which is maximally diverse and representative of the space of scenes. For this reason the probe images are the images which human participants found to be most typical of 707 SUN dataset categories [19].

We initially ask AMT workers to provide text descriptions of the individual probe images. From thousands of such tasks (hereafter HITs, for human intelligence tasks) it emerges that people tend to describe scenes with five types of attributes: (1) Materials (e.g. cement, vegetation), (2) surface properties (e.g. rusty) (3) functions or affordances (e.g. playing, cooking), (4) spatial envelope attributes (e.g. enclosed, symmetric), and (5) object presence (e.g. cars, chairs). An example of the open-ended text description UI is shown in Fig. 2.3a.

found by [51], so we manually add binary versions of those attributes so that our taxonomy is a superset of prior work. In total, we find 38 material, 11 surface property, 36 function, and 17 spatial layout attributes. Attributes which were reported in less than 1% of trials were discarded.

2.3 Building the SUN Attribute Database

With our taxonomy of attributes finalized we create the first large-scale database of attribute-labeled scenes. We build the SUN attribute database on top of the existing SUN categorical database [83] for two reasons: (1) to study the interplay between attribute-based and category-based representations and (2) to ensure a diversity of scenes. We annotate 20 scenes from each of the 717 SUN categories. Of the full SUN database, which has over 900 categories, only 717 contain at least 20 instances. Our goal is to collect ground truth annotations for all of the 102 attributes for each scene in our dataset. In total we gather more than four million labels. This necessitates a crowdsourced annotation strategy and we once again utilize AMT.

The Attribute Annotation Task. The primary difficulty of using a large, non-expert workforce is ensuring that the collected labels are accurate while keeping the annotation process fast and economical [73]. From an economic perspective, we want to have as many images labeled as possible for the lowest price. From a quality perspective, we want workers to easily and accurately label images. We find that particular UI design decisions and worker instructions significantly impacted throughput and quality of results. After several iterations, we choose a design where workers are presented with a grid of 4 dozen images and are asked to consider only a single attribute at a time. Workers are asked to click on images which exhibit the attribute in question. Before working on our HITs, potential annotators are required to pass a quiz covering the fundamentals of attribute identification and image labeling. The quiz asked users to select the correct definition of an attribute after they were shown the definition and example pictures. Users were also graded on how many images they could identify containing a given attribute. The quiz closely resembled the attribute labeling task. An example of our HIT user interface is shown in Figure 2.5.



Even after the careful construction of the annotation interface and initial worker screening, many workers’ annotations are unreasonable. We use several techniques to filter out bad workers and then cultivate a pool of *trusted* workers:

Filtering bad workers. Deciding whether or not an attribute is present in a scene image is sometimes an ambiguous task. This ambiguity combined with the financial incentive to work quickly leads to sloppy

Scene Attribute Labeling

Click on the scenes below that contain the following lighting or material:

camping *Either an actual camp site, or scene in wilderness suitable enough for humans to make a tent and/or sleep.*

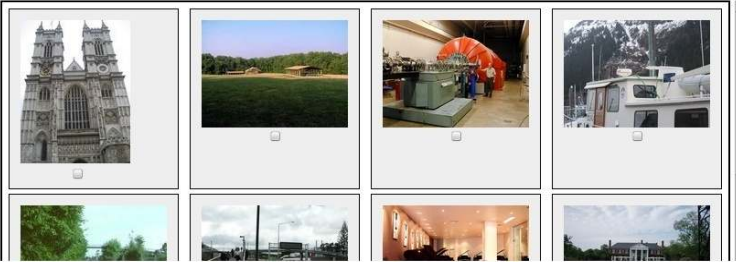



When you mouse over one of the images, a larger version of that image will appear in the box below.

These HITs are reviewed before being approved or rejected.

For further instructions Click Here!

This task can be very subjective. If you are not sure about which images should be selected, please ***SKIP THIS HIT*** or email us to ask for clarification. There are more HITs with less subjective attributes.



Images continued down the page ... ↓

Figure 2.5: Annotation interface for AMT workers. The particular attribute being labeled is prominently shown and defined. Example scenes which contain the attribute are shown. The worker can not scroll these definitions or instructions off of their screen. When workers mouse over a thumbnail a large version appears in the preview window in the top right corner.

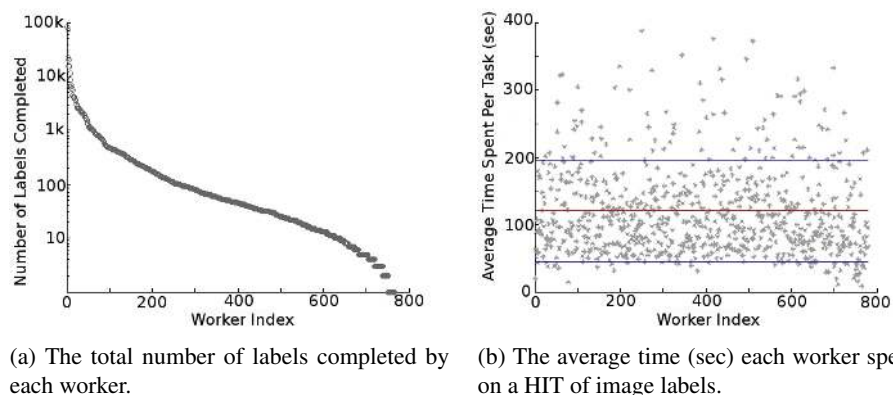


Figure 2.6: These plots visualize our criteria for identifying suspicious workers to grade. Figure 2.6a shows the heavy-tailed distribution of worker contributions to the database. The top workers spent hundreds of hours on our HITs. The red line in plot 2.6b demarcates the average work time across all workers, and the blue lines mark the positive and negative standard deviation from the mean. Work time statistics are particularly useful from identifying scam workers as they typically rush to finish HITs.

annotation from some workers. In order to filter out those workers who performed poorly, we flag HITs which are outliers with respect to annotation time or labeling frequency.

Some attributes, such as ‘ice’ or ‘fire’, rarely appear and are visually obvious and thus those HITs can be completed quickly. Other attributes, such as ‘man-made’ or ‘natural light’, occur in more than half of all scenes thus the expected completion time per HIT is higher. We use the behavioral trends shown in if Fig. 2.6 to help filter out poorly performing workers. We only use workers who give higher quality labels. This choice is supported by research such as [42] where good workers were shown to be faster *and* more accurate than the average of many workers.

Cultivating good workers. The pay per HIT is initially \$0.03 but increases to \$0.05 plus 10% bonus after workers have a proven track record of accuracy. The net result of our filtering and bonus scheme is that we cultivate a pool of trained, efficient, and accurate annotators as emphasized by [6]. In general, worker accuracy rose over time. We cull over one million poorly done early annotations from the final dataset. Worker accuracy improved over time as the workers who did not follow instructions were culled from the pool of workers who were offered the opportunity to complete HITs.

After labeling the entire dataset once with the general AMT population, we identify a smaller group of 38 trusted workers out of the ~ 800 who participated. We repeat the labeling process two more times using only these trusted workers. We repeat the labeling process in order to obtain consensus as the presence of some of the scene attributes may be a subjective decision. No worker is allowed to label the same image for the same attribute more than once. The idea of finding and heavily utilizing *good* workers is in contrast to the “wisdom of the crowds” crowdsourcing strategy where consensus outweighs expertise. our choice to utilize only workers who give higher quality labels is supported by recent research such as [42] where good workers were shown to be faster *and* more accurate than the average of many workers. Figure 2.6 shows the contributions of all workers to our database.

Figure 2.7 qualitatively shows the result of our annotation process. To quantitatively assess accuracy we manually grade ~ 600 random positive and ~ 600 random negative AMT annotations in the database. The population of labels in the dataset is not even (8% positive, 92% negative). This does not seem to be an artifact of our interface (which defaults to negative), but rather it seems that scene attributes follow a heavy-tailed distribution with a few being very common (e.g. ‘natural’) and most being rare (e.g. ‘wire’).

We graded equal numbers of positive and negative labels to understand if there was a disparity in accuracy between them. For both types of annotation, we find $\sim 93\%$ of labels to be reasonable, which means that we as experts would agree with the annotation.

Attribute	Images given 0 votes	Images given 1 vote	Images given 2 votes	Images given 3 votes
Camping				
Diving				
Medical Activity				
Cluttered Space				
Fire				

Figure 2.7: The images in the table above are grouped by the number of positive labels (votes) they received from AMT workers. From left to right the visual presence of each attribute increases. AMT workers are instructed to positively label an image if the functional attribute is *likely to occur* in that image, not just if it is actually occurring. For material, surface property, or spatial envelope attributes, workers were instructed to positively label images only if the attribute is present.

In the following sections, our experiments rely on the consensus of multiple annotators rather than individual annotations. This increases the accuracy of our labels. For each of our 102 attributes, we manually grade 5 scenes where the consensus was positive (2 or 3 votes) and likewise for negative (0 votes). In total we grade 1020 images. We find that if 2 out of 3 annotations agree on a positive label, that label is reasonable $\sim 95\%$ of the time. Many attributes are very rare, and there would be a significant loss in the population of the rare attributes if consensus was defined as 3/3 positive labels. Allowing for 2/3 positive labels to be the consensus standard increases the population of rare attributes without degrading the quality of the labels.

2.4 Exploring Scenes in Attribute Space

Now that we have a database of attribute-labeled scenes we can attempt to visualize that space of attributes. In Fig. 2.9 we show all 14,340 of our scenes projected onto two dimensions by t-Distributed Stochastic Neighbor Embedding (t-SNE) [76]. Each subplot in Fig. 2.9 highlights the population of all images with a given attribute.

To better understand where images with different attributes live in attribute space, Fig. 2.8 illustrates

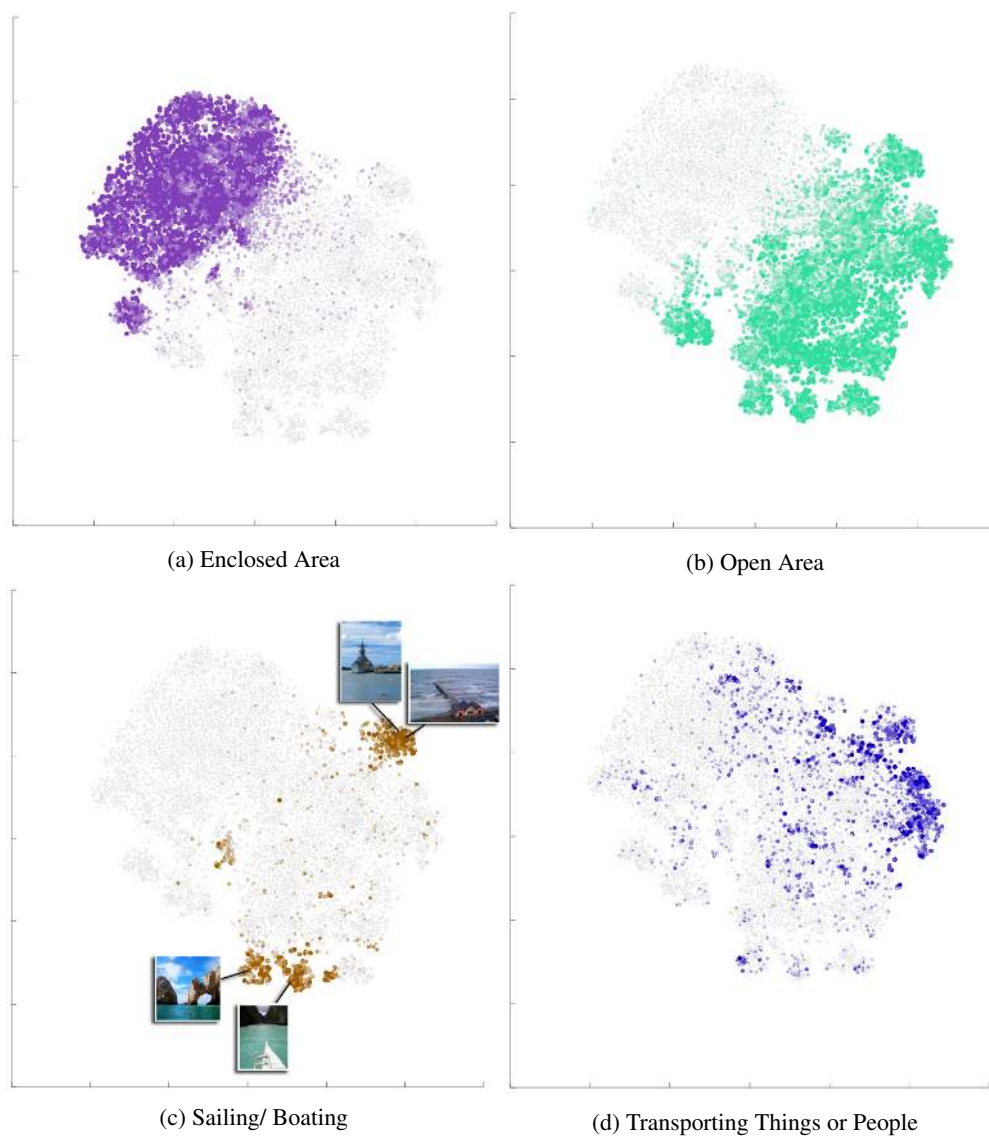


Figure 2.8: *Distributions of scenes with the given attribute.* This set of plots highlights the populations of images with the listed attributes. Each point is represented by their 102-dimensional attribute vector, reduced to a 2D projection using t-SNE. Grey points are images that do not contain the given attribute. The boldness of the colored points is proportional to the amount of votes given for that attribute in an image, e.g. darkest colored points have 3 votes. ‘Enclosed area’ and ‘open area’ seem to have a strong effect on the layout of scenes in “attribute space”. As one might hope, they generally occupy mutual exclusive areas. It is interesting to note that ‘sailing/ boating’ occurs in two distinct regions which correspond to open water scenes and harbor scenes.

where dataset images that contain different attributes live in this 2D version of the attribute feature space.

Figure 2.10 shows the distribution of images from 15 scene categories in attribute space. The particular scene categories were chosen to be close to those categories in the 15 scene benchmark [43]. In this low dimensional visualization, many of the categories have considerable overlap (e.g. bedroom with living room, street with highway, city with skyscraper). This is reasonable because these overlapping categories share affordances, materials, and layouts. With the full 102 dimensional attribute representation, these scenes could still be differentiated and we examine this task in Section 2.6.

2.5 Recognizing Scene Attributes

A motivation for creating the SUN Attribute dataset is to enable deeper understanding of scenes. For scene attributes to be useful they need to be machine recognizable. To assess the difficulty of scene attribute recognition we perform experiments using the baseline low-level features used for category recognition in the original paper introducing the SUN database [83]. Our classifiers use a combination of kernels generated from gist, HOG 2x2, self-similarity, and geometric context color histogram features. (See [83] for feature details). These four features were chosen because they are each individually powerful and because they can describe distinct visual phenomena.

How hard is it to recognize Attributes? To recognize attributes in images, we create an individual classifier for each attribute using random splits of the SUN Attribute dataset for training and testing data. Note that our training and test splits are scene category agnostic – for the purpose of this section we simply have a pool of 14,340 images with varying attributes. We treat an attribute as present if it receives at least two votes, i.e. consensus is established, and absent if it receives zero votes. As shown in Figure 2.7, images with a single vote tend to be in a transition state between the attribute being present or absent so they are excluded from these experiments.

We train and evaluate independent classifiers for each attribute. Correlation between attributes could make ‘multi-label’ classification methods advantageous, but we choose to predict attributes independently for the sake of simplicity.

To train a classifier for a given attribute, we construct a combined kernel from a linear combination of gist, HOG 2x2, self-similarity, and geometric context color histogram feature kernels. Each classifier is trained on 300 images and tested on 50 images and AP is computed over five random splits. Each classifier’s train and test sets are half positive and half negative even though most attributes are sparse (i.e. usually absent). We fix the positive to negative ratio so that we can compare the intrinsic difficulty

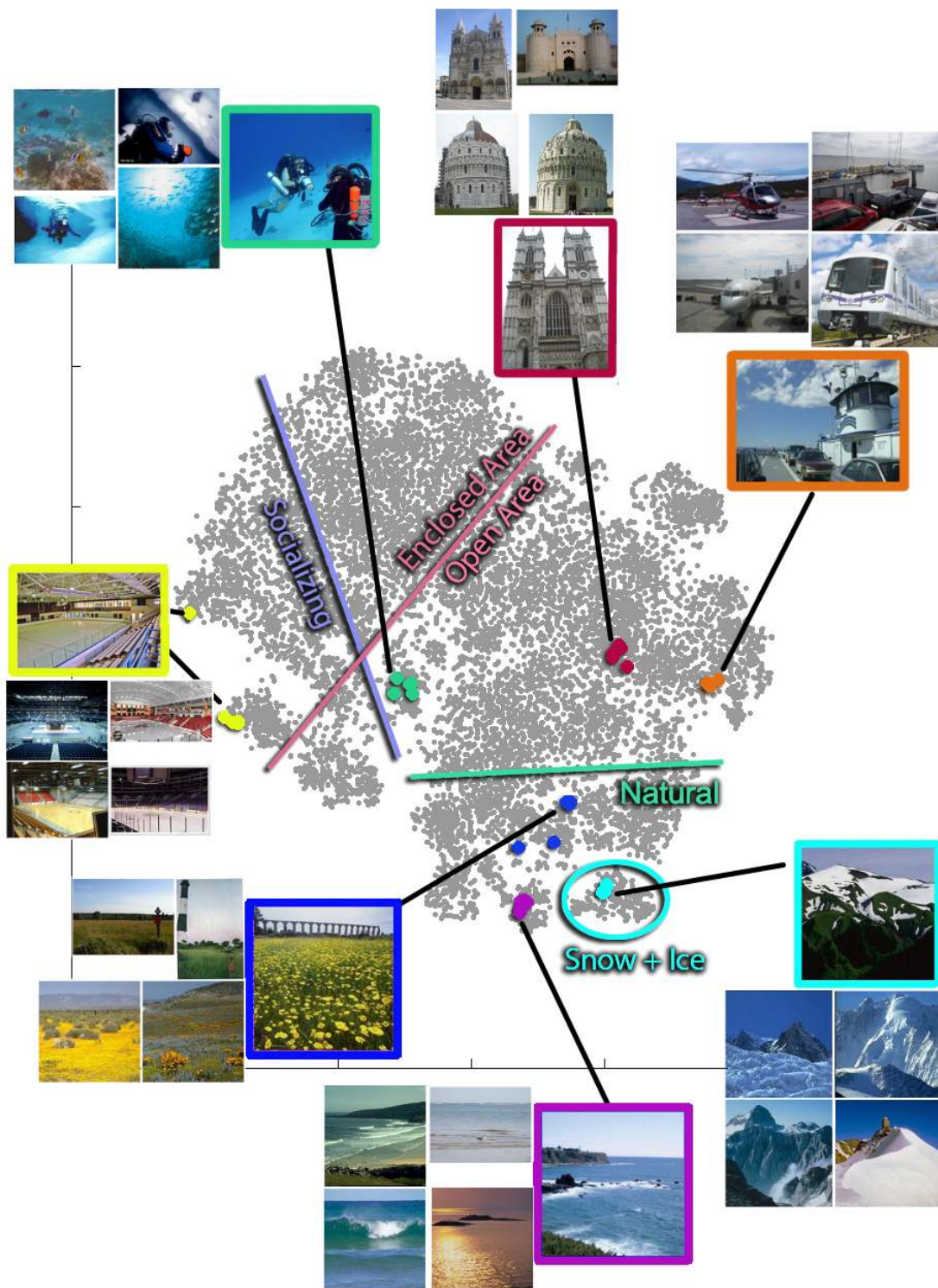


Figure 2.9: 2D visualization of the SUN Attribute dataset. Each image in the dataset is represented by the projection of its 102-dimensional attribute feature vector onto two dimensions using t-Distributed Stochastic Neighbor Embedding [76]. There are groups of nearest neighbors, each designated by a color. Interestingly, while the nearest-neighbor scenes in attribute space are semantically very similar, for most of these examples (underwater_ocean, abbey, coast, ice skating rink, field_wild, bistro, office) *none* of the nearest neighbors actually fall in the same SUN database category. The colored border lines delineate the approximate separation of images with and without the attribute associated with the border.

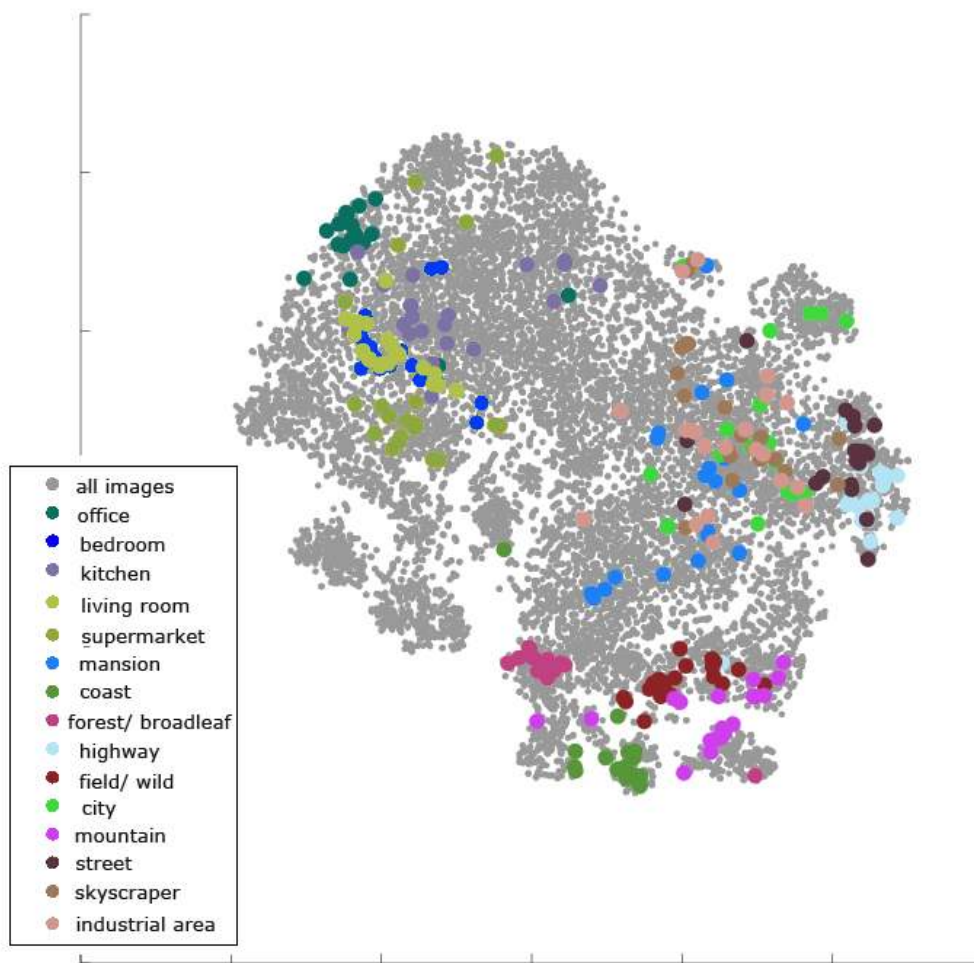


Figure 2.10: 2D visualization of 15 scene categories in Attribute Space. 20 images from each of the listed scene categories are displayed in a 2D t-SNE visualization of attribute feature space. It is interesting to see how some categories, such as 'office', 'coast', and 'forest/broadleaf' are tightly clustered, while others, such as 'bedroom', 'living room', and 'kitchen' have greater overlap when represented by scene attributes.

of recognizing each attribute without being influenced by attribute popularity. Figures 2.11 and 2.12 plot the average precision of classifiers for each attribute, given different positive/negative training example ratios. For the balanced 50% positive/ 50% negative training set in Fig. 2.11, the average precision across all attributes is 0.879. The current state of the art method, a CNN trained on the Places database, obtains an average precision of 0.915 over all attributes [86].

Some attributes are vastly more popular than others in the real world. To evaluate attribute recognition under more realistic conditions, and to make use of as much training data as the SUN attribute database affords us, we train classifiers on 90% of the dataset and test on the remaining 10%. This means that some attributes (e.g. ‘natural’ will have thousands of positive examples, and others e.g. ‘smoke’ will have barely 100). Likewise, chance is different for each attribute because the test sets are similarly skewed. The train and test instances for each attribute vary slightly because some images have confident labels for certain attributes and ambiguous labels for others and again we only use scenes with confident ground truth labels for each particular attribute classifier. Figure 2.12 shows the AP scores for these large scale classifiers. More popular attributes are easier to recognize, as expected. Overall, the average AP scores for different types of attributes are similar - Functions/ Affordances (AP 0.44), Materials (AP 0.51), Surface Properties (AP 0.50), and Spatial Envelope (AP 0.62). Average precision is lower than the previous experiment not because the classifiers are worse (in fact, they’re better) but because chance is much lower with a test set containing the natural distribution of attributes.

The classifiers used for Fig. 2.12 and the code used to generate them are publicly available.² The attribute classifiers trained on 90% of the SUN Attribute dataset are employed in all further experiments in this chapter.

Attribute Classifiers in the Wild. We show qualitative results of our attribute classifiers in Fig. 2.13a. Our attribute classifiers perform well at recognizing attributes in a variety of contexts. Most of the attributes with strong confidence are indeed present in the images. Likewise, the lowest confidence attributes are clearly not present. It is particularly interesting that function/ affordance attributes and surface property attributes are often recognized with stronger confidence than other types of attributes even though functions and surface properties are complex concepts that may not be easy to define visually. For example the golf course test image in Figure 2.13a shows that our classifiers can successfully identify such abstract concepts as ‘sports’ and ‘competing’ for a golf course, which is visually quite similar to places where no sports would occur. Abstract concepts such as ‘praying’ and ‘aged/worn’ are also

²SUN Attribute Classifiers along with the full SUN Attribute dataset and associated code are available at www.cs.brown.edu/~gen/sunattributes.html.

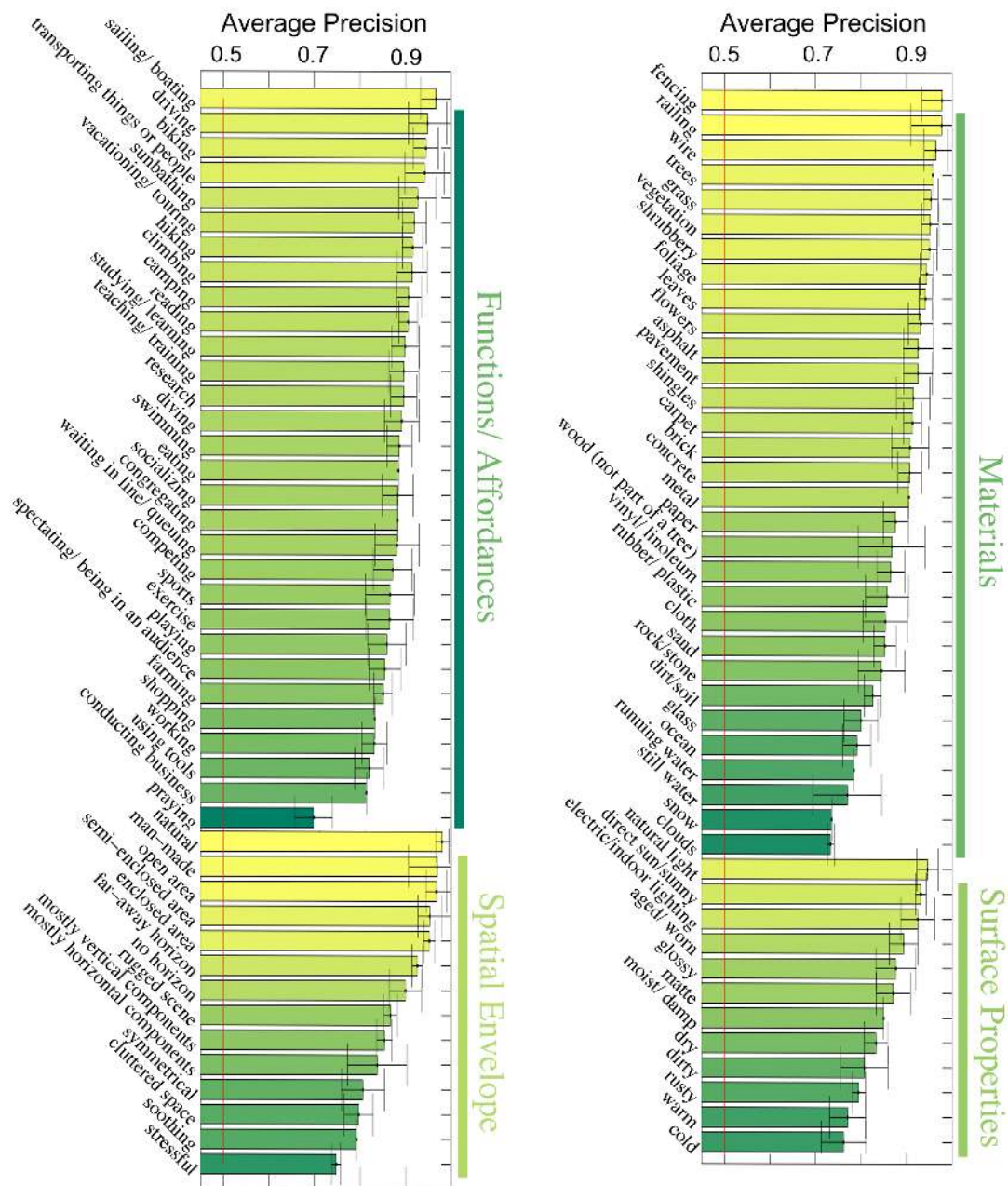


Figure 2.11: Average Precision for Attribute Classifiers, Balanced Training Set. 300 training/ 50 test examples; training and testing sets have a balance positive to negative example ratio. The AP of chance selection is marked by the red line. AP scores are often high even when the visual manifestation of such attributes are subtle. This plot show that it is possible to recognize global scene attributes. Attributes that occur fewer than 350 times in the dataset were not included in this plot.

recognized correctly in both the abbey and mosque scenes in Fig. 2.13a. Figure 2.13b shows several cases where the most confidently detected attributes are incorrect.









In earlier attribute work where the attributes were discovered on smaller datasets, attributes had the problem of being strongly correlated with each other [23]. This is less of an issue with the SUN Attribute dataset because the dataset is larger and attributes are observed in many different contexts. For instance, attributes such as “golf” and “grass” are correlated with each other, as they should be. But the correlation is not so high that a “golf” classifier can simply learn the “grass” visual concept, because the dataset contains thousands of training examples where “grass” is present but “golf” is not possible. However, some of our attributes, specifically those related to vegetation, do seem overly correlated with each other because the concepts are not semantically distinct enough.

Figure 2.13a shows many true positive detections. Somewhat surprisingly, many affordance attributes are often estimated correctly and strongly positively for images that contain them. This may be because different activities occur in very distinct looking places. For example, scenes for eating or socializing are distinct from scene for playing sports which are distinct from natural scenes where no human activity is likely to take place. Unsurprisingly, attributes related to vegetation and the shape of the scene are also relatively easy to detect. These attributes are common in the dataset, and their classifiers benefit from the additional training data.

Fig. 2.13b shows false positive detections. False negative detections can be inferred from Fig. 2.13b. This figure can help us qualitatively understand why attribute recognition may fail. In the first image in Fig. 2.13b, there is a broken-down blue car in a field grown wild. This somewhat unusual juxtaposition of a car that looks different from other, working cars in the dataset and a natural-looking scene that wouldn't normally contain cars results in the mis-estimation of the attribute ‘swimming’.

The next two images in Fig. 2.13b both fail for reasons of image scale. The images are zoomed in much closer than other images in the SUN dataset which typically try to capture a whole scene. In the case of the cat, it results in the cat being mis-identified as a snowy mountain type landscape and the carpet attribute is not recognized at all.

Figure 2.14 shows the most confident classifications in our test set for various attributes. Many of the false positives, highlighted in red, are reasonable from a visual similarity point of view. ‘Cold’, ‘moist/ damp’, and ‘eating’ all have false positives that could be reasonably considered to be confusing. ‘Stressful’ and ‘vacationing’ have false positives that could be subjectively judged to be correct - a crowded subway car could be stressful, and the New Mexico desert could be a lovely vacation spot.

Test Scene Images	Detected Attributes	Test Images	Detected Attributes
	<i>Most Confident Attributes:</i> vegetation, open area, sunny, sports, natural light, no horizon, foliage, competing, railing, natural <i>Least Confident Attributes:</i> studying, gaming, fire, carpet, tiles, smoke, medical, cleaning, sterile, marble		<i>Most Confident Attributes:</i> swimming, asphalt, open area, sports, sunbathing, natural light, diving, still water, exercise, soothing <i>Least Confident Attributes:</i> tiles, smoke, ice, sterile, praying, marble, railroad, cleaning, medical activity, gaming
	<i>Most Confident Attributes:</i> shrubbery, flowers, camping, rugged scene, hiking, dirt/soil, leaves, natural light, vegetation, rock/stone <i>Least Confident Attributes:</i> shingles, ice, railroad, cleaning, marble, sterile, smoke, gaming, tiles, medical		<i>Most Confident Attributes:</i> cold, concrete, snow, sand, stressful, aged/ worn, dry, climbing, rugged scene, rock/stone <i>Least Confident Attributes:</i> medical activity, spectating, marble, cleaning, waves/ surf, railroad, gaming, building, shopping, tiles
	<i>Most Confident Attributes:</i> eating, socializing, waiting in line, cloth, shopping, reading, stressful, congregating, man-made, plastic <i>Least Confident Attributes:</i> gaming, running water, tiles, railroad, waves/ surf, building, fire, bathing, ice, smoke		<i>Most Confident Attributes:</i> carpet, enclosed area no horizon, electric/indoor lighting, concrete, glossy, cloth, working, dry, rubber/ plastic <i>Least Confident Attributes:</i> trees, ocean, digging, open area, scary, smoke, ice, railroad, constructing/ building, waves/ surf
	<i>Most Confident Attributes:</i> vertical components, vacationing, natural light, shingles, man-made, praying, symmetrical, semi-enclosed area, aged/ worn, brick <i>Least Confident Attributes:</i> railroad, ice, scary, medical, shopping, tiles, cleaning, sterile, digging, gaming		
	<i>Most Confident Attributes:</i> vertical components, brick, natural light, praying, vacationing, man-made, pavement, sunny, open area, rusty <i>Least Confident Attributes:</i> ice, smoke, bathing, marble, vinyl, cleaning, fire, tires, gaming, sterile		

(a) *Successful Cases.* For each query, the most confidently recognized attributes (green) are indeed present in the test images, and the least confidently recognized attributes (red) are either the visual opposite of what is in the image or they are irrelevant to the image.

(b) *Failure Cases.* In the top image, it seems the smooth, blue regions of the car appear to have created false positive detections of 'swimming', 'diving', and 'still water'. The bottom images, unlike all of our training data, is a close-up object view rather than a scene with spatial extent. The attribute classifiers seem to interpret the cat as a mountain landscape and the potato chips bag as several different materials - 'carpet', 'concrete', 'glossy', and 'cloth'.

Figure 2.13: *Examples of Scene Attribute Detection.*

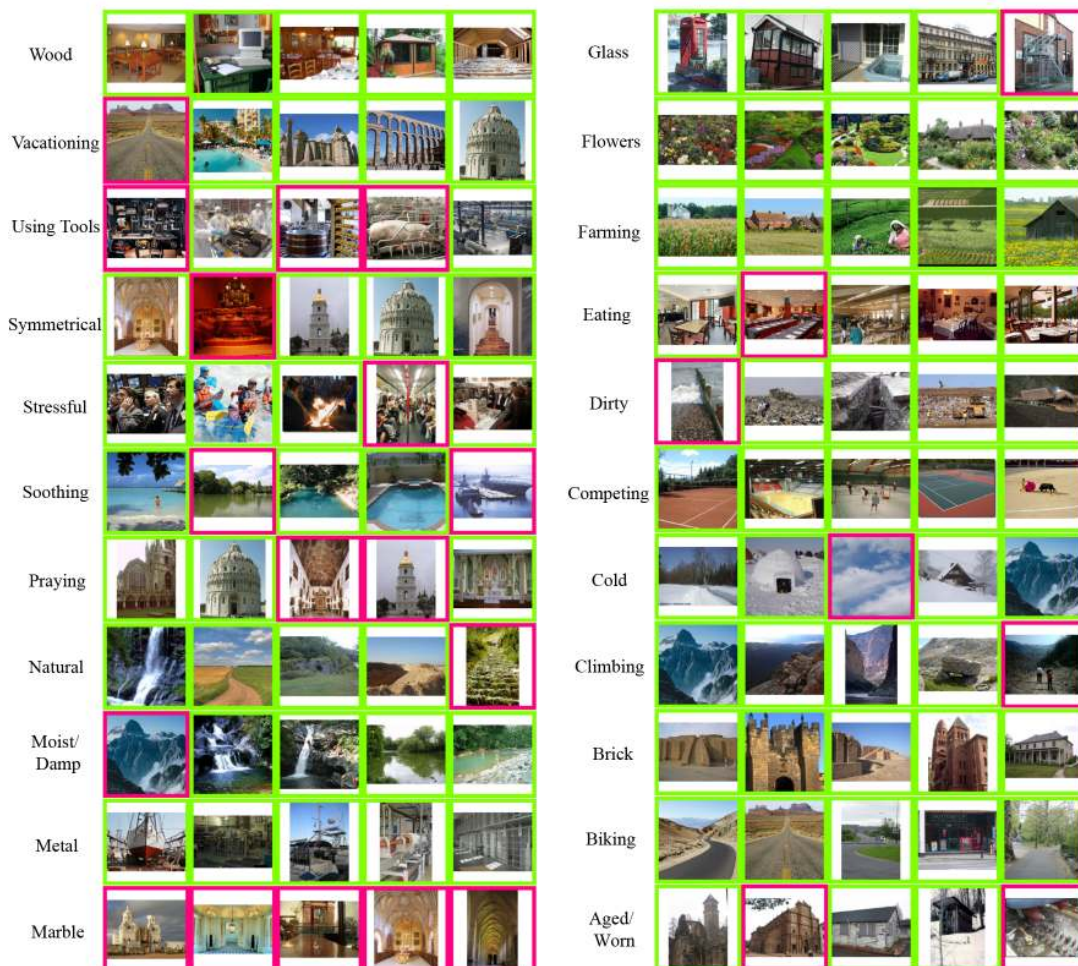


Figure 2.14: *Top 5 Most Confident detections in Test Set*. For each attribute, the top five detections from the test set are shown. Images boxed in green are true positives, and red are false positives. Examples of false positives, such as the ‘praying’ examples, show how attributes are identified in images that arguably contain the attribute, but human annotators disagreed about the attribute’s presence; in this case the false positives were a sacristy, which is a room for the storage of religious items, and a cathedral pictured at a distance. The false positive for ‘glass’ also contain glass, although photographed under glancing illumination, which may have caused the human annotators to mislabel it. For several of the examples, all of the top 5 detections are true positives. The detections for ‘brick’, ‘metal’, and ‘competing’ demonstrate the ability of attribute classifiers to recognize the presence of attributes in scenes that are quite visually dissimilar. For ‘brick’ and ‘metal’ even the kinds of bricks and metals shown are different in type, age, and use case. The false positives in the praying example are an art gallery and a monument.

Correlation of Attributes and Scene Categories. To better understand the relationships between categories and attributes, Table 2.1 lists a number of examples from the SUN 397 categories with the attribute that is most strongly correlated with each category.

The correlation between the scene category and the attribute feature of an input image is calculated using Pearson’s correlation. We calculate correlation between the predicted attribute feature vectors for 50 examples from each of the SUN 397 categories and a feature vectors that indicate the category membership of the example images.

Table 2.1 has many interesting examples where an attribute is strongly correlated with visually dissimilar but semantically related categories, such as ‘praying’ for both the indoor and outdoor church categories. Even attributes that are quite abstract concepts, such as ‘socializing’ and ‘stressful’, are the most strongly correlated attributes for ‘pub/indoor’ and ‘cockpit’, respectfully. Scene attributes capture information that is intrinsic to the nature of scenes and how humans interact with them.

2.6 Predicting Scene Categories from Attributes

Predictive Power of Attributes

In this section we measure how well we can predict scene category from *ground truth* scene attributes. While the goal of scene attributes is not necessarily to improve the task of scene categorization, this analysis does give some insight into the interplay between scene categories and scene attributes. In the next experiment, we used the attribute labels made by the crowd workers as the input feature for scene classification. This experiment gives us an upper bound for how useful scene attributes on their own could be for the task of scene classification. In the next sub-section, we will *estimate* scene attributes for previously unseen test images and use the *estimated* scene attributes as features for scene classification.

One hundred binary attributes could potentially distinguish the hundreds SUN dataset scene categories if the attributes were (1) independent and (2) consistent within each category, but neither of these are true. Many of the attributes are correlated (e.g. “farming” and “open area”) and there is significant attribute variation within categories. Furthermore, many groups of SUN database scenes would require very specific attributes to distinguish them (e.g. “forest_needleleaf” and “forest_broadleaf”), so it would likely take several hundred attributes to perfectly predict scene categories.

Figure 2.15 shows how well we can predict the category of a scene with *known* attributes as we increase the number of training examples per category. Each image is represented by the ground truth

Table 2.1: *Most Correlated Attributes*. A sampling of scene categories from the SUN 397 dataset listed with their most correlated attribute.

Category	Most Corr. Attribute	Pearson's Corr. Coeff.	Category	Most Corr. Attribute	Pearson's Corr. Coeff.
airport terminal	socializing	0.051	cockpit	stressful	0.048
art studio	cluttered space	0.039	construction site	constructing/ building	0.041
assembly line	working	0.055	corn field	farming	0.111
athletic field/outdoor	playing	0.116	cottage garden	flowers	0.106
auditorium	spectating	0.096	dentists office	medical activity	0.070
ball pit	rubber/ plastic	0.149	dining room	eating	0.064
baseball field	sports	0.088	electrical substation	wire	0.054
basilica	praying	0.101	factory/indoor	working	0.047
basketball court/outdoor	exercise	0.074	fastfood restaurant	waiting in line	0.057
bathroom	cleaning	0.092	fire escape	railing	0.051
bayou	still water	0.092	forest path	hiking	0.111
bedroom	carpet	0.054	forest road	foliage	0.095
biology laboratory	research	0.053	fountain	running water	0.041
bistro/indoor	eating	0.055	ice skating rink/indoor	sports	0.058
bookstore	shopping	0.079	ice skating rink/outdoor	cold	0.065
bowling alley	competing	0.055	iceberg	ocean	0.148
boxing ring	spectating	0.049	lecture room	studying/ learning	0.080
campsite	camping	0.053	mosque/indoor	cloth	0.060
canal/natural	still water	0.080	mosque/outdoor	praying	0.066
canal/urban	sailing/ boating	0.038	operating room	sterile	0.058
canyon	rugged scene	0.110	palace	vacationing	0.045
car interior/backseat	matte	0.079	poolroom/establishment	gaming	0.068
car interior/frontseat	matte	0.098	poolroom/home	gaming	0.075
casino/indoor	gaming	0.070	power plant/outdoor	smoke	0.074
catacomb	digging	0.081	pub/indoor	socializing	0.065
chemistry lab	research	0.067	restaurant	eating	0.088
chicken coop/indoor	dirty	0.039	restaurant kitchen	working	0.058
chicken coop/outdoor	fencing	0.045	stadium/ football	spectating	0.132
cathedral/indoor	praying	0.148	subway station/platform	railroad	0.052
church/outdoor	praying	0.088	underwater/coral reef	diving	0.165
classroom	studying/ learning	0.070	volcano	fire	0.122
clothing store	cloth	0.063	wheat field	farming	0.133

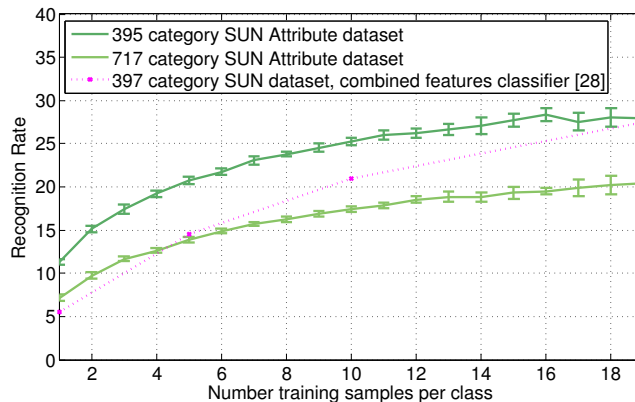


Figure 2.15: *Category recognition from ground truth attributes using an SVM.* We plot accuracy for the 717 category SUN Attribute dataset and for a subset of 395 categories which roughly match the evaluation of the SUN 397 dataset [83] (two categories present in [83] are not part of the SUN Attribute dataset). We compare attribute-based recognition to visual recognition by plotting the highest accuracy from [83] (pink dotted line).

attribute labels collected in Sec. 2.3. We compare this to the classification accuracy using low-level features [83] on the same data set. With 1 training example per category, attributes are roughly twice as accurate as low-level features. Performance equalizes as the number of training examples approaches 20 per category.

From the results in Fig. 2.15, it is clear that attributes alone are not perfectly suited for scene classification. However, the performance of our attribute-based classifiers hints at the viability of zero-shot learning techniques which have access to attribute distributions for categories but no visual examples. The fact that category prediction accuracy increases significantly with more training examples may be a reflection of intra-class attribute variations.

Attributes allow for the exploration of scenes using information that is complementary to the category labels of those scenes. To the best of our knowledge these experiments are the first to explore the use of attributes as features for scene classification. As with objects [40], attributes also offer the opportunity to learn new scene categories without using any training examples for the new categories. This “zero-shot” learning for scenes will be explored in the next section.

Scene Classification

Attributes as Features for Scene Classification.

Although our attributes were discovered in order to understand natural scenes more deeply than categorical representations, scene classification remains a challenging and interesting task. As a scene

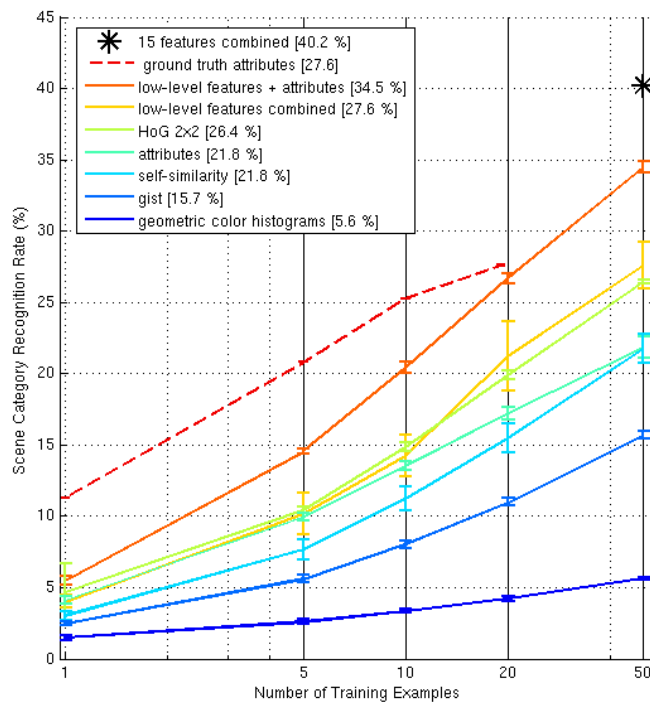


Figure 2.16: *Scene Category Recognition Rate vs. Number of Training Examples*. Classification tested on the SUN 397 dataset [83]. Images that occur in both the SUN 397 and SUN Attribute datasets were omitted from the training and test sets of the above classifiers.

classification baseline, we train one-vs-all non-linear SVMs with the same low level features used to predict attributes. Figure 2.16 compares this with various classifiers which instead operate on attributes as an intermediate representation.

The simplest way to use scene attributes as an intermediate representation is to run our attribute classifiers on the scene classification training instances and train one-vs-all SVMs in the resulting 102 dimensional space. This “predicted attribute feature” performs better than three of the low-level features, but worse than the HoG 2x2 feature.³

In Fig. 2.16 each trend line plots the scene classification accuracy of the associated feature. All predicted features use the same test/ train sets, and results averaged over several random test/ train splits.

³The images in the SUN Attribute dataset were originally taken from the whole SUN dataset, which includes more than 900 scene categories. Thus, some portion of the SUN Attribute images also appear in the SUN 397 dataset, which is also a subset of the full SUN dataset. The scene classifiers using low-level and predicted attribute features were trained and tested on the SUN397 dataset minus any overlapping images from the SUN Attribute dataset to avoid testing scene classification on the same images used to train attribute classifiers.

When combined with the 4 low-level features originally used in the attribute classifiers, the ‘attributes’ feature clearly improves performance over a scene classifier that only uses low-level features. This further supports our claim that attributes are encoding important semantic knowledge. Classification accuracy using 15 different low-level features (the same features used in Xiao et al.) plus attribute features at 50 training examples is 40.22%, slightly beating the 38.0% accuracy reported in [83].

The current state of the art performance on the SUN 397 benchmark is 56.2% in the paper introducing the Places dataset [86]. In [86], Zhou et al. use 150 training examples per category. Figure 2.15 shows that perfectly estimated attributes by themselves could achieve nearly 30% accuracy with only a tenth the number of training examples per category. Scene attributes do a great job of category prediction where there are few training examples available, and CNN-trained features do well with lots of training examples. Scene attributes capture important generalizable visual concepts.

The ground truth feature classifier in Fig. 2.16 deserves slightly more explanation. The ground truth attribute feature in Fig. 2.16 is taken from 10 random splits of the SUN Attribute dataset. Thus the number of test examples available for the ground truth feature are $(20 - n_{train})$, where n_{train} is the number of training set images whose attribute labels were averaged to come up with the attribute feature for a given category. As the number of training examples increases, the ground truth feature trend line is less representative of actual performance as the test set is increasingly small. Using ground truth attributes as a feature gives an upper bound on what attribute features could possibly contribute to scene classification.

It is important to note that the low-level features live in spaces that may have thousands of dimensions, while the attribute feature is only 102-dimensional. Partly for this reason, the attribute-based scene classifier seems to benefit less from additional training data than the low level features. This makes sense, because lower dimensional features have limited expressive capacity and because the attribute distribution for a given category isn’t expected to be especially complex (this is, in fact, a motivation for zero-shot learning or easy knowledge transfer between observed and unobserved categories).

Learning to Recognize Scenes without Visual Examples.

In zero-shot learning, a classifier is presented (by some oracle) a ground truth distribution of attributes for a given category rather than any visual examples. Test images are classified as the category whose oracle-annotated feature vector is the nearest neighbor in feature space to the test images’ features.

Canonical definitions of zero-shot learning use an intermediate feature space to generalize important

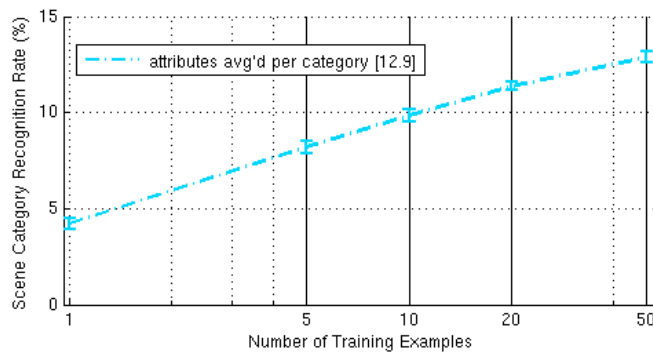


Figure 2.17: *Scene Category Recognition without Visual Examples*. The ‘attributes averaged per category’ feature is calculated by averaging the predicted attribute features of all of the training instances of a given scene category in the SUN 397 dataset. Test instances are evaluated by selecting the nearest neighbor scene category feature, and taking that scene category’s label.

concepts shared by categories [40, 53]. Lampert et al. use an attribute representation to enable knowledge transfer between seen and unseen categories, and Palatucci et al. uses phonemes. In these zero-shot learning scenarios, it is prohibitively difficult or expensive to collect low-level feature examples of an exhaustive set of categories. The use of oracle features for those unseen categories is a way to identify them without collecting enough examples to train a classifier.

The goal of zero-shot learning is to learn a classifier $f : \mathcal{X} \rightarrow \mathcal{Z}$ for a label set \mathcal{Z} , where some categories in \mathcal{Z} were not seen during training. This is accomplished by learning two transfer functions, $g : \mathcal{X} \rightarrow \mathcal{A}$ and $h : \mathcal{A} \rightarrow \mathcal{Z}$. The set \mathcal{A} is an intermediate feature space like attributes or phonemes. Some oracle provides the labels for the unseen categories in \mathcal{Z} using the feature space of \mathcal{A} . In traditional zero-shot learning experiments, instances from the unseen categories in \mathcal{Z} are not used to learn the transfer function $g : \mathcal{X} \rightarrow \mathcal{A}$. This makes sense if obtaining examples of the unseen categories is difficult as in [40, 53].

Because we already had a nearly exhaustive set of scene categories in the SUN Attribute dataset, the attribute classifiers were trained using images that belonged to categories that were held out during the “zero-shot” testing of the transfer function $h : \mathcal{A} \rightarrow \mathcal{Z}$. In our “zero-shot” experiment, all of the possible scene category labels in \mathcal{Z} were held out. The experiments conducted using scene attributes as features in this subsection are an expanded version of traditional zero-shot learning, and we have maintained that term to support the demonstration of how a scene category can be identified by its typical attributes only, without any visual examples of the category. The entire “zero-shot” classification pipeline in this section never involved showing the classifier a visual training example of any scene category. The classifier gets an oracle feature listing the typical attributes of each of the 397 categories.

Our goal is to show that given some reasonable estimate of a scene’s attributes it is possible to estimate the scene category without using the low-level features to classify the query image. Scene attributes are correlated with scene categories, and query scenes can be successfully classified if only their attributes are known. In this sense our experiment is similar to, but more stringent than canonical knowledge transfer experiments such as in Rohrbach et al. because the scene category labels were not used to help learn the mapping from pixel-features to attributes [64].

Despite the low number of training examples (397, one oracle feature per category, for zero-shot features vs. $n \times 397$ for pixel-level features), the zero-shot classifier shown in Fig. 2.17 performs about as well as the gist descriptor. It does, however, perform significantly worse than the attribute-based classifier trained on n examples of predicted attributes shown in Fig. 2.16. Averaging the attributes into a single “characteristic attribute vector” for each category is quite lossy. In some ways, this supports the argument that there is significant and interesting intra-category variation of scene attributes.

2.7 Predicting Human Confusions

Scene classification is a challenging task, even for humans. In the previous sections, we show that attributes do not always out-perform low-level features at scene classification. Fig. 2.18 shows the performance of several features at another challenging task - predicting human confusions for scene classification on the SUN 397 dataset. At this task, attributes perform slightly better than any other feature.

We compare the confusions between features and humans using the scene classification confusion matrices for each feature. The human classification confusion for the SUN 397 dataset is reported in [83]. We determined that a feature classifier and the humans had the same confusion if the largest off-diagonal elements of the corresponding rows of their confusion matrices were the same, e.g. both the attribute classifier and the human respondents confused ‘bayou’ for ‘swamp’.

In [83], the low-level features that performed the best for scene classification also performed the best at predicting human confusions. Here we demonstrate that although predicted attributes do not perform as well as HoG 2x2 features at scene classification, they are indeed better at predicting human confusions.

This result supports the conclusions of [29]. Attributes, which capture global image concepts like structure and affordance, may be closer to the representations humans use to do rapid categorization of scenes than low-level image features by themselves.

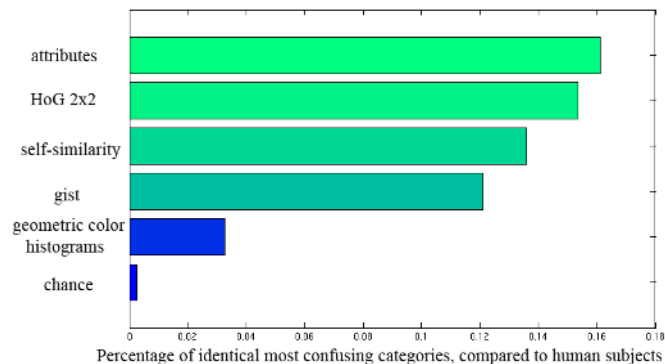


Figure 2.18: *Comparison to Human Confusions*. Using the human scene classification confusions from [83], we report how often the large incorrect (i.e. off-diagonal) confusion is the same for a given feature and the human classifiers.

2.8 Conclusions from building the SUN Attribute Dataset

In this chapter, we use crowdsourcing to generate a taxonomy of scene attributes and then annotate more than ten thousand images with individual attribute labels. We explore the space of our discovered scene attributes, revealing the interplay between attributes and scene categories. We measure how well our scene attributes can be recognized and how well predicted attributes work as an intermediate representation for zero-shot learning and image retrieval tasks.

Scene attributes are a fertile, mostly unexplored recognition domain. Many attributes are visually quite subtle, and new innovations in computer vision may be required to automatically recognize them. Even though all of our attribute labels are global, many attributes have clear spatial support (materials) while others may not (functions and affordances). Further experimentation with scene attributes will lead to better ways of describing scenes and the complicated events that take place in them.

Chapter 3

Scalable Dataset Creation

3.1 Introduction

Objects and scenes may seem like entirely different problems. Scenes are full of sub-events that make them complicated and tricky to separate into categories, while objects are obvious each of a single defined category. However, like with scenes object category labels provide a limited approximation of the human understanding. The categorical object model has several of the same limitations present with scene categories: (1) We have no way to express *intra*-category variations, e.g. “fresh apple” vs. “rotten apple.” (2) A categorical representation alone cannot help us to understand the state of objects relative to other objects in a scene. (3) The categorical object model indeed prevents researchers from responding to complex questions about the contents of a natural scene, the same way it did with scenes. The final limitation in our list is a particular obstacle in Visual Question Answering [2] or the Visual Turing Test [25].

To alleviate these limitations, we aim to add semantic visual attributes [23] to objects. The space of attributes is effectively infinite but the majority of possible attributes (e.g., “This man’s name is John.”, “This book has historical significance.”) are not interesting to us. We are interested in finding attributes that are likely to visually distinguish objects from each other (not necessarily along categorical boundaries). In this chapter, we expand on the type of attributes introduced by Farhadi et al. and explore object attributes that are essential for “common sense reasoning” (as described in [45]) about both inter- and intra-category variations.

Discriminative visual attributes provide important details about object instances. By creating a dataset

of attribute annotations for objects in the Microsoft COCO (Common Objects in Context) dataset [45], we enable a broader and deeper understanding of what is happening in the images. In Sec. 3.3, we explain how we determined which attributes to include in our dataset. To collect attributes for MS COCO, we implement a crowd-in-the-loop content generation system. Sec. 3.4 illustrates the burden of taking a naïve approach to attribute labeling. In that section we exhaustively label all of our discovered attributes for a subset of 6500 object instances. This ‘exhaustive’ sub-dataset is then used to bootstrap our economic labeling pipeline described in Sec. 3.5.

The Microsoft COCO dataset contains 500,000 images and 2M individually annotated objects. Given the scale of this dataset, it is economically infeasible to annotate all attributes for all object instances. The Economic Labeling Algorithm (ELA) introduced in Sec. 3.5 intelligently selects a subset of attributes that is likely to contain all of the positive labels for a novel image. By labeling the attributes most likely to be positive first, we are able to reduce the number of annotations required without greatly sacrificing overall label recall. We annotate objects from 29 of the most-populated MS COCO object categories with nearly 200 discovered attributes.

Currently, our MS COCO Attributes dataset comprises 84,044 images, 188,426 object instances, 196 object attributes, and 3,455,201 object-attribute annotation pairs. Multiple objects are often present in one image. The objects in the dataset vary widely, from cars to sandwiches to cats and dogs. These categories are visually distinctive, but have wide ranging intra-category variations.

The images in the MS COCO dataset are more complex than those in many similar datasets – they are objects and scenes from messy daily life. Objects often occlude or interact with each other, and scenes may be unusually presented compared to a prototypical example of the corresponding scene category. This complexity makes collecting an attribute taxonomy a even more challenging task. In Sec. 3.3 we employ proven techniques, such as text-mining, image comparison tasks, and crowd shepherding, to find the attributes we later use to label the dataset [5, 13, 18, 56, 65].

Our contribution is straightforward — we obtain attribute labels for thousands of object instances at a reasonable cost. For the sake of estimating an upper bound on the cost of annotating attributes across the MS COCO dataset, let us assume several `/Users/gen/dissertation/figures/` relating to the number of annotations and cost per annotation using the widely employed Amazon Mechanical Turk platform (AMT).

Let’s assume that crowd workers are asked to annotate 50 images per human intelligence task (HIT).

Our dataset contains approximately 200 visual attributes. For MS COCO Attributes, we annotate attributes for a subset of the total MS COCO dataset, approximately 180,000 objects across 29 object categories. The cost of exhaustively labeling 200 attributes for all of the object instances contained in our dataset would be : $180k \text{ objects} \times 200 \text{ attributes} / 50 \text{ images per HIT} \times (\$0.07 \text{ pay per HIT} + \$0.01 \text{ Amazon fee}) = \$57,600$. The pay per HIT was selected using the authors’ previous experience using AMT. If we annotate each attribute for the top 10% of object instances mostly likely to contain a particular attribute, the overall annotation cost would drop to a reasonable \$6,480. But how do we discover the most informative and characteristic attributes for the objects in the MS COCO dataset, and how do we estimate which images are most likely to contain each attribute? We present our answer to this question in Sec. 3.5.

Every dataset inherently suffers from annotator mistakes which reduce the overall ground truth recall of the dataset. Hopefully, only a small number of positive or negative examples of a given label are missed. Our ELA introduces a new source of dataset error. Because we are purposefully annotating only a subset of the total labels, we are bound to miss some positive examples that would have been found if the dataset were exhaustively annotated. In Sec. 3.5 we show that it is possible to obtain 80% recall of exhaustively obtained MS COCO Attribute labels while only asking for 10% of the attributes to be labeled, and 90% recall if we label 20% of the attributes.

To further verify the quality of the MS COCO Attributes dataset, we explore attribute classification. In Sec. 3.6, we show that a CNN finetuned on our ELA labeled training set to predict multi-label attribute vectors performs similarly to classifiers trained on exhaustively labeled instances.

3.2 Related Work

To our knowledge, no attribute dataset has been collected containing both the number of images and the number of object attributes as our MS COCO Attributes dataset. Existing attribute datasets concentrate on either a small range of object categories or a small number of attributes.

One notable exception is the Visual Genome dataset, which also aims to provide a dataset of complex real-world interactions between objects and attributes [38]. Krishna et al. create a dataset with myriad types of annotations, all of which are important for deeper image understanding. For MS COCO Attributes, we focus on making the largest attribute dataset we possibly can. In that regard we have been able to collect more than double the number of object-attribute pair annotations. MS COCO Attributes and the Visual Genome dataset together open up new avenues of research in the vision community

by providing non-overlapping attribute datasets. The dataset collection effort illustrated in this chapter aims to scale up the attribute collection demonstrated in well-cited attribute literature such as the CUB 200 dataset, the SUN Attribute dataset, Visual Genome, and other important works of attribute annotation [3, 38, 54, 56, 80].

Initial efforts to investigate attributes involved labeling images of animals with texture, part, and affordance attributes [22, 23, 40]. These attributes were chosen by the researchers themselves, as the interesting attributes for animals were clear at the time of publication. MS COCO dataset images are more complicated than those in Farhadi et al. . They often have object occlusions and complicated backgrounds. The MS COCO Attributes are more detailed and descriptive than those in earlier datasets.

Other attribute datasets have concentrated on attributes relating to people. Kumar et al. and Liu et al. introduced datasets with face attributes and human activity affordances respectively [39, 48]. The influential Poselets dataset labeled human poses and has been crucial for the advancement of human pose estimation [4].

Vedaldi et al. used a specialized resource for collecting attributes [78]. They collected a set of discriminative attributes for airplanes by consulting hobbyist and expert interest websites. It is possible that the best method for collecting high-quality attributes is to use a more sophisticated crowd, reserving the general-expertise crowd to label the dataset. We explore bootstrapping the attribute discovery process by mining discriminative words from a corpus of descriptive text written by language ‘experts’ — novels and newspapers. Similar methods have been demonstrated successfully in [3, 54, 65]. We use the descriptive words found in these texts to seed a crowd pipeline that winnows the large variety of seed words down to the attributes that visually describe the MS COCO objects.

Several datasets have collected attributes for the purpose of making visual search more tractable. The Whittlesearch dataset contains 14,658 shoe images with 10 instance-level relative attributes [36]. Parikh and Grauman show that predicted attributes can be used to better describe the relative differences between objects of the same and different categories [55]. Our experiments in the following sections further the attribute annotation and recognition research begun in those papers by concentrating on scaling up the size of the attribute dataset.

A number of past projects sought to bootstrap dataset annotation using active learning [1, 9, 60, 79]. While these references demonstrate the usefulness of active learning, we elect not to employ similar methods. Vijayanarasimhan and Grauman and Patterson et al. show that the crowd in combination with

active learning can rapidly converge on a visual phenomena. However, these methods result in high precision classifiers trained on data that may be missing the most visually unusual examples of a particular visual phenomena. While we seek to annotate this dataset with maximum efficiency, we choose not to make the visual approximation inherently imposed by an active learning pipeline. We eliminate the possible bias that would be introduced by using visual classifiers to comb an unlabeled dataset for likely instances of a given attribute.

Admittedly, our Efficient Labeling Algorithm (ELA) accepts the possible bias that may occur when we label a subset of the total number of attributes. We determine that it would be of more benefit to the vision community to have a dataset without the visual bias incurred by any particular classification system. Section 3.5 describes the trade-offs among visual diversity, label accuracy, and annotation cost.

We identify additional cost-saving annotation strategies by imitating successes in multi-class recognition [12, 14]. Deng et al. define the Hierarchy and Exclusion (HEX) graph, which captures semantic relationships between object labels [12]. HEX graphs describe whether a pair of labels are mutually exclusive, overlap, or subsume one or the other. In Deng et al. HEX graphs are used for object classification. We use the hierarchy of the MS COCO objects to inform our economic labeling algorithm (ELA), described in Sec. 3.5.

Ultimately, we introduce a large new dataset and a novel way to cheaply collect annotations without introducing visual bias. The experiments presented in Secs. 3.5 and 3.6 corroborate related work that shows large scale datasets can be rapidly collected in a manner that is both economically efficient and robust to annotation noise [67, 77]. Sec. 3.5 reassuringly shows that a dataset collected with the ELA protocol has similar annotation density as an exhaustively labeled dataset. Finally, our comparison of two classification paradigms, recognizing attributes individually or in a multilabel setting, reveals that classifiers trained on our ELA collected data are similarly powerful to classifiers trained on exhaustively labeled data.

3.3 Attribute Discovery

The first stage of creating the MS COCO Attributes dataset is determining a taxonomy of relevant attributes. For MS COCO Attributes, we search for attributes for all of the object categories contained under the MS COCO super-categories of Person, Vehicle, Animal, and Food. These categories are person, bicycle, car, motorcycle, airplane, bus, train, truck, boat, bird, cat, dog, horse, cow, sheep, elephant, bear, zebra, giraffe, banana, apple, orange, broccoli, carrot, hot dog, pizza, donut, cake, and sandwich.

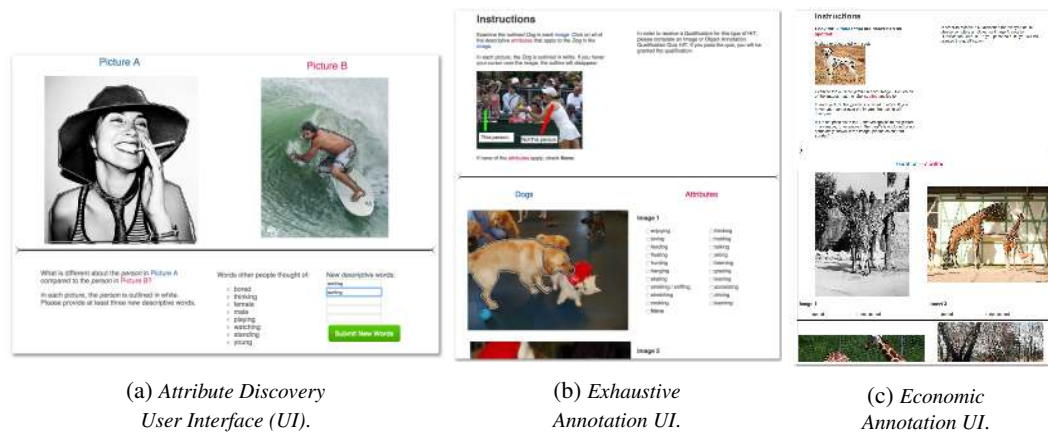


Figure 3.1: Amazon Mechanical Turk (AMT) Task Interfaces used in the creation of MS COCO Attributes.

We must determine the attributes that would be useful for describing these objects.

Asking Amazon Mechanical Turk (AMT) workers to describe the objects from scratch might result in terms that do not generalize well to other objects in the same hierarchical group or are too common to be discriminative, for example ‘orange’ does not help us describe the difference between oranges. To bootstrap the attribute discovery process, we mine a source of English text likely to contain descriptive words – the New York Times Annotated Corpus [69]. This corpus contains all of the articles published by the NYT from 1987–2008. We extract all adjectives and verbs occurring within five words of one of our object words. This results in hundreds of descriptive words. Unfortunately, not all of these candidate attributes describe visually recognizable phenomena.

In order to filter the attributes mined from the NYT corpus, and indeed add a few new ones, we design an AMT Human Intelligence Task (HIT). Our attribute discovery HIT, shown in Fig. 3.1a, encourages AMT workers to submit visual attributes by asking them to discriminate between two images. In this experiment, we show workers two randomly selected MS COCO objects from the same category. The worker types in several words that describe one of the images but not the other. To help focus our workers and guide them to make better suggestions, we show a random subsampling of the attributes discovered via the NYT corpus or submitted in previous HITs.

In the end, approximately 300 unique terms were submitted by AMT workers to describe the 29 different categories. The authors manually condensed the combined list of NYT corpus attributes and AMT worker attributes. Attributes that do not refer to a visual property were also removed, e.g. ‘stolen’ or ‘unethical’. The final attribute list comprises 196 attributes.

3.4 Exhaustive Annotation

In annotating our attributes we would like to avoid asking redundant questions (e.g. asking if a person is “sitting” when they’re already labeled as “standing”). To intelligently avoid these situations we need to understand the correlations among attributes. We first build an exhaustively annotated dataset that has a ground truth label obtained via the crowd for every possible object-attribute pair. Our exhaustively labeled dataset serves as a training set for the ELA method introduced in Sec. 3.5. A portion of the exhaustively labeled set is set aside as a validation set to measure the performance of the ELA.

To create the exhaustively labeled part of the MS COCO Attributes dataset, we employ the annotation UI shown in Fig. 3.1b for AMT. The object instances in this part of the dataset were chosen as follows: for all categories we exhaustively annotate 10% of object instances that are larger than 32×32 px. AMT workers are shown 10 images per HIT and 20 possible attributes subsampled from the total 196. Workers are asked to check all attributes that apply to the object outlined in white. The attributes are roughly grouped by type, such as action word, emotion, surface property, etc.

We annotate a total of 20,112 object instances with all 196 attributes (5000 person instances and approximately 500 instances of every other object). Three different workers annotate each object-attribute pair. If two or more annotators agree on a positive annotation, we consider that a true positive.

Responding to comments from our workers, we pay \$0.10 per exhaustive annotation HIT. In total, this portion of the dataset cost : $20112 \text{ images} \times 196 \text{ attributes} / \text{avg. } 196 \text{ annotations per HIT} \times (\$0.10 \text{ pay per HIT} + \$0.01 \text{ Amazon fee}) \times 3 \text{ workers repeat each annotation} \approx \$6,637$. If we continued this annotation policy to annotate the remaining ‘person’, ‘animal’, ‘vehicle’, and ‘food’ objects from MS COCO (285k instances), the total annotation cost would be \$94,050. Using the ELA, we will be able to accomplish this task for only $\sim \$27,400$.

To improve annotation quality, we implement several quality control techniques. We require that workers complete an annotation quiz in order to begin working on HITs, we identify poor workers using a statistical evaluation of worker meta-data, and we manually review workers to block those who are submitting sub-par work. The quiz looks identical to the UI in Fig. 3.1b. The worker is required to score 90% recall of the attributes present in that HIT.

3.5 Economic Labeling

Attributes in many domains are sparse. With 196 attributes, we find that across all 29 categories, the average number of positive attributes per object is 9.35. Ideally, we could identify the exact attributes that are positive for each object and only ask the AMT workers to annotate (or verify) those attributes. Annotating a new dataset with a huge number of possible attributes would then be relatively inexpensive. However, we do not possess an oracle capable of identifying the perfect set of attributes to ask about, and we choose not to use active learning with visual classifiers to avoid bias.

Without the benefit of an attribute oracle, we apply a method of selecting attributes that are likely to be positive for a given object instance. For an unlabeled object, we calculate the probability that a given attribute a_i is true using Eqn. (3.1). The likelihood of the attribute given the category is the frequency of that attribute in all observations of that category.

$$P(a_i = 1|y) = \frac{N_{a_i,y}^1}{N_y} + \alpha \quad (3.1)$$

To avoid a zero count for a rare attribute, we augment this probability with the α term explained in Eqn. (3.2). The value of α depends on the observed population of a_i in the hierarchical super-category containing y , which has K total sub categories. A Beta prior could be applied to the category, or a hierarchical Beta prior could be used to model the relationship between object category and super-category. In this initial investigation, we opt for simplicity.

$$\alpha = \frac{\sum_{x=1}^K N_{a_i,x}^1}{\sum_{x=1}^K N_x} \quad (3.2)$$

Finally, the most likely attribute given the category y is determined by Eqn. (3.3).

$$a_i = \arg \max_{a_i \in A} P(a_i = 1|y) \quad (3.3)$$

Essentially, our economic labeling algorithm follows these steps: (1) Obtain an exhaustively annotated training set \mathcal{T} . (2) For each object in the unlabeled dataset \mathcal{D} label the most popular attribute from \mathcal{T} , using Eqn. (3.1). (3) For each partially labeled object in \mathcal{D} select the subset of examples from \mathcal{T} that share the labels given to that object. Eqns. (3.1) and (3.2) are slightly changed in this step so that y represents all object instances of category y that also share previously labeled attributes. (4) Annotate the attribute determined by Eqn. (3.3). (5) Continue this process until each object in \mathcal{D} has at least N

attributes labeled. This process is more precisely described in Algo. 2.

Input: Dataset \mathcal{D} of unlabeled images, fully labeled training set \mathcal{T} , labels to annotate A

Output: Labeled dataset \mathcal{D}'

```

1 for  $I_j \in \mathcal{D}$  do
2      $\triangleright I_j$  is an unlabeled image from  $\mathcal{D}$ ,  $j \in \{0, N_D\}$ 
3     while NumLabels( $I_j$ ) <  $N$  do
4          $\triangleright$  Repeat annotation until  $N$  labels are acquired
5          $\mathcal{D}_S = \text{MatchingSubset}(I_j, \mathcal{D})$ 
6         if isEmpty( $\mathcal{D}_S$ ) then
7              $\mathcal{D}_S = \text{AltMatchingMethod}(I_j, \mathcal{D})$ 
8         end
9          $\mathcal{Q}_n = \text{SelectAttributeQuery}(\mathcal{D}_S)$ 
10         $I_j[n] = \text{Annotate}(\mathcal{Q}_n)$ 
11    end
12 end
13 return  $\mathcal{D}'$ 

```

Algorithm 1: Economic Labeling Algorithm (ELA)

Our ELA may seem straight forward. There is however one stage of the ELA that presents a problem. What should be done if the subset of \mathcal{T} returned by the function `MatchingSubset` in Algo. 2 is empty? We explore four possible alternatives for overcoming the problem of an uninformative subset.

Our four alternative varieties of the `MatchingNeighbors` method are only used if either the matching subset is empty or the remaining attributes from the matching subset have no positive labels. Otherwise, the ELA continues to ask for annotation of the most popular attributes in decreasing order. For our experiments, we also deemed a matching subset to be “empty” if it contained fewer than 5 matching instances.

The first alternative is the Random method, which randomly selects the next attribute to label from the set of unlabeled attributes. The second strategy is the Population method, which proceeds to query the next most popular attribute calculated from the whole labeled set \mathcal{T} .

Our third alternative, Backoff, retreats backward through the previously calculated subsets until a subset with a positive attribute that has not been labeled is found. For example, if a dog instance is annotated first with ‘standing’ and then with ‘not furry’, there may be no matching dog instances in the training that have both of those labels. The Backoff method would take the subset of training instances labeled ‘dog’ and ‘standing’, calculate the second most popular attribute after ‘furry’, and ask about that second most popular attribute. The Backoff method is similar to the Population method except that the Population method effectively backs off all the way to the beginning of the decision pipeline to decide the next most popular attribute.

The fourth method we explore is the Distance method. This alternative uses the current subset of annotated attributes as a feature vector and finds the 100 nearest neighbors from the set \mathcal{T} , given only the subset of currently labeled attributes. For example, if a partially labeled example had 10 attributes labeled, the nearest neighbors would be calculated using the corresponding 10-dimensional feature vector. The next most popular attribute is selected from the set of nearest neighbors.

In order to compare these alternative methods, we split our exhaustively labeled dataset into test and train sets. We use 19k object instances for the training set and 1k object instances for test. The object instances are randomly selected from all 29 categories. In our simulation, we use the ELA methods to generate the annotated attributes for the test set. For each object instance in the test set, we begin by knowing the category and super-category labels. For example, given a test image we might know that it is a ‘dog’ and an ‘animal’.

We proceed by following the steps of the ELA up to a limit of N attribute queries. We determine the response to each query by taking the ground truth value from the test set. In this way we simulate 3 AMT workers responding to an attribute annotation query and taking their consensus. After N queries, we calculate the recall for that instance by comparing the number of positive attribute annotations in the ELA label vector to the exhaustive label vector. If we use the ELA to label a ‘dog’ instance up to 20 queries and obtain 8 positive attributes, but the ground truth attribute vector for that dog has 10 positive attributes, then the recall for that instance is 0.8.

We compare the four alternative methods in Fig. 3.2. Each method is tested on 1k object instances, and the mean recall score averages the recall of all test instances. In all of the plots in Fig. 3.2, the four methods perform approximately the same for the first 10 attribute queries. This is to be expected as none of the methods will perform differently until the partially labeled instances become sufficiently distinctive to have no matching subsets in the training set.

After approximately 10 queries, the methods begin to diverge. The Random method shows linear improvement with the number of queries. If these plots were extrapolated to 196 queries, the Random method would achieve a recall of 1.0 at query 196. The other methods improve faster, approaching perfect recall much sooner, thus at less expense. The Backoff method initially out-performs the population method, indicating that at early stages querying the popular attributes from a more specific subset is a better choice than querying the most popular attributes overall. This distinction fails to be significant after more queries are answered.

The best performing method is the Distance method. This method comes closer to selecting the

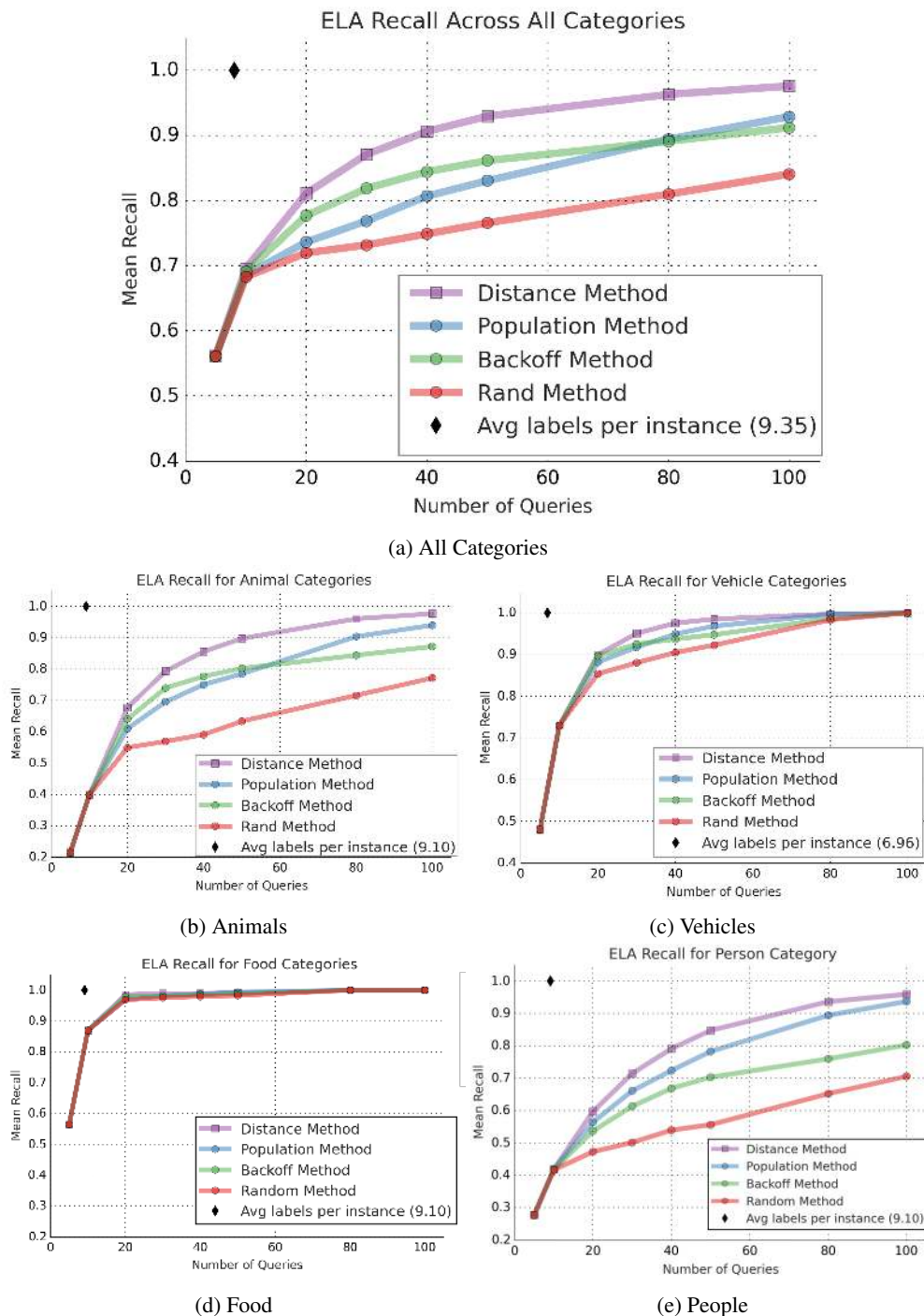


Figure 3.2: Mean Recall Comparison of Alternative ELA methods. Fig. 3.2a plots the mean recall of the test dataset alternatively labeled with each ELA method and stopped for a range of query limits. All categories were included in this comparison. The Distance method is the clear winner, obtaining 80% recall for only 20 attribute queries, approximately 10% of the total number of attributes. The sub-figures above show the mean recall across all test instances of their type of category. The vehicle categories achieve a higher recall with fewer queries than the animal categories. This may be due to the smaller subset of attributes relevant to vehicles than to animals. People are more difficult to label approximately while still achieving high recall, while food is the easiest.

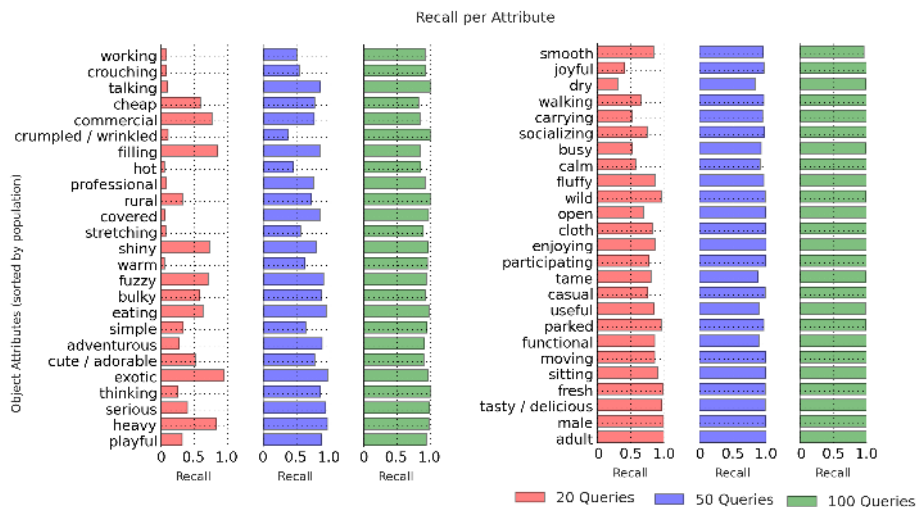


Figure 3.3: *Mean Recall Across all Categories for 50 Attributes*. This plot shows the recall of the ELA-Distance method for annotating 50 randomly selected attributes from the full set of 196. The recall is calculated across all instances of the exhaustively labeled test set. The attributes are sorted by their popularity in the exhaustively labeled dataset. A version of this plot showing all attributes is available in Fig. B.3.

best subset of attribute queries both for the full hierarchy and for the animal and vehicle sub-trees. The success of the Distance method indicates that the most likely next attribute for a given image may be found by looking at examples that are similar but not exactly the same as a given object.

To further examine the performance of the ELA with the Distance method, we plot a selection of per attribute recall scores in Fig. 3.3. The attributes in Fig. 3.3 are sorted by ascending population in the dataset. One would expect the more popular attributes to have higher recall than the less popular attributes for a lower number of attribute queries. This is not strictly the case however. ‘Stretching’, for example, is a popular attribute, but does not obtain higher than 0.9 recall until 100 queries. This indicates that ‘stretching’ is not strongly correlated with other attributes in the dataset. Conversely, even at 20 attribute queries many of the rarer attributes still have a reasonable chance of being queried.

We also attempt hybrid versions of the methods described above. We repeat the simulation shown in Fig. 3.2 by first annotating the top 10 most popular attributes, and then continuing with the alternate methods. In this way we might be able to discover unusual objects early, thus making our method more robust. However, the performance of the hybrid methods were barely different than that shown in Fig. 3.2.

Attribute Annotation using the ELA

For ELA attribute annotation, we ask workers to label one attribute at a time. We cannot use the UI from Fig. 3.1b. Instead we ask the AMT workers to select all positive examples of a single attribute for a set of images from a given category, example shown in Fig. 3.1c. We elect to ask for fewer annotations per HIT in the ELA stage (50 object-attribute pairs) than in the exhaustive stage (200 object-attribute pairs). This choice was made to lessen worker fatigue and improve performance. The difference in worker performance for the exhaustive and ELA HITs is discussed more in the supplemental materials.

Thus far we have collected approximately 3.4M object-attribute pairs. Ultimately, the goal of this project is to annotated at least 40 attributes for every object instance, thus obtaining an estimated dataset recal of 90% according to Fig. 3.2a. At this time, we have collected at least 20 attributes for 24,492 objects at a cost of \$2351 ($\sim 24\text{k objects} \times 20 \text{ attributes} / 50 \text{ attributes per HIT} \times 3 \text{ repeat workers} \times \0.08 per HIT). If we used the exhaustive annotation method, this would have cost \$8,082.

A visualization of the collected dataset is shown in Fig. 3.4. The data points in Fig. 3.4 were obtained by applying a 2D t-SNE dimensionality reduction of the 196-D attribute vectors for each point in the dataset. If an object was labeled using the ELA and has fewer than 196 attribute labels, the remaining unlabeled attributes are assumed to be false. Only objects with at least 20 attribute labels are shown in Fig. 3.4.

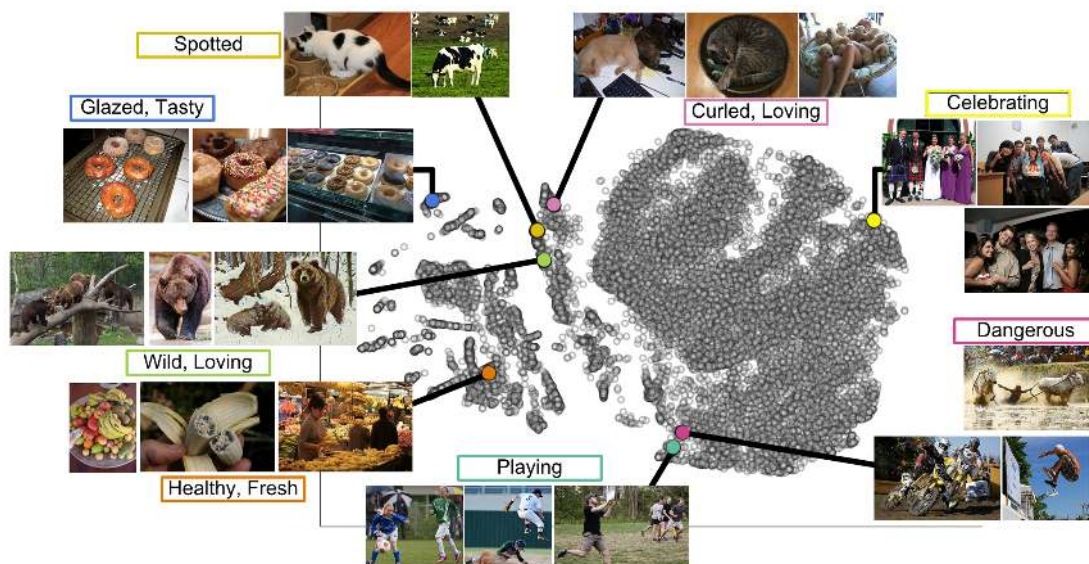
Even though this feature representation of the dataset does not contain the object labels, objects of similar categories are often grouped together. Objects of different categories are near to each other when they share attributes. Fig. 3.4a qualitatively suggests that our attributes are indeed expressing interesting inter- and intra-category variations.

In the end, the MS COCO Attributes dataset has a variety of popular and rare attributes. 75% of attributes have more than 216 positive examples in the dataset, 50% have more than 707, and 25% have more than 2511. Fig. 3.6 shows some qualitative examples from MS COCO Attributes.

3.6 Attribute Classification

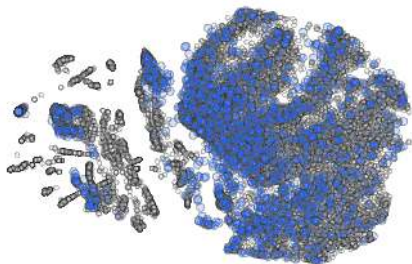
Ultimately, attribute labels are only useful if visual classifiers can be built from them. To verify the detectability of our attributes, we trained 1 vs. All classifiers for each attribute, agnostic of object category. Fig. 3.7 shows AP scores of 100 randomly selected attribute classifiers.

To train the attribute classifiers, features for each object instance’s bounding box were extracted using



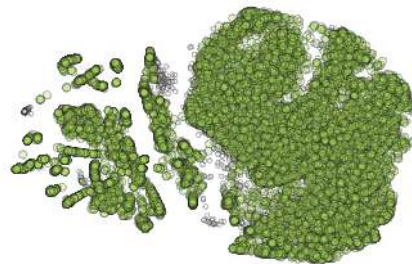
(a) Sample Instances and their Nearest Neighbors

●●● Exhaustively Labeled Instances



(b) Exhaustively Labeled Instances. Cost \$6,637.

●●● ELA Labeled Instances



(c) ELA Labeled Instances. Cost: \$2,351.

Figure 3.4: MS COCO Attribute Dataset show in *t*-Stochastic Nearest Neighbor Embedding (*t*-SNE) [76].

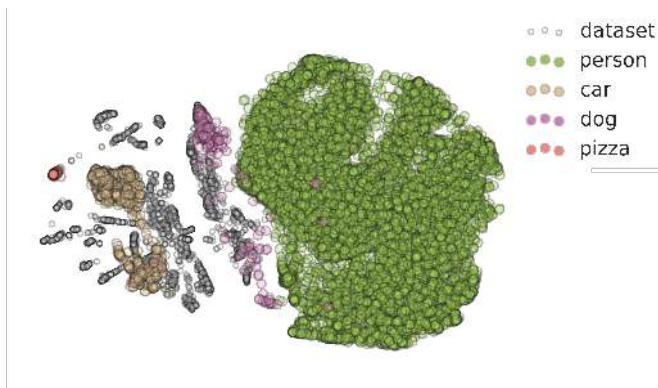


Figure 3.5: *t-SNE Visualization of 4 Object Categories in Attribute Space*. The 2D t-SNE embedding shown above marks every point in dataset with a gray circle. All dataset points are instances in the dataset, represented by their ELA generated attribute vector. This embedding was created without knowledge of the object category for each instance. However, this plot demonstrates that attributes are discriminative of object category.



Figure 3.6: *Examples from MS COCO Attributes*. These are positive examples of the listed attributes from the dataset. Examples such as the man cuddling a horse or the dog riding a surfboard shows how this dataset adds important context to image that would otherwise be lost by only listing the objects present in an image.

the pretrained Caffe hybridCNN network released with the publication of the Places Dataset [33, 86]. For the features used in these classifiers, we take the output of the final fully connected layer, randomly subsample 200 dimensions, and apply power normalization as per the recommendations of [63]. We then train a linear SVM using all available positive and negative examples for that attribute. Chance is calculated as the ratio of true positives to total training examples for each attribute. To fairly compare

the AP score of classifiers trained with differing numbers of true positives, we employ a normalized AP score as suggested by Hoiem et al. [31].

As a counterpoint to recognizing attributes in isolation, we trained a multi-label CNN to simultaneously predict all attributes for a novel test image. We created this network by fine-tuning the BVLC reference network from the Caffe library [33]. Our attribute network uses a sigmoid cross-entropy loss layer instead of a softmax layer to optimize for multi-label classification, as suggested by previous research in multi-label and attribute classification [28, 30, 71]. The fine-tuning was accomplished with SGD with momentum slightly higher than the reference net, but with learning rate lower and regularization stronger to account for the sparsity of positive labels in the training set.

This network, unlike the SVMs in the previous experiment, is trained with a complete attribute vector for each object in the train and test sets. Our training and test sets contained all objects in our dataset with 20 or more labeled attributes. The remaining unlabeled attributes were assumed to be false for this experiment. With this multi-label CNN, we try to understand if our attribute dataset is useful despite missing positive labels. We also want to determine if our attributes are capable of generalizing across object classes.

Fig. 3.7 compares the per attribute AP over the test set predictions from our multi-label CNN to the 1 vs. all SVMs trained independently for each attribute. This plot shows that exploiting the correlations between attributes often improves classifier performance for the CNN compared to the independent SVMs, especially for rarer attributes. Over all attributes, the mean AP for the SVMs is 0.35 compared to our CNN which has a mAP of 0.36. This indicates that the ‘missing’ labels, which we assumed false, were not detrimental to performance averaged over all attributes. Generally, both types of classifiers are performing better than chance, indicating that our attributes are recognizable across categories.

For a qualitative demonstration of classifiers trained with our MS COCO Attributes dataset, please refer to Fig. 3.9.

3.7 Worker Performance

Interestingly, worker performance improves during the ELA annotation. For both exhaustive and ELA annotation, we record each worker’s agreement with the other workers who annotated the same object-attribute pairs. We use a worker’s agreement on positive annotations to determine if a worker was doing a poor job. If a worker frequently misses annotating an attribute that was selected positive by two other workers, we consider that problematic.

AP for Attribute Classifiers trained across all Object Categories

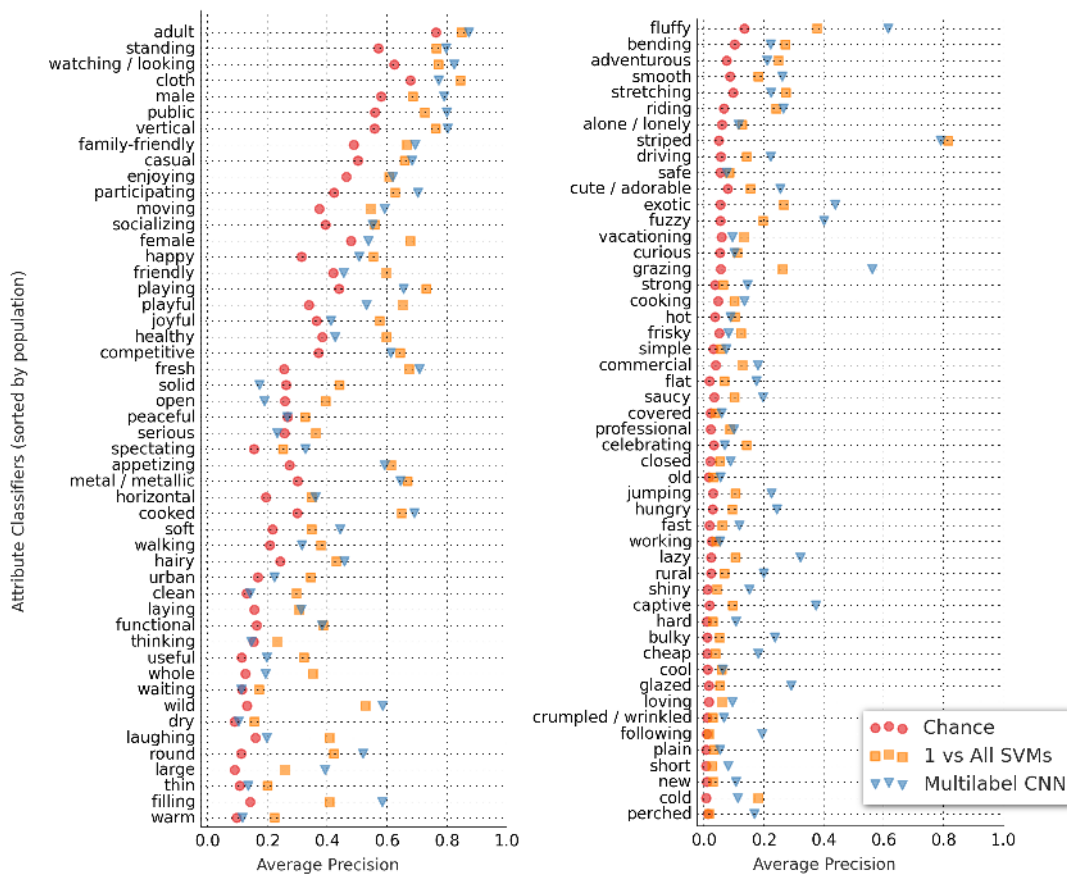


Figure 3.7: *Average Precision vs. Chance*. Performance is shown for 100 randomly selected attributes. Attributes are sorted in descending order by their population in the dataset. Each yellow square represents an SVM that was trained to recognize that particular attribute. All SVM training and test sets are composed of consensus AMT labels. A positive attribute label occurs when 2/3 AMT workers voted ‘true’ for the presence on a given attribute and 0 workers voted for false. Thus, the training and test sets for each attribute SVM are a different size. Each blue triangle represents the AP for that attribute calculated on the full multi-label test set predictions of our CNN. The objects in the training set for all classifiers shown are members of the MS COCO ‘train2014’ set, and test instances are members of the ‘val2014’ set. The ratio of train to test for each SVM is approximately 2-to-1. For the multi-label CNN, the train/test set sizes were 27k/18k. A version of this figure showing the recognition performance for all attributes is available in Fig. B.3.



Figure 3.8: *Example Attribute Classifications*. In the figure above, images from the MS COCO dataset that were not part of the MS COCO Attributes set are shown. The objects are identified with either a green or blue bounding box. Beside the test images, 5 of the most confident attribute detections for that bounding box are listed along with the SVM confidence score. The attribute classifiers used for this figure correspond to the SVM classifiers in Fig. 3.7. This figure is a qualitative example of how detected MS COCO Attributes can give a deep and broad description of the context of the object instance.

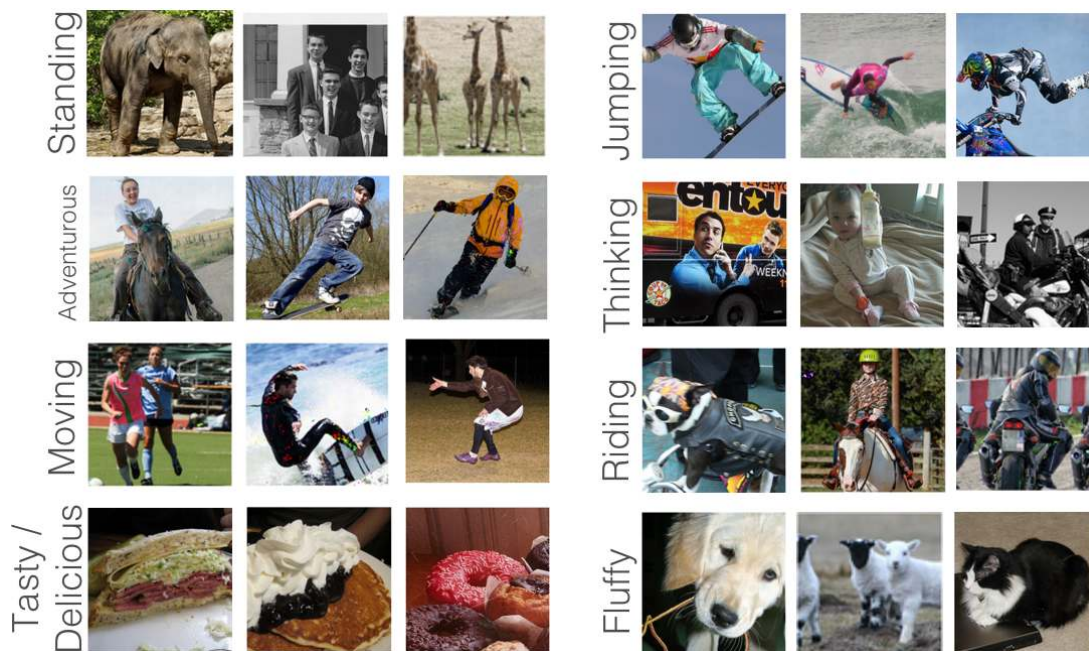


Figure 3.9: *Example Attribute Classifications*. In the figure above, images from the MS COCO dataset that were not part of the MS COCO Attributes set are shown. Several positive examples from the test set are shown for 8 attributes. The attribute classifiers used for this figure correspond to the multilabel CNN in Fig. 3.7.

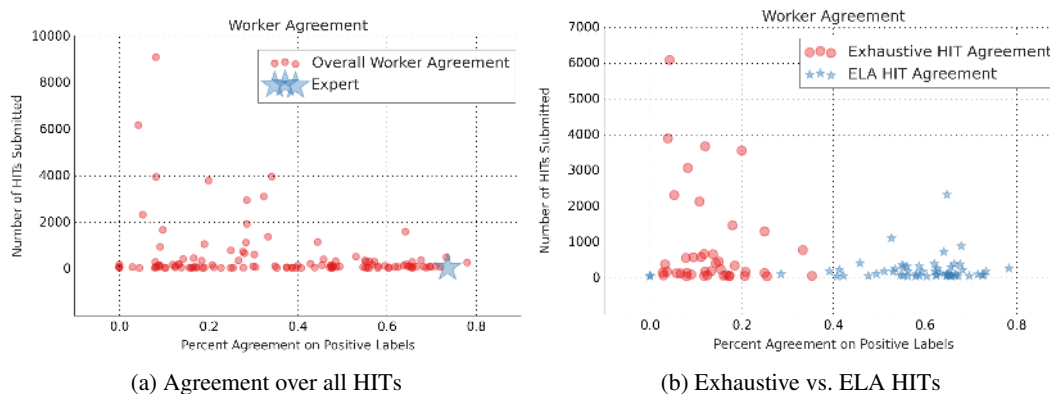


Figure 3.10: *Worker Agreement*. In both plots above, each point represents a worker who completed over 50 HITs. In Fig. 3.10a the “Expert” worker is the author of this dissertation. Fig. 3.10b demonstrates that there is greater worker consensus when the ELA format HIT is used rather than the Exhaustive format HIT. Workers are able to agree more often if they are only annotation one attribute at a time.

Both plots in Fig. 3.10 plot the agreement scores of individual workers against the number of HITs they completed. When workers fell below 10% agreement, the authors manually reviewed dozens of the questionable worker’s HITs. The worker who completed the most hits appears to have a low agreement score in Fig. 3.10a. This worker was disqualified from doing further HITs, and all their work product was discarded.

Figure 3.10b shows that the worker agreement for ELA HITs was much higher than for exhaustive HITs. This is because the object-attribute pairs presented to workers in the ELA HITs were selected expressly because they are expected to have positive labels. This means the ELA HITs did not have sparse occurrence of positive labels, and as a result worker performance improved. Improved worker agreement numbers also makes it easier to spot poor workers like the worker who has nearly 0% agreement on their ELA HITs in Fig. 3.10b.

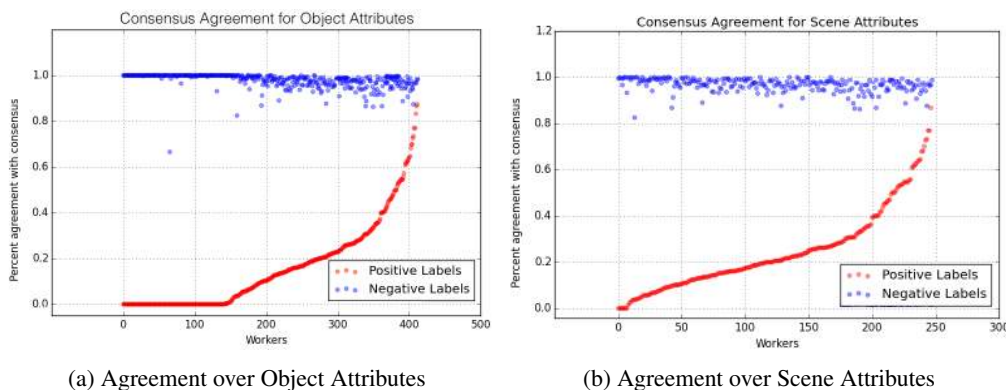


Figure 3.11: *Consensus Agreement on both Positive and Negative Labels*. In both plots above, each point represents a worker who completed over 50 HITs. The above plots show how often individual workers agreed with the consensus. Two trend lines are shown, for when the consensus labels were negative and for where they were positive. Because attributes are sparse, it is much easier for workers to agree with the consensus when the consensus is negative, i.e. if a worker labeled every attribute negative for all objects, their accuracy would be very high. Interestingly, the workers that disagree more often on negative labels have a higher rate of agreement on positive labels. This shows that attribute labeling is subjective, and that poor workers can be identified by their rate of agreement with the majority by comparing consensus agreement on positive and negative ground truth labels.

3.8 Conclusions from building the COCO Attributes Dataset

This chapter introduced an efficient method for labeling a large dataset. The attribute discovery process demonstrated in this chapter, combined with the ELA, are capable of scaling up the annotation effort to much larger dataset sizes. Further analysis of alternative selection methods could result in improved recall for low numbers of attribute queries. The ELA also has the potential for use in other multi-label annotation cases. This economical annotation method begs to be used on larger dataset annotation efforts.

Chapter 4

Using the crowd to discover features and train detectors

This chapter introduces two novel systems for employing a crowd to drive classifier and detector creation. Beginning in Sec. 4.1, we show how the crowd can be used to identify and train mid-level feature detectors. In Sec. 4.3, the power of the crowd is demonstrated to an even greater effect. The crowd trains fine-grained part and object classifiers without the benefit of a labeled dataset.

4.1 Crowd-enabled Mid-Level Feature Discovery

Recently, several publications have demonstrated state of the art performance at scene classification using discriminative patches [16, 34, 72]. These papers introduce different methods for identifying salient visual elements in the form of mid-sized image patches, and training classifiers to detect the visual phenomena observed in training patches. Singh et al. and Juneja et al. each propose pipelines for training discriminative patch classifiers, but both employ a library of discovered patches to create “bags of parts” for scene classification. The methods shown in [34, 72] show state of the art performance compared to other single features on the MIT Scene 67 dataset. Doersch et al. shows impressive capability to discriminate between the architecture of different European cities.

The insight of these papers is that scene and object categories can be separated from each other by observing a small number of visual events that are highly discriminative. Unlike bag of words models, not all of the locations are equally discriminative – only a sparse set of locations in an image are useful

for determining the category. Detectors trained on discriminative patches can build features that are more useful for scene classification than directly using low-level features. This would be similar to identifying only the most discriminatively powerful visual words, and only using those words in the bag of words codebook. Patches also have the advantage of being larger than the typical visual word, thus enabling them to capture a visual element that could have higher-level semantic significance.

While there are several proposed methods to discover discriminative patches, [34, 72], we examine an interesting alternative to the often time consuming methods of automatic discriminative patch discovery. We directly ask non-expert humans to select visually and semantically similar image patches. We train classifiers to recognize the human-identified visual elements. Our motivation stems from the observation that humans are capable of rapidly discriminating scenes, and thus can be considered an upper bound on identifying which elements of scenes are the most salient.

Other human-in-the-loop methods such as [5, 26, 37] demonstrate success at visual classification tasks. Humans (often non-expert, crowdsourced humans) are commonly used in vision algorithms at two stages (a) annotation time, exhaustively annotating a dataset or (b) test time, coupled with a computational method to improve human accuracy and / or reduce human effort. In contrast, we put humans in the loop at neither annotation time nor test time, but rather at “representation discovery” time. The humans are directly telling the computer which visual elements should be discriminative. This has some similarity to part-based annotation of visual phenomena, except that our method does not require any explicit semantic meaning for the parts. In the experiments of Chapter 4, the humans never see entire training images, only sets of candidate image patches. Putting humans “in the loop” at this stage in a recognition algorithm is a novel investigation of incorporating users into the classifier creation process.

Our starting point for building human-in-the-loop patches is the automatic method presented in Singh et al. We first find candidate patches within the 15 scene dataset [44] using the code of [16, 72]. The algorithm selects 1200 random patches from each category and for each patch finds its 24 nearest neighbors. Each random patch and its nearest neighbors are a candidate discriminative patch model. Individual patches are 80x80 pixel windows represented as 2112 dimensional HoG features.

At this point the Singh et al. use an iterative cross-validation method to discover sets of 5 patches that when used to train a linear SVM result in discriminatively powerful classifiers. Instead of this computationally expensive process, we present AMT workers with a page showing a group of 25 nearest neighbor patches. We create a patch selection task for 400 randomly selected patch groups from each category in the 15 scene dataset. our user interface is shown in Fig. 4.7.

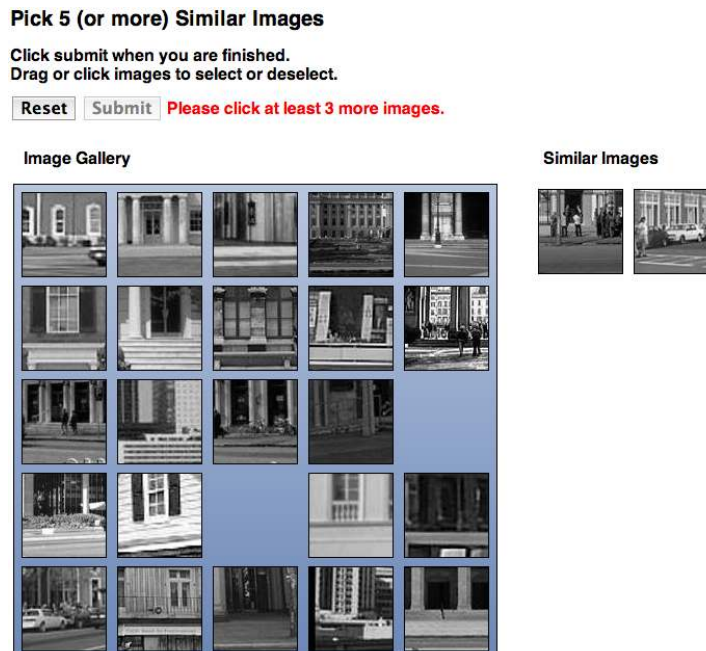


Figure 4.1: *AMT patch cluster refinement interface*. Users are presented with a group of 25 nearest neighbor patches. These user-selected patches are used to train a model for a discriminative patch.

Using the interface in Fig. 4.7, we obtain 3 user responses for each nearest neighbor group. We manually examine the user responses to this task and only observe a few spurious responses out of thousands of HITs. This suggests that this HIT is not attractive to cheating Turkers. We discard redundant responses where the selected patches are nearly the same and train discriminative patch models from the remaining responses. For each discriminative patch classifier, the user selected patches are the positive examples and a large set of randomly selected patches from other categories are the negative examples.

In the AMT UI, the image patch in the top left corner is the center of the cluster. The images patches following in left-to-right, row-follow-row reading order are the nearest neighbors of increasing distance. Figure 4.2 is a heat map of which grid locations were most often selected to be part of the final coherent group. Interestingly, the first nearest neighbor is more popular than the cluster center patch. Understandably, the father away neighbors make it into the final group less frequently.

We obtain consensus responses from the raw data in the following manner: if two or more users agree that 3 or more patches are visually coherent, that group of 3 or more patches forms the training set for a discriminative patch model. If a user selects a set of patches that intersect by 2 or fewer patches with their colleagues' responses, we consider that the user has identified an independent visual phenomena and use that response to train a new discriminative patch classifier.

Pick 5 (or more) Similar Images

Click submit when you are finished.
Drag or click images to select or deselect.

Please click at least 3 more images.

Image Gallery

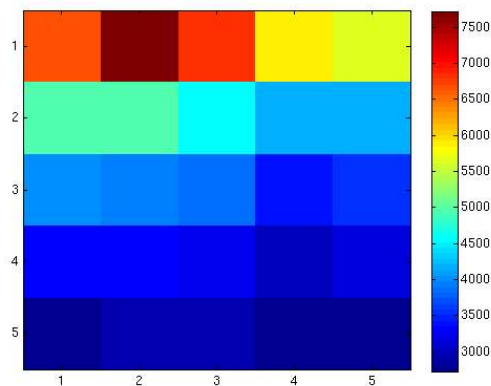


Figure 4.2: Popularity of images in the corresponding positions on the UI.

In the algorithm used for automatic patch discovery in [72], the models for the discriminative patches were trained using linear SVMs. Because the pipeline for building patch models from human responses was hugely faster, we trained the models using linear SVM and sigmoid SVMs.

Qualitative results for the most commonly firing patch detectors for a subset of the scene categories are show in Fig. 4.3. Subjectively speaking, it is difficult to differentiate which patch classifiers were made automatically and which were made with the help of the crowd.

After all of the discriminative patches for one of the 15 scene categories are identified, we also use a different method than [72] to select which discriminative patches to use in our feature library. Singh et al. suggest ranking the output discriminative patches by the posterior probability that a given patch will fire in the category from which it was discovered. We use this method to create the features used to train the scene classifiers. The performance of these classifiers is shown in Fig. 4.4. Instead of ranking the crowd discovered patches, we simply selected them at random from the available library of patches discovered from the crowd responses. No ranking was used for the crowd patches.

4.2 Scene Classification with Discriminative Patches

Figure 4.4 shows the performance of scene classification on the 15 scene dataset using automatically discovered and human-in-the-loop discriminative patch models. In the case of human-in-the-loop patches,

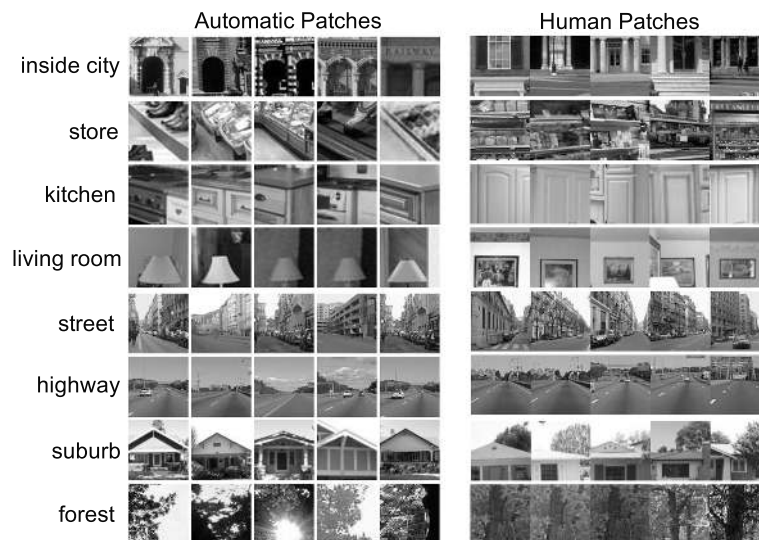


Figure 4.3: *Most confident detections by Discriminative Patches on a test set from the 15 scene dataset.* Each row shows the top 5 most confident detections for a discriminative patch discovered for the listed scene category.

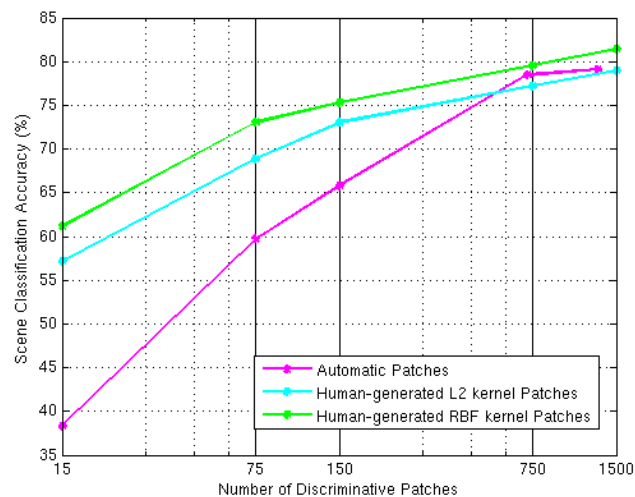


Figure 4.4: *Scene Classification Performance of Human and Automatic Patches on the 15 scene dataset.* The training set includes 100 images from each of the 15 categories, and the test set contains 80-90 images from each category. In this plot each trend line shows the performance of a different kind of discriminative patch. The best performing scene classifier (either χ^2 or L1 kernel SVMs) are shown for each different kind of patch. The automatically generated patch models are linear SVMs (see [72]). The human generated patch models are trained using either linear or RBF kernel SVMs.

we tried both linear and non-linear SVM patch classifiers. Each set of discriminative patch classifier generates a “bag of parts” histogram for every image which encodes how often a particular patch was

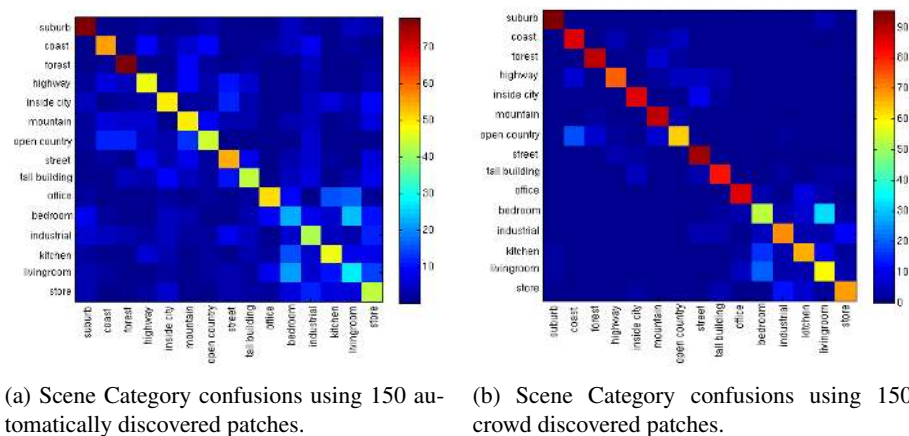


Figure 4.5: Comparison of Scene Category Confusions made by Automatic and Crowd Patch Classifiers

found. A second SVM is trained to classify images into scene categories based on these “bags of parts”. we set a detection threshold at a confidence of -1.0, as suggested in [72]. An equal number of patches are selected from each scene category. Fig. 4.4 shows that the patches discovered using human intervention perform better than the automatically discovered patches for scene classification on the 15 scene dataset. Overall, the sigmoid kernel crowd patch classifier achieves the best performance at 81.89% accuracy; the linear kernel crowd patch classifier achieves 79.11%, and the automatic patch classifier tops out at 71.82.

Scene Category Confusions

As in Sec. 2.7, it is important to compare the classification confusions made by the automatically discovered patches and the crowd discovered patches. Figure 4.5 shows the scene category confusions of the two methods side by side. This figure helps to explain why the automatically discovered patches were not as successful as the crowd patches. The patch classifiers discovered by the crowd lead to fewer inter-category confusions. It is interesting to note that the categories most often confused by the crowd patch classifiers are ‘open country’/‘coast’ and ‘living room’/‘bedroom’. The confusions made by these classifiers are confusions that humans ourselves could easily make when attempting to distinguish scene categories.

Classifier Training Costs

Interestingly, our human-in-the-loop patch discovery method is arguably cheaper to implement than the automatic method. In our experience, it took 300 CPU hours plus 1200 AMT Human Intelligence tasks

to discover patches for one scene category using our method. It took us roughly 6000 CPU hours to automatically discover discriminative patches for one category. The human-generated patches method is approximately 20x faster than the iterative cross-validation method in [72]. Using the default cost of an Amazon AWS instance (\$0.06 per instance per hour) and the cost of our AMT HITs (\$0.04 per HIT), it would cost \$66 to discover the human patches and \$360 to discover the automatic patches for one of the 15 scene categories. In light of the efficiency and accuracy shown by human patch discovery, we were inspired to further investigate the use of the crowd inside the loop of classifier training. The following sections of this chapter show another approach to employing the crowd as a crucial component of a detection pipeline.

4.3 Tropel: Crowdsourcing Detectors with Minimal Training



Figure 4.6: Top detections from a detector created with one seed example and trained using our Tropel system.

What kind of interaction would most appeal to the budding sartorialist or amateur ornithologist? In the example of Michelle the bird spotter from Sec. 1.3, Michelle wants to identify a bird species from one positive example. Besides her example image, Michelle and users like her also have a wealth of contextual knowledge and ability to train themselves to identify novel concepts. Our initial investigations in [58, 59] suggest that untrained members of the crowd can easily identify contextually related visual events with little filtering or quality control.

The simplest interaction for end users like Michelle would be to show the CV system their one novel example, and in return be presented with a set of images containing ‘similar’ objects. In traditional image retrieval, this is where the problem is solved and all interaction with the user ends. We contend that this is only the starting point for a deep and interesting collaboration between consumer and vision system that will provide a satisfying experience for the user. Figure 4.6 is an example of what a single user submitted input image would generate with our proposed pipeline.

In the first presentation of detected similar images, the end user may find (1) exactly what they were looking for among the most confident detections, (2) a small number of detections that are similar to what they want, or (3) results that are visually similar to their search item but not contextually similar in the way that they wanted. In this sense the initial detections are an approximation of what a set of

crowdsourced labels would look like for the user’s search object before a strict quality control system has been put in place to monitor the crowd annotators. At this point, the user could take from the detections the visual information they required, or they could alter their annotation/ classifier creation pipeline in order to produce results that are better for them. After several repetitions of this process, the end user will possess a set of annotations that are a gold-standard ground truth for their desired visual concept, and/ or a classifier capable of producing automatic annotations that are a sufficient approximation of ground truth from the perspective of the end user.

We live in a heavy-tailed visual world. There are visual concepts people encounter frequently (e.g. ‘car’, ‘bird’, ‘flower’) but a long tail of items which occur rarely (e.g. ‘Caspian Tern’, ‘Louis Vuitton handbag’, ‘abandoned steel mill’, ‘frosted graham cracker cookies’, ‘Sith Lord Halloween costume’, ‘Polynesian wedding ceremony’ etc.). For almost any object type (e.g. ‘guitar’) there are hundreds of fine-grained subtypes that are of interest to some people (e.g. ‘1943 Gibson J-45’, ‘1966 Fender telecaster’).

An ambition of the computer vision community is to be able to detect *anything* in images. This presents the challenge of trying to collect labeled training data for every conceivable event. There exist impressive efforts to try and exhaustively annotate everything, e.g. ImageNet [11] in the case of objects, or to learn from the huge amount of weak supervision available on the internet, e.g. NEIL [7] and LEVAN [15], or to empower domain experts to curate fine-grained databases, e.g. Visipedia [62].

An alternative (which we adopt) is to admit that we can’t anticipate the desires of all possible end users and instead only collect annotations as needed for a particular classifier. In this case *active learning* is the natural strategy to bypass the exhaustive annotation of a training dataset and instead iteratively label only those instances which are predicted to be most informative to the final classifier. However, this can still be tedious for moderately-sized datasets as a user repeatedly waits for the training and evaluation of each intermediate classifier before offering tens or hundreds of additional annotations.

In this chapter, we examine a scenario where an end user provides what is likely the minimum amount of supervision for non-trivial tasks – a single training example (e.g. one Puffin head to train a Puffin head detector). While we have seen rapid increases in recognition performance in recent years, a single training example is still grossly inadequate to train a reliable detector. On the other hand, the human visual system is surprisingly good at learning from a single example. Our Tropel (“noisy crowd” in Spanish) system exploits this gap. The single example provided by the end user is used to train *the crowd* who in turn provide hundreds of training examples to train a detector in an active learning setting.

For a typical detector, a dozen crowd workers will collaborate intermittently over several hours as a computer vision system mines informative training examples and posts HITs to Amazon Mechanical Turk to ask for training labels.

In everyday life, people can learn to recognize visual concepts from a verbal description or a single visual example. To the authors' knowledge, exploiting this common human ability in the crowd context has not been addressed in the literature. In this chapter we characterize the crowd's ability to train classifiers for rare visual phenomena given minimal training. We create over 200 detectors using the Tropel pipeline in several different visual domains.

With Tropel, we deliver a system that can create classifiers on demand, without a previously labeled dataset. This lowers the bar to entry for casual end users. Without specialized knowledge of how to design detectors for a particular visual domain (clothing, animals, architecture, etc.), users can employ Tropel to create detectors with minimum startup requirements. With Tropel, we seek to push the limits of what the minimum initialization can be for detector creation. We use the Tropel system to investigate how crowd active learning enables users to create useful detectors with the minimum effort or expertise in either computer vision or the visual domain of interest.

In order to accomplish the goals of the Tropel project, the problems before us are:

1. What is the fewest number of user-submitted training examples required to show the crowd what they should annotate?
2. How specialized a concept can the crowd be trained to understand given the high turn over of workers for this type of task?
3. How can the responses of workers be cheaply evaluated and combined?

4.4 Related Work

To the authors' knowledge, this chapter explores question 1 more directly and extensively than previous literature. Experiments with Tropel also explore questions 2 and 3 in a novel context.

Obtaining high quality image labels from crowdsourced annotators, who are often untrained and of unknown expertise, is an open research topic [27, 46, 82]. Active learning has been used to label objects that are *easy for non-experts recognize* such as pedestrians and PASCAL Visual Object Classes [1, 9, 70, 79]. Collins et al. [9] and Vijayanarasimhan and Grauman[79] both use active learning with

crowdsourced respondents as part of larger pipelines to create labeled datasets and detect the PASCAL challenge object categories.

However, both of these active learning systems need upwards of a hundred labeled training examples per category at initialization. The systems described in [1, 9, 79, 82] also require laborious attention to the crowd workforce. The workers themselves are modeled, measured, solicited or cast out based on analytics collected by researchers. The goal of [79] was to create detectors for the common objects of the PASCAL dataset ('bike', 'sheep', 'bottle', etc.). Crowd workers are able to identify this type of everyday item with little or no training. We explore the broader problem of how to use a crowd to train detectors capable of successfully identifying *anything* with the minimum amount of crowd training and quality control.

Crowd workers have been incorporated in the classifier creation process before. Deng et al. demonstrated a method for using the crowd to learn discriminative mid-level features for performing K-way classification on the CUB 200 dataset [13]. Their approach uses a gamified interface to ask workers to identify the distinguishing elements between two easily confused categories. Deng et al. showed how non-technical crowd workers could be used to make fine-grained distinctions. While we also demonstrate results on the CUB 200 dataset, we do not use the ground truth categorical labels at training time and we are not necessarily interested in K-way image categorization – we aim simply for one detector from one training example in any visual domain.

The Tropel pipeline employs a large number of workers to answer small, simple-for-humans questions. Crowd users simply click on images that contain the query object. We require less of the worker than other successful active learning methods that ask users to segment the relevant object or draw a bounding box, such as Vijayanarasimhan and Grauman [79]. However, we are asking the worker to concentrate on more specialized objects than the PASCAL VOC categories, as in [79].

Research into crowd micro-tasks has recently been described by [8, 18, 47]. Both Little et al. and Dow et al. show that minimal training helps crowd workers complete tasks more successfully than they would have been able to without training [18, 47].

In a rigorous investigation, Welinder et al. propose a method for accurately predicting the expertise and bias of a crowd worker [82] for visual tasks. Unfortunately, the method introduced in [82] assumes that a labeled validation set is available and it is possible to interact with an individual crowd worker over a number of questions. In the scenario addressed in this chapter, the training dataset is assumed to be unlabeled and we have no expectation of retaining a worker for long enough to confidently establish their

expertise. We investigate several approximations of the Welinder et al. method that improve detector accuracy over simple majority consensus among the workers. We concentrate on examining how to exploit the human ability to learn from a small number of examples to alleviate the lack of expertise in crowd workers.

We believe that Tropel is the first system to exploit the *combined efforts* of an end-user and the crowd via active learning. In the next section, we describe the details of the Tropel system and demonstrate its potential for easy parallelization and cost effective creation of large numbers of detectors.

4.5 Bootstrapping Classifiers

Our pipeline addresses a typical problem for end users. An end user, Jon, has access to a large set of images. Jon wants to search these images for a specific visual event, and begins his search from a positive example image. For example, he has a catalog picture of a jacket that he would like to buy. He wants to search a dataset of fashion images to find outfits where other people are wearing this type of jacket.

Tropel bootstraps detector training for user-specified objects of interest. The full input to Tropel is a single positive training example, an appropriate unlabeled database to learn from, and optionally a text label for the concept (e.g. ‘Puffin head’ or ‘stiletto heel’). Later experiments in this section vary the number of initialization images. Equipped with a set of test objects and corresponding example images, the operation of the Tropel pipeline is formalized in Algorithm 2.

Input: Dataset \mathcal{D} of unlabeled images, items to detect A
Output: Classifiers C for items A

- 1 $A \leftarrow$ item ▷ acquired through consultation with end user
- 2 $S_i \leftarrow$ seed example of A
- 3 $\mathcal{NN} = \text{NearestNeighbors}(S_i, \mathcal{D})$ ▷ 200 nearest neighbors of the seed example in \mathcal{D}
- 4 $N_0, P_0 = \text{crowdQuery}(\mathcal{NN})$ ▷ initial active query, crowd selects N_j negative and P_j positive images from the set of the seed example’s nearest neighbors
- 5 $C_0 = \text{svmTrain}(S_i \cup P_0, N_0)$
- 6 $\mathcal{D}_j = \mathcal{D} \setminus \{x : x \in P_0 \cup N_0\}$
- 7 $j = 1$ ▷ current iteration counter
- 8 **repeat**
- 9 9 $Q_j = \text{sortDetections}(C_j, \mathcal{D}_j)$ ▷ get top 200 detections by C_j in \mathcal{D}_j
- 10 10 $N_j, P_j = N_{(j-1)}, P_{(j-1)} \cup \text{crowdQuery}(Q_j)$ ▷ crowdsourced active query
- 11 11 $C_j = \text{svmTrain}(S_i \cup P_j, N_j)$
- 12 12 $\mathcal{D}_j = \mathcal{D}_j \setminus \{x : x \in P_j \cup N_j\}$
- 13 13 $j += 1$
- 14 **until** convergence()
- 15 **return** C, A

Algorithm 2: Crowd Active Learning for Fine-Grained Detectors

To begin the training process, each classifier is seeded with one user-submitted cropped image of the given object or part. Initially, our system only possesses the seed example and a set of millions of training patches that have no associated class labels. In the first stage of crowd training, we ask the crowd to annotate the first 200 nearest neighbor patches of the seed example (we do not train an initial classifier because we have no trustworthy negative examples). These 200 nearest neighbor image patches are the closest patches to the seed examples out of the millions of image patches extracted from all images in the training set. The nearest neighbors are found using L2 distance in the 200-dimensional DeCAF-derived [17] feature space used throughout the system. We make the assumption that the nearest neighbor patches are more likely to contain true positives or negative examples that closely resemble true positives than randomly sampled patches.

The response to this first active query provides the positive and negative training examples needed for the first iteration of the classifier. At each iteration of the active learning pipeline we train a linear SVM.

In the first iteration, where workers are reviewing nearest neighbor patches, and in later iterations, where workers are evaluating the top classifier detections, we only ask workers to look at a maximum of 200 image patches. We would prefer to show more images at a time to collect more training examples, but obviously this can lead to worker fatigue if we ask them to look at too many images.

Once the first classifier is trained, the next round of active learning can begin. The first iteration classifier evaluates all of the image patches in the original training set minus any patches that have already been annotated. Until the detector converges to the ideal detector, the top 200 most confident detections are likely to contain the strongest misclassifications. Thus, we ask the crowd workers to evaluate the top 200 detections on the training set. As the iterative retraining approaches convergence, the top detections are more likely to be correct, but in the initial stages these top detections will provide hard negatives that rapidly shrink the version space. To easily compare detector improvement, we set a hard limit for the convergence criteria — 5 active learning iterations. The most confident detections for each iteration’s classifier are presented to the crowd using the user interface shown in Fig. 4.7. The crowd we query for the following experiments is from Amazon Mechanical Turk (AMT).

Tropel is meant to be classifier agnostic. We use linear SVMs because they are lightweight to store and train. The choice of feature or image representation likely has a bigger impact than the final classifier.

At each iteration of the active learning process, untrained crowd workers view a set of “Candidate Images”. These are the top 200 detections from the current active learning classifier. The patches shown

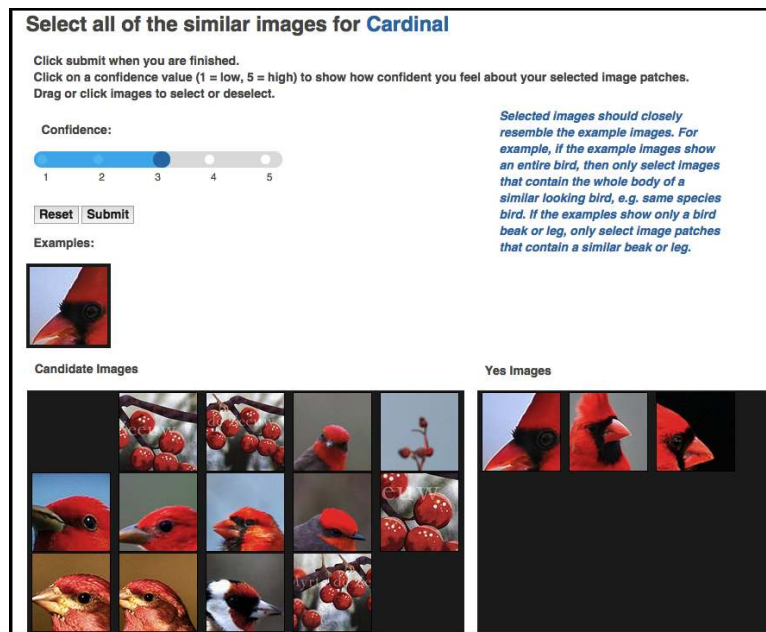


Figure 4.7: *Active Query User Interface*. The UI shown in this figure implements the function `crowdQuery()` in Alg. 2. Users click on the image patches they believe are visually and semantically similar to the example images. The column titled “Candidate Images” is scrollable so that users can view all 200 query images.

are pruned so that none of the patches have an overlap of greater than 0.3. The overlap ratio is defined by the area of the intersection divided by the union of the two patches. The workers select all of the patches that they believe match the seed example(s).

The interface in Fig. 4.7 also asks users to self-report their confidence in the accuracy of the selected images patches. The informativeness of this worker-supplied meta-data will be addressed in later sections.

Three workers independently answer each active learning query. If 2 out of 3 workers agree that a cropped image patch is a positive or negative case, that crop is added to the positive or negative training set for the next iteration’s classifier. Crops that are not agreed upon are returned to the set of queryable images.

In the Performance Comparison section we compare the average precision of detectors created by a crowd given a range of visual training examples. In the Worker Consensus Protocol Comparison section we examine the effect of weighting the votes from different workers by several metrics. We also compare the accuracy of individual workers to their time per hit, number of hits completed, and self-reported confidence.

4.6 Experimental Evaluation - CUB dataset

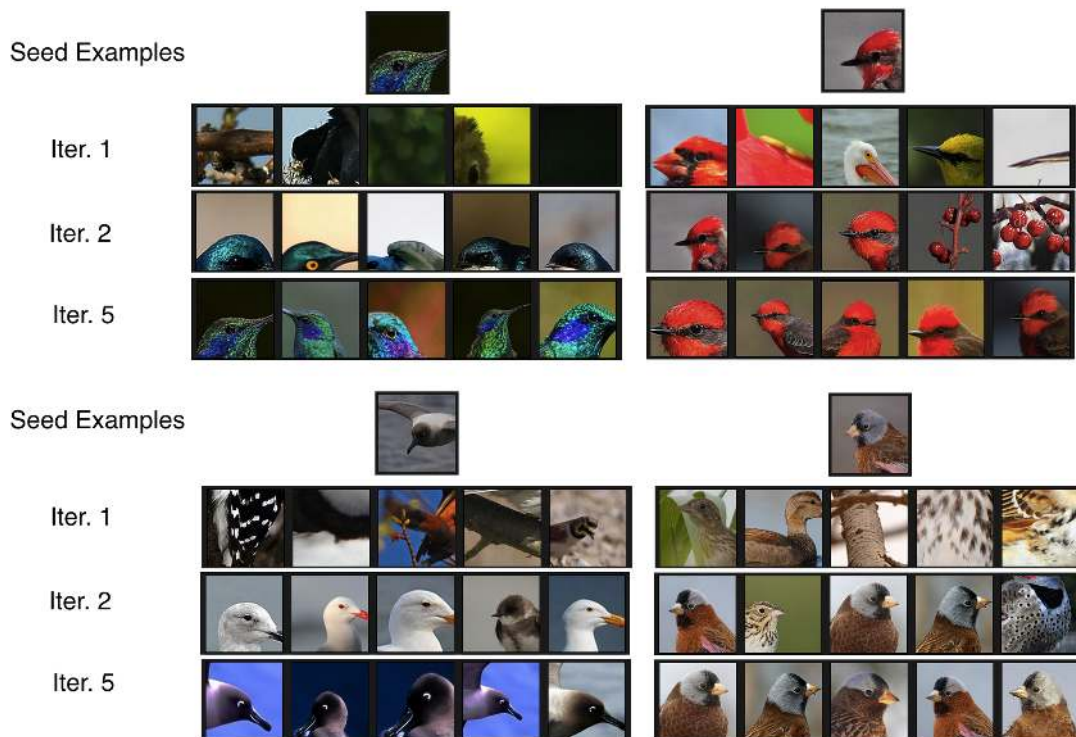


Figure 4.8: Example detections at different iterations of the Tropel pipeline. This figure shows the top 5 detections of 4 bird head classifiers. The species from left to right are: *Green Violetear*, *Vermillion Flycatcher*, *Sooty Albatross*, and *Gray-crowned Rosy Finch*. The first row in each block of images shows the seed examples used to start the pipeline. The following rows are the top 5 most confident detections on the training set. The different rows show the top 5 test detections from the first, second and fifth rounds of active learning.

For the purposes of investigating our pipeline, we used the head part annotations that accompany the CUB 200 dataset [80]. All of the images in this dataset have a head location annotated by a crowd worker. The larger part of these annotations were verified by experts. We obtained our set of example seeds by cropping 75×75 pixel head patches centered on those locations. For the fashion item tests a member of our team manually cropped 10 example patches for 10 different items of clothing and accessories.

We use a coarse-to-fine sliding window classifier. Tropel poses queries over the set of all sliding windows in the training set. The minimum window size is 75×75 pixels. Image patches are cropped at 4 different scales, up to 200×200 pixels. Typical image resolution is 500 pixels per side. The set of sliding window patches constitutes the training set for our active learning system.

We represent each patch with CNN features calculated using the pretrained DeCAF network from [17]. This network was trained on ImageNet which contains images of both birds and clothes. None of

the ImageNet images are in our experiment datasets, which prevents pollution of the test images with images that were used to train the CNN. We reduce the dimensionality of the DeCAF feature by using only the first 200 activations from the last fully connected layer as suggested by Razavian et al. [63] when using a pretrained network on a novel dataset. This dimensionality subsampling is especially helpful for a detection task where features are computed and stored for millions of sliding windows.

We first evaluate the ability of our system to build detectors for the heads of 200 different bird species. We compare the performance of our crowd active learning pipeline with a baseline detector.

While our pipeline is designed to train detectors from unlabelled datasets, we use the CUB 200 [80] dataset to quantify the accuracy of the learned detectors. We train classifiers for the heads of all 200 types of birds in the CUB 200 dataset, which contains 11,788 images split nearly evenly between training and test sets. The CUB 200 database contains ground truth part locations for the head of each bird. The seed examples for each bird head classifier were randomly selected from these labeled parts. Figure 4.8 shows the output of several bird head detectors at different stages of the Tropel pipeline, illustrating improving detector performance as the number of training iterations increase.

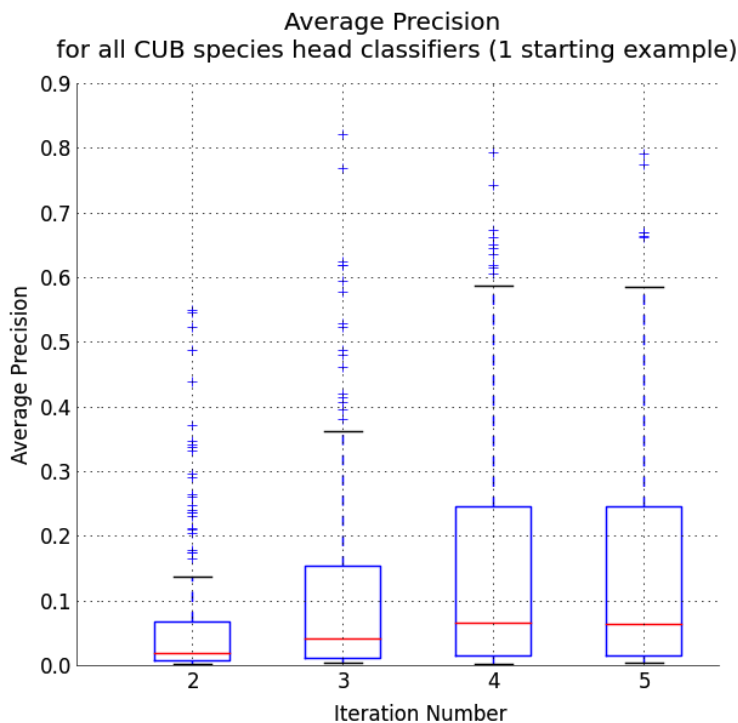


Figure 4.9: *Detector Average Precision at different iterations of the pipeline.* Each species has approximately 28 true positives in a dataset of millions of image patches. This plot gives an impression of the improving precision and recall with increasing iterations. Average Precision, the average value of the precision over the range of recall from recall = 0.0 to recall = 1.0 for each detector, is used for brevity.

The box plot in Fig. 4.9 shows the iterative improvement of all 200 bird head detectors. Iterations 2-5 are shown as no classifier is trained during iteration 1. The precision of each detector is calculated by counting true positives in the 30 most confident detections in the CUB test set. Note that there are at most 30 positive bird head bounding boxes for any species in the test set. Detections that overlapped by a ratio of greater than 0.3 were removed from the set of most confident detections to eliminate multiple detections of the same bird.

Some classifiers drastically increase in accuracy, most improve more modestly, and the bottom 25% of detectors hardly improve. The detectors that fail to improve are often hindered because those birds bear striking visual similarity to related species in the species taxonomy, e.g. many type of Sparrows look strongly similar. This drift to identifying related birds, and thus similar visual phenomena, is discussed in the Hierarchical Similarity section.

Performance Comparison

First, we compare Tropel to a typical computer vision approach – detectors trained only with trustworthy annotations and no active learning. The baseline classifier is a linear SVM trained with all the positive head patch examples in the training set and 10,000 randomly selected negatives. While we refer to these as ‘baseline’ classifiers, they could be expected to perform better than our pipeline because they train on crowd annotations that have been cleaned up and verified by experts. All detectors are evaluated by their detection average precision score on the CUB test set. We calculate detection AP using the PASCAL VOC standard, with the alteration that detections may be counted as true positives if they have an overlap ratio of 0.3 instead of the PASCAL threshold of 0.5 [21].

The detection AP scores for the Tropel detectors are plotted against the performance of the baseline detectors in Fig. 4.12. We also show the performance of crowd classifiers when the workers are given slightly more training – 5 examples instead of 1. As expected, the AP scores for the detectors with 5 seed examples more closely approach the scores of the ground truth trained detectors. Overall, our crowd active learning process seeded with one or five training examples approaches the performance of the traditional detection pipeline. In Fig. 4.12 the average precision scores may appear low, however it is important to notice the difficulty of this baseline as the linear SVM trained on the fully annotated training set also has relatively low AP scores.

The points plotted in Fig. 4.12 were obtained as follows: 1) a detector for each of the 200 bird species was trained using each of the 4 methods, 2) we calculated the average precision for all 800 detectors, 3)

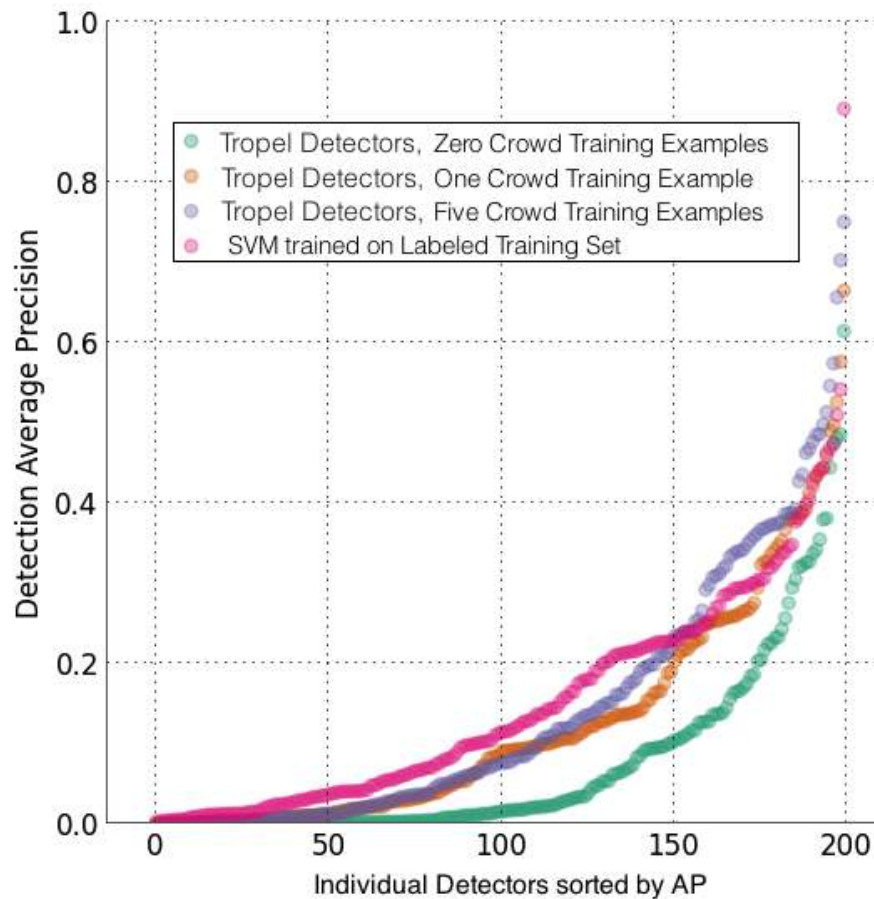


Figure 4.10: *Tropel Detector Average Precision for 200 Bird Species*. This plot shows all of the 200 head detectors sorted by score for each method. Overall, this graph indicates that our methods approach the performance of ground truth, and more training examples for the crowd leads to higher AP.

for each training pipeline, we took the 200 associated detectors and plotted their AP scores sorted from worst-to-best performance. For each of the trend lines, the ordering of the plotted detectors is different. We decided to plot each trend line in strictly ascending order to give the general impression for each training pipeline of how many classifiers performed badly, average, and well.

Our overall observation is that bird detectors that succeeded had one or more of these traits: 1) the bird had strong lines and sharp gradients in the head pattern, 2) the original example for the Turkers was an iconic-looking, well framed image, 3) the bird had no relatives that looked nearly identical.

Figure 4.12 shows that seeding the active learning process with five examples works slightly better than using a single example. There are two reasons for this – the candidate image patches shown to the workers in the first iteration are more likely to have positive examples and the human workers themselves are better able to pick out the positive examples because they have more training examples. That said, the

difference between one and five examples ends up being fairly small (and critically they both perform nearly as well as detectors trained on exhaustive ground truth annotations). This raises the question of whether we could use even less supervision than a single example. Is it possible to train the crowd with no visual examples?

In a zero example training scenario there is no visual training at all. In our case that could mean a text label instead of a seed patch. Omitting the seed patch entirely is not viable in our current pipeline because we would have no reasonable initialization for the active learning. We could resort to showing random image patches, but for any given concept (e.g. ‘cardinal head’) there would likely be no positive examples for the crowd users to select and thus it is unlikely they could improve the detector even if they happen to know the concept.

Still, it is interesting to know what happens if we use the seed example to initialize the active learning but hide the seed example from the crowd workers. This gives us some indication for how much ‘training’ the humans are getting from a single visual example. For this experiment, we initialize our pipeline identically to the one-example detectors – one initial seed patch was used to find nearest neighbor patches for the first active query. However, the crowd never sees the seed patch. The crowd workers are shown a text description of what to look for, i.e. the species name. The UI for this zero crowd training experiment is the UI from Fig. 4.7 with the example image removed. An example of this UI is shown in Fig. 4.11.

The zero example detectors perform strictly worse than the one example version with the same initialization, verifying that it was important to ‘train’ the crowd workers instead of expecting them to know the concept already (See Fig. 4.12). Two notable outliers among the zero shot detectors are the detectors for the *Green Jay* and *Horned Puffin*. The initialization for these birds was unusually good (because they are distinctive birds) so it is likely that the crowd ‘learned’ what these birds looked like just by noticing what type of bird dominated the candidate patches. The Green Jay is a particular easy bird to guess – no other bird in the dataset has a similar distinctive neon green color, and so the workers who simply clicked on all green birds would have been making accurate selections.

It is important to note that in many low performing cases the appearance of the bird species often varies greatly for males, females, mating birds, etc. For example the *Black Tern* changes head color from black to white with a black eye-patch when the bird enters the mating stage. We could not train workers for this wide variety of appearance in the one example training setting. The examples for the 5 example detectors were selected at random from the ground truth head patches, and may not have captured all

Select all of the images containing the head of a Pomarine Jaeger

Please click on images that are the just the * head * of the same species of bird written in blue above.

Click submit when you are finished.

Click on a confidence value (1 = low, 5 = high) to show how confident you feel about your selected image patches.
Drag or click images to select or deselect.

Confidence:

1 2 3 4 5

Reset Submit

Candidate Images

Yes Images



Figure 4.11: *User Interface with Zero-Shot learning instructions.* This is the interface seen by workers during the zero-shot training type of active query tasks. In order to give the worker some indication of what to look for, we specify the species name and part of the bird that we are looking for. No visual training examples are shown to the worker.

appearance variations.

In some cases in Fig. 4.12 the Tropel detectors seem to be out-performing the baseline (although usually not by a large margin). To show this more clearly, Fig. 4.12 shows the difference between two types of Tropel detectors and the baseline detector. In this plot, the difference between the AP score of the Tropel detector and the baseline detector are plotted. The vertically aligned points are both for the same bird species. This graph shows that about one third of the 5 example classifiers and one sixth of the 1 example classifiers in fact have better performance than the baseline. But on average, the baseline with ground truth training data has an AP of .157 compared to .106 for Tropel with one example.

As an additional experiment, we changed the user training examples to include negative examples. For each of the 200 bird species, we showed the crowd workers one positive example and 3 negative examples. The negative examples were selected from other species in the same family, so as to best show the subtle differences that distinguish one species from its cousin species. Overall this approach had little to no effect on the average AP score across categories. Table 4.1 shows 5 detectors where the negative examples helped workers and 5 where the negative examples were unsuccessful. In the cases where the negative examples didn't improve performance the workers may have been confused

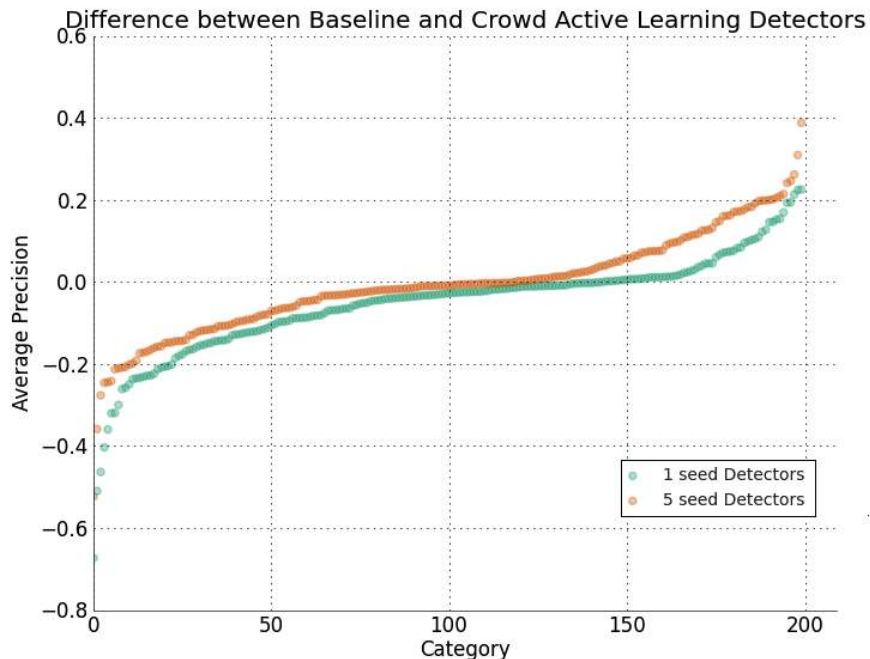


Figure 4.12: *Difference in Detector Average Precision*. Y-axis is the difference in AP score between the Tropel detector and the baseline SVM detector. The x-axis locations each represent a unique bird species. The classifiers were sorted from worst-compared-to-baseline to best-compared-to-baseline.

by the fact that the incorrect species birds were visually similar to the correct species. Generally these unsuccessful cases had very few new positive examples selected by workers.

Comparison to Active Learning Baseline

As an alternative baseline, we also compare Tropel to a canonical active learning pipeline. Our “Active Learning with an Oracle” baseline uses the same active learning process as Tropel, but instead of crowd responses uses an oracle to respond to the active queries. The oracle responses are obtained using the ground truth labels provided with the CUB dataset.

Table 4.2 compares the active learning baseline, the linear SVM with all available labels baseline, the Tropel detectors, and an active learning baseline where responses were random. From Table 4.2, we can see that the crowd creates detectors that are on average slightly better than the oracle baseline. This is likely due to human workers selecting rotated, occluded or otherwise varied examples of the initial visual concept that are missed by only relying on the pre-existing oracle responses. Overall, the detectors trained with all of the labeled data do significantly better. This is due to the fact that many bird species have wide appearance variety that is difficult to capture with only one example. The next section further

Table 4.1: *Crowd Training with Negative Examples*. This table compares detector average precision (AP) two types of crowd worker training. In the left results column, the detectors were trained by workers who were shown both a positive example of the species and 3 incorrect examples from birds of cousin-species. In the right column, workers were only shown one positive example.

Bird Species	Average Precision	
	With Negative Crowd Training	Only Positive Crowd Training
Groove billed Ani	0.12	0.00
Eastern Towhee	0.10	0.01
Least Auklet	0.17	0.06
Purple Finch	0.21	0.12
Painted Bunting	0.58	0.50
Red faced Cormorant	0.00	0.25
Rhinoceros Auklet	0.02	0.13
Sooty Albatross	0.03	0.25
Brewer Blackbird	0.00	0.16
Brown Creeper	0.00	0.26

Table 4.2: *Comparison to Active Learning Baseline*. Averages are taken over 40 detectors for randomly selected bird species.

Detector Type	Avg. AP Score
Active Learning with Oracle	0.095
Tropel with One Example	0.106
SVM trained on Labeled Dataset	0.157
Active Learning with Random Responses	1.2e-4
Chance	3.9e-4

explores this apparent shortcoming of Tropel.

Hierarchical Similarity

One cause of detector failure on the CUB data set is classifier drift. As observed by Welinder et al. [82], a non-expert crowd may have an incorrect preconceived notion of an attribute or object. In particular, Welinder et al. [82] showed that AMT workers confused grebes for ducks. Figure 4.14 shows that while a similar error is occurring with our detectors, the detection errors occur in a reasonable way. The top detections of ‘Song Sparrow’ and ‘Caspian Tern’ shown in Fig. 4.14 include the heads of other birds in the Sparrow and Tern families. The top detections for these two classifiers are failing in a predictable way given the taxonomy of this problem.

Table 4.3 characterizes the detector drift across all bird head detectors. In the CUB 200 dataset there are 35 families of birds, such as *Sparrows*, *Terns*, *Wrens*, *Cuckoos*, etc. All species have family membership. Table 4.3 shows that for several families, the precision of the detector in the 30 most confident test detections for the family is much higher than the precision for a particular species’ head

Table 4.3: *Hierarchical Similarity of Top Detections*. The precision scores listed above show the ability of some seemingly low performance detectors to do well at recognizing visually similar bird heads. Precision was calculated as the number of true positives among the 30 most confident detections on the CUB test set. Please note that for each species of bird there are approximately 30 true positive examples in the test set. For bird families such as the *Gulls*, *Sparrows*, *Terns*, and *Warblers*, the precision of the individual species detectors are low because those detectors are picking up other, similar looking members of the families. This is shown by the higher family precision scores. This table demonstrates that when our detectors fail, they do so by drifting to detecting visually and semantically similar phenomena.

Bird Species	Precision in Top 30 Detections	
	Species	Family
Cardinal	0.93	0.93
Green Violetear	0.80	0.87
Horned Puffin	0.77	0.80
European Goldfinch	0.70	0.73
American Goldfinch	0.67	0.67
Cape Glossy Starling	0.60	1.00
Scarlet Tanager	0.60	0.93
Purple Finch	0.57	0.80
Song Sparrow	0.30	0.60
Le Conte Sparrow	0.17	0.80
Tree Sparrow	0.17	0.63
Clay colored Sparrow	0.17	0.63
Field Sparrow	0.27	0.53
House Sparrow	0.13	0.23
White throated Sparrow	0.03	0.77
Lincoln Sparrow	0.03	0.90
Western Gull	0.40	1.00
Ivory Gull	0.43	0.80
Herring Gull	0.17	1.00
California Gull	0.10	1.00
Ring billed Gull	0.10	0.63
Heermann Gull	0.27	0.27
Glaucous winged Gull	0.10	0.83
Slaty backed Gull	0.03	0.60
Artic Tern	0.53	0.90
Elegant Tern	0.13	0.97
Caspian Tern	0.20	0.97
Forsters Tern	0.13	0.97
Least Tern	0.07	0.43
Common Tern	0.10	0.33
Prairie Warbler	0.47	0.90
Bay breasted Warbler	0.33	0.33
Prothonotary Warbler	0.43	0.97
Black and white Warbler	0.33	0.37
Golden winged Warbler	0.23	0.63
Canada Warbler	0.10	0.70
Kentucky Warbler	0.50	0.93
Yellow Warbler	0.27	0.80
Pine Warbler	0.23	0.87
Cape May Warbler	0.03	0.77
Black throated Blue Warbler	0.03	0.13
Mourning Warbler	0.07	0.30



Figure 4.13: *Classifier Drift*. The two groups of images above show the seed examples and 20 most confident detections of the classifiers trained for the heads of the ‘Song Sparrow’ and ‘Caspian Tern’. These classifiers detect a coherent visual event, but that event is not the same as what is illustrated by the seed examples. The different sparrows have slightly different feathering patterns and the terns have different beak coloration for example.

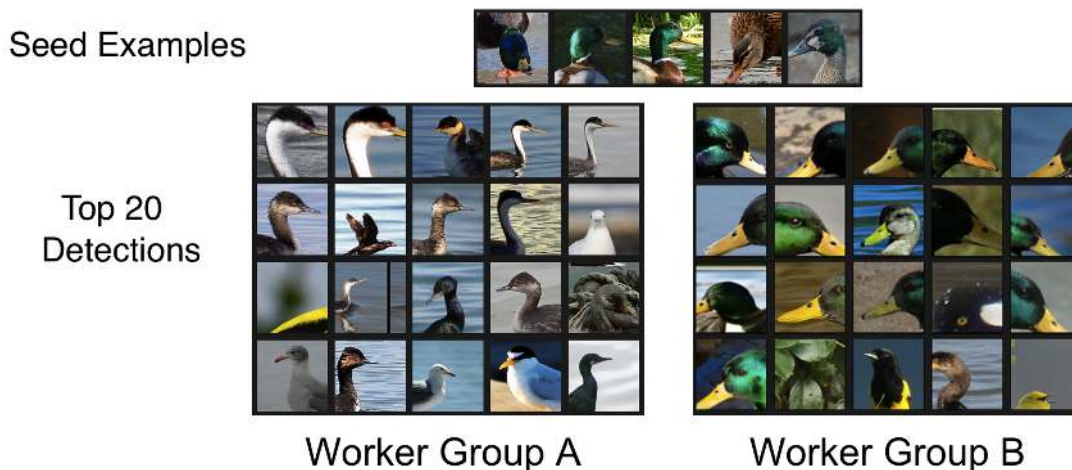


Figure 4.14: *Classifier Drift Comparison*. In this figure, the output from two detectors seeded with the same 5 examples are shown. The output of these two detectors are quite different. The workers in Group A created a drifting classifier that recognizes many different kinds of water birds. Group B created a classifier that successfully identifies Mallards. This may be explained by the fact that Mallards are not an indigenous species to India, and a significant number of our workers are located in India.

detector. The detectors are generalizing to a family of bird heads in the same way we would expect a non-expert human to do. For these particularly fine-grained recognition tasks it is likely that some domain specific instruction would need to be provided to the crowd (e.g. which key features distinguish the Sparrow of interest). The results shown in Table 4.3 are encouraging for applications such as image retrieval where it is important that the most confident results are all or mostly correct.

Worker Consensus Protocol Comparison

The Tropel system is intended for use on entirely unlabeled datasets. Thus in a typical use case we do have ground truth that is available for the CUB dataset (which we have used to evaluate the detectors).

When creating detectors with the CUB dataset, we are able to use the gold standard positives to score the accuracy of our crowd workers. Our embedded gold standard tests include 5 ground truth positives, all ground truth head patches for the given species, and 5 negatives, which were randomly selected negative patches from the CUB dataset. The test image patches are randomly interspersed with the active query images on every human intelligence task (HIT).

We found that assessing worker recall (true positives/(true positives + false negatives)) is a more informative metric for worker behavior than precision. Selecting only correct examples was easy for workers but finding all of the positive catch trails was hard. The catch trial recall was an average of 0.318, and [0.264, 0.338, 0.395, 0.517] across the [25%, 50%, 75%, and 100%] quartiles. For comparison, the recall of the author of this dissertation over 25 HITs was 0.462.

Some workers are good, some mediocre, and some apathetic in that they never identify any positive training examples. It would be helpful if we could even approximately determine which workers to preferentially trust. Tropel seeks to avoid the added burden of collecting extra data to generate catch trials. Successful existing alternatives, such as Welinder et al. and Ipeirotis et al., require an extended interaction with a crowd worker to fit parameters to accurately model the workers' expertise and biases [32, 82]. The simple worker consensus approaches investigated in this chapter will not lead to dramatic improvements. However, in keeping with the low-overhead goals of Tropel, we restrict ourselves to work with the relatively weak metadata collected through our UI without any additional instrumentation or data collection.

We examine 3 simple mechanisms which all perform a better than simple consensus. Our workers are ephemeral, and we typically know limited data about them. Across all experiments in this chapter so far, the average number of unique workers that participated in creating each detector was 11, out of a possible 12 (4 active queries, with 3 respondents each).

The data easily available to us includes the amount of time a worker spent answering an active query, how many times that worker has worked for us before, and a self-reported confidence value. In the UI shown in Fig. 4.7, there is a slider bar at the top of the page. Workers are asked to identify their level of confidence about their responses by selecting a value between 1 (low) - 5 (high).

To examine how to estimate a worker's reliability using only these metrics, we created detectors for

20 randomly selected bird types. For these experiments we ask 9 workers (instead of 3) to respond to each active query. The workers votes are weighted by each of the three factors or not weighted at all (the average consensus method). For a patch to be counted as having a true or false label, the margin between the weighted true or false votes has to be greater than 1/3 of the total of the weighted votes. Patches that land inside that margin are discarded.

Averaged over the 20 different classifiers, considering the worker’s history with our system was the most reliable. Considering how long a worker spent on a task and weighting by the worker’s self-reported confidence both outperform average consensus. Of the three metrics we used, the self-reported confidence had the strongest correlation with the workers’ own recall, although the correlation coefficient was still a relatively weak 0.19.

Table 4.4: *Weighted Response Strategies*. The averaged AP scores for the four response weighting methods. Averages are taken over 20 randomly selected bird head detectors.

Weighting Strategy	Average AP Score
Worker Self-Reported Confidence	0.0898
Time Spent Answering Query	0.0874
Number of Previous Tasks Completed	0.1105
Average Consensus	0.0844

Across all experiments, 1781 workers participated. Workers were from 33 countries, although 1031 were from the United States and 338 from India. On average, workers were paid approximately \$2 per hour.

Cost Comparison

A primary motivation of our pipeline is that it requires relatively few (expensive) trustworthy annotations compared to traditional classifier construction. For example, the CUB dataset has full ground truth annotations and we estimate the cost of training a classifier for a single item on the CUB training set as approximately \$60. We estimate the cost of labeling the CUB training set for one part as the number of training images (5994), divided by the number of images shown in each labeling task (25), multiplied by the number of repeat annotators (5) and the fee for each task (\$0.05) [80]. Our estimate does not include the cost of validating annotations and correcting errors in the CUB dataset, tasks which were completed by experts in ornithology and trained volunteers.

The cost of training a detector with our system is \$0.60, which is the number of rounds of active queries (4), multiplied by the cost of each query (\$0.05) and the number of repeat annotators (3). Note that the cost of training a classifier with our system does not depend on the number of images in the

training set, although a larger training set will increase computational training time. Tropel is more computationally expensive than the baseline method, taking an average of 260 cpu hours to train one detector. Training a single linear SVM on the CUB training set takes approximately 10 mins. on comparable hardware. The calculation steps in Tropel are embarrassingly parallelizable however.

4.7 Experimental Evaluation - Fashion dataset

Using our crowd-powered Tropel system, we are able to train detectors on a fresh, unlabeled dataset. To test our pipeline with unlabeled data, we obtain a set of unlabeled images from theSartorialist.com, which is a website devoted to images of contemporary fashion worn on the street in New York, London, Milan, Tokyo and Paris. We selected this set of images because they are all taken by a single professional photographer, thus their quality is high. The environment and pose of the subjects are complex and unpredictable. All of the images are taken of people walking the streets of the aforementioned cities. Most pictures contain busy backgrounds and multiple subjects. People are sitting, standing, walking, and interacting with each other without necessarily paying any attention to the photographer. This gives us a challenging dataset with which to test our pipeline. This dataset contains 4785 images. A self-described fashion expert on our team collected the example images for the fashion items using Google image search.

Figure 4.15 shows the top detections of 10 fashion detectors trained using 5 iterations of our system. In this case study, the dataset was not divided into test and train. Our goal was to create detectors using all available data. This case study is similar to the image retrieval problem in that respect.

As the crowd workers selected positive training examples, those image patches and any patches with an intersection over union (IoU) with the positive patch of 0.5 were removed from the pool of unlabeled images that could be used in the next active learning iteration. Some image patches that are zoomed out or shifted versions of the worker selected patches are left in the pool of possible training examples. Our goal was to make it possible to get a diverse set of patches that show a given positive example at slightly different perspectives.

Figure 4.15 shows the incremental improvement of several fashion classifiers trained using our system. As fashion concepts are more commonly know than fine-grained bird attributes, we observe classifier drift in a larger percentage of the fashion classifiers. In Fig. 4.15, the leggings classifier also returns skinny jeans, the notch collar classifier returns a peak collar, and the boots classifier returns other types of shoes in the top 5 detections.

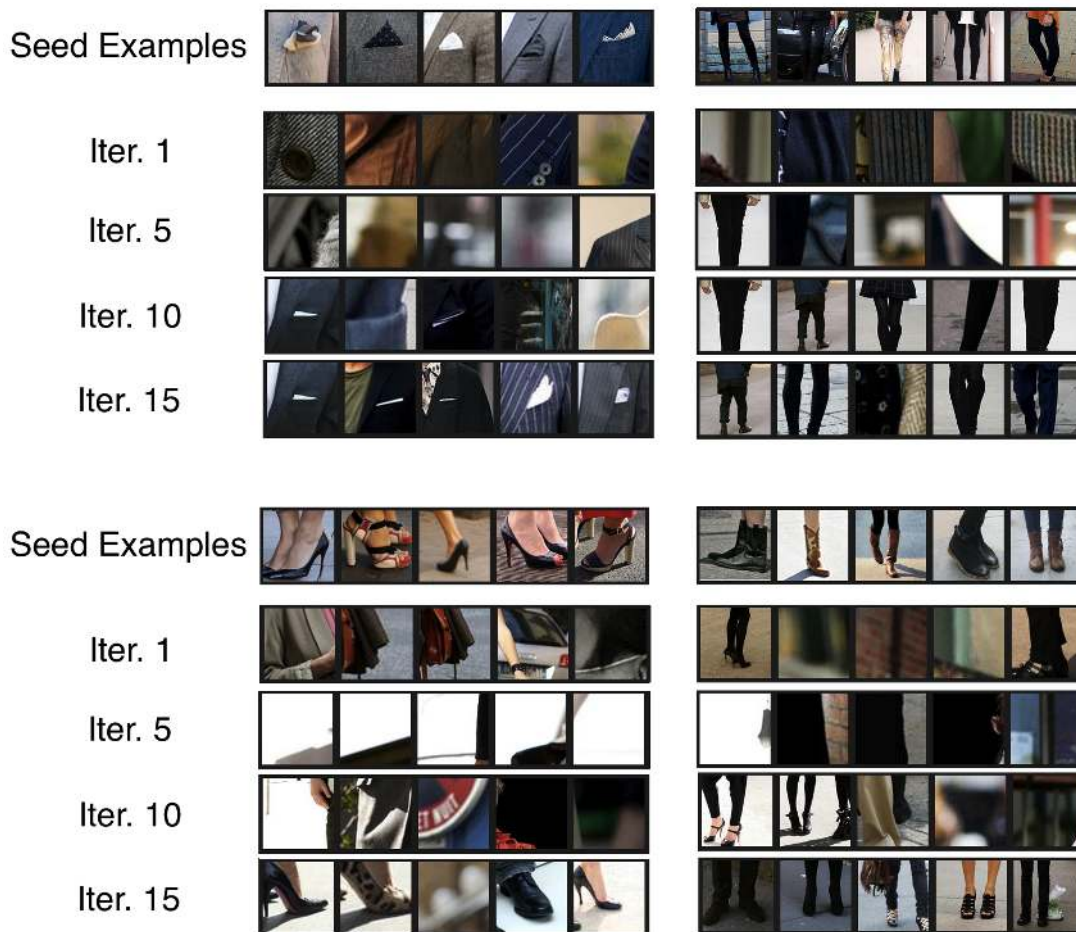


Figure 4.15: *Example detections at different iterations of the pipeline.* This figure shows the top 5 detections of our fashion dataset. Incremental results for the following classifiers are shown: ‘pocket square’, ‘leggings and opaque stockings’, ‘high heel’, and ‘boots and booties’. The first row in each block of images shows the seed examples used to start the pipeline. The following rows are the top 5 most confident detections on the held out test set. The rows show the top 5 test detections from the first, fifth, tenth and final rounds of active training.

Figure 4.16 shows that the most confident detections are quite similar to the worker selected positives. Where the detectors fail, it is because the dataset didn’t contain many examples of the desired item, for example *women’s shorts* or *epaulet*.

Fig. 4.17a shows a detector that is not particularly successful. We can see that while the most confident detections for the ‘glasses’ detector does find a number of glasses, many of confident detections are wrong. The training set for this detector, which contains hundreds of image patches containing glasses, none the less has many images that are not well aligned and at different magnification levels. This contributes to the glasses classifier performing less well. Basically, the humans did their job well here and

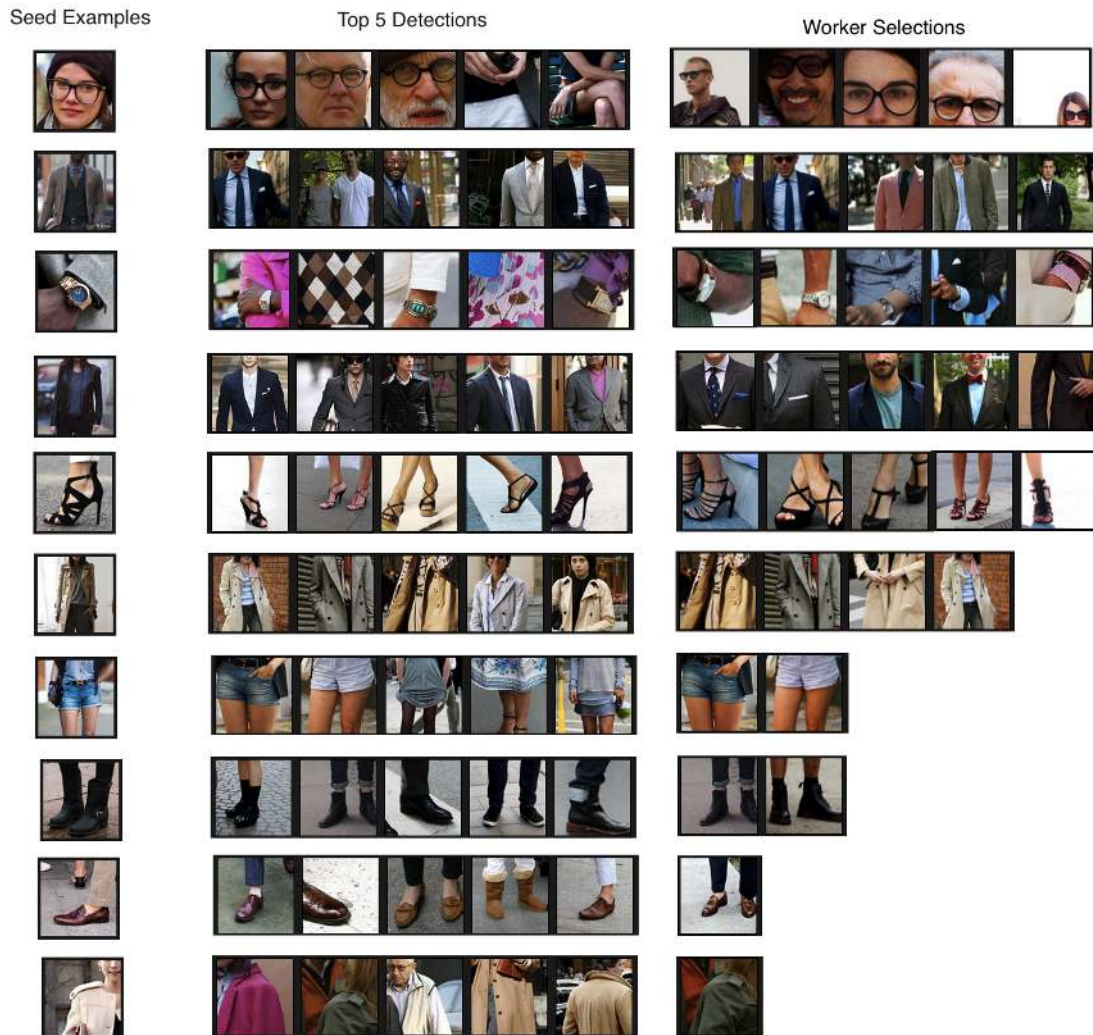


Figure 4.16: *Example detections from 5 fashion concepts.* This figure shows the 5 most confident detections of on our fashion dataset. Results for the following classifiers are shown: *glasses, men's blazer, watch, jacket, strappy heels, trench coat, women's shorts, boots, men's loafers, and epaulet.* On the far right, 5 randomly sampled worker selected positive patches are shown. If the workers selected fewer than 5 positive training examples, all of the positives are shown. The results of the watch and epaulet detectors are especially interesting as those items are physically small, thus making them a much harder detection challenge.

the computer vision and learning techniques did not. However, Fig. 4.17b shows a detector that performs very well. In this example the worker selections are all similarly aligned and cropped. This results in a detector that has accurate and well-aligned top detections.

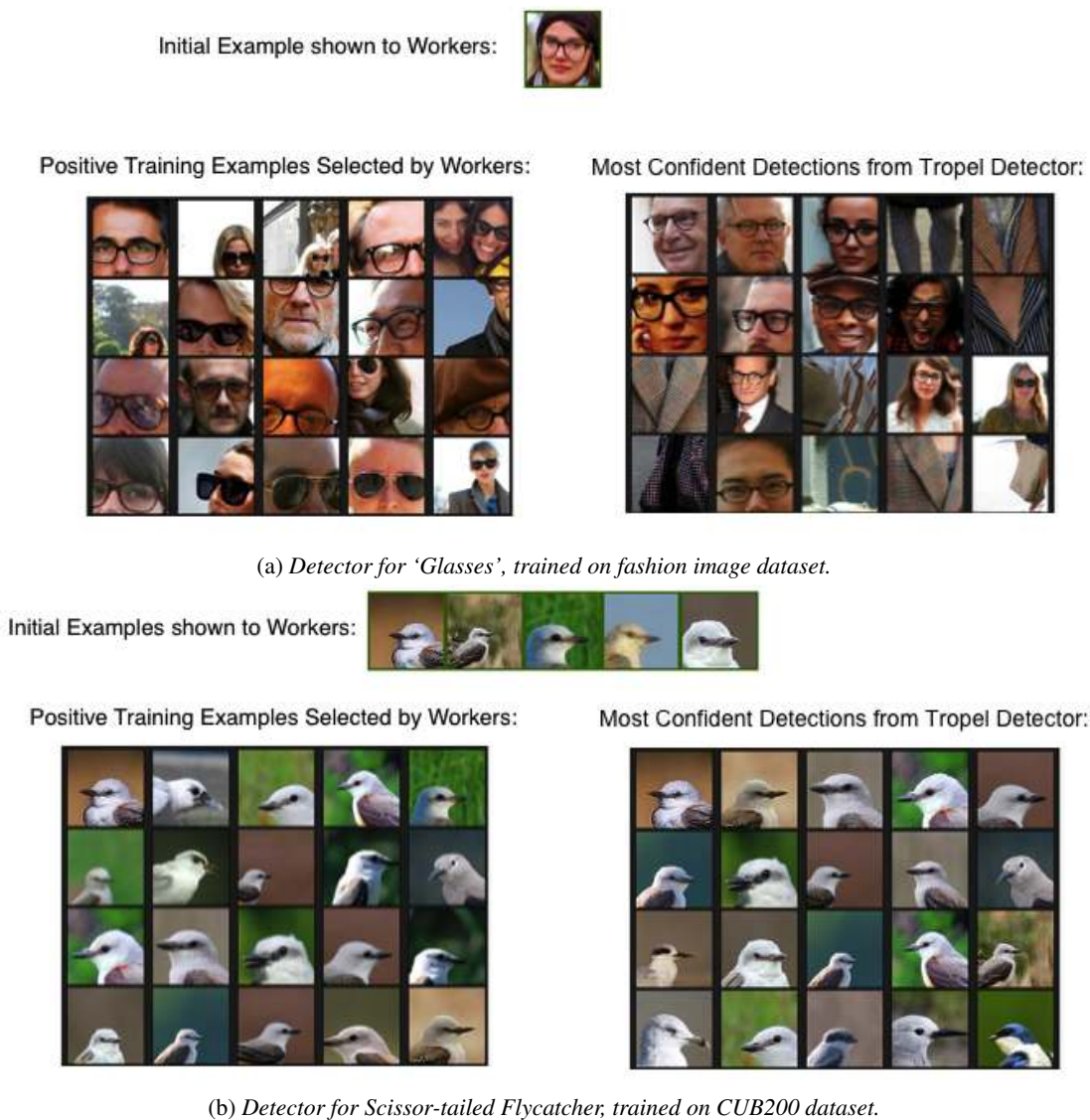


Figure 4.17: Comparison of Worker selected training examples and detector output.

Runway Fashion

For a comparison of the visual complexity of different attributes, we train our fashion classifiers on another set of images. our second fashion dataset is 20,000 images, split evenly into test and train sets, obtained from the New York Times repository of runway images from the major Fashion Weeks of 2012-2014. The set of images from theSartorialist.com had a large percentage of menswear, and the images were taken outside in cities. The NYT runway dataset contains proportionately more womenswear, and the images have simpler backgrounds. Fig. 4.18 shows two cases where classifiers trained on these two fashion datasets had disparate success. Fig. 4.18 indicates that our method, like any supervised training

pipeline, is subject to the size and bias of the training set.

Note that for the classifiers illustrated in Fig. 4.18, the classifiers are trained and tested separately on each dataset. The classifiers trained on the street style images are not tested on the runway images or vice versa.

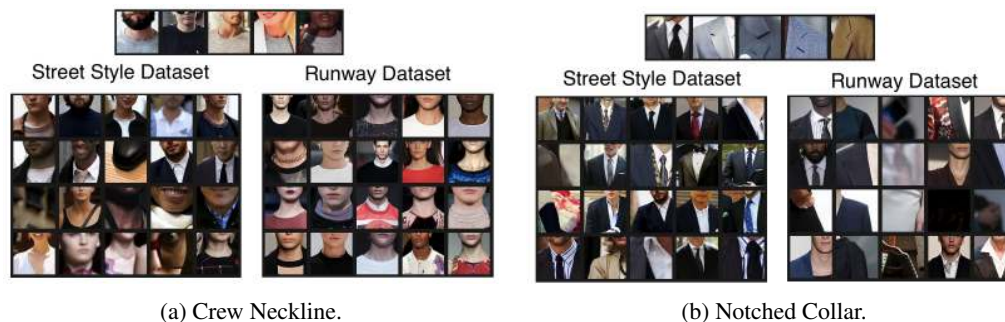


Figure 4.18: *Top 20 detections of classifiers trained on two different datasets.* The top row of images are the seed examples for the ‘crew neckline’ (left) and ‘notched collar’ (right) classifiers. The left block of images contains detections from the classifier trained and tested on the Sartorialist dataset (street style images). The right block contains detections from the classifier trained and tested on the NYT Runway dataset (runway fashion images). The ‘crew neckline’ classifier has better performance on the runway dataset because the runway images contain more examples of that attribute without coats, scarves, or other layered clothing obstructing the neckline, as is the case with the street style images. The ‘notched collar’ classifier has better performance on the street style dataset because the Sartorialist dataset has a focus on menswear.

We have shown that the crowd is surprisingly capable of training detectors after they are exposed to our one-shot learning method. An end user can spend minimal effort to provide one or five training examples which, by themselves, are not capable of generating an accurate detector. The crowd can learn from these few examples to build a high precision detector.

The linear SVM classifier and CNN image features selected for these experiments likely underperform domain specific detection strategies and fine-tuned deep networks. Still, our active learning system can create successful detectors in two disparate visual domains. our work shows that by appreciating human one-shot learning it is possible to bootstrap classifiers that approach the performance of traditional baselines yet are still inexpensive in both time and capital investment.

4.8 Detecting difficult to name concepts

Another advantageous property of Tropel is that it can train detectors for concepts that aren’t easy to name. For instance, what if a user wants to detect architectural elements similar to the Parisian bridge span shown in Fig. 4.19? There exists no annotated dataset to train such a detector. We use an unlabeled

Select Window



Figure 4.19: *Seed Patch Selection*. A user selects the region which will seed the crowd active learning of a detector.

dataset of 70 thousand Paris photos from *flickr.com* for the active learning process. Figure 2.14 shows the top 5 most confident detections for five distinctive building elements from Paris street scenes. These five examples were identified by Tropel users using the selection tool show in Fig. 4.19.

The detectors shown in Fig. 2.14 are qualitatively similar to the discriminative patch detectors found automatically by Doersch et al. in “What makes Paris look like Paris?” [16] by mining a collection of Paris and non-Paris photographs. Our scenario is different in that a user directs which concept should be detected.

The pipeline presented in Doersch et al. automatically discovers discriminative architectural details from an unlabeled dataset of street view images from Paris. Our pipeline gives users the opportunity to identify architectural details they find discriminative or important, and directly create classifiers for those often unnamable things. To the authors’ knowledge, active learning has not been employed in the literature to create classifiers for non-nameable visual events. Figure 4.19 shows how a user could select a tricky to describe visual element to seed the detector creation process.

Tropel also makes it possible to dynamically create a multi-detector search. In Figs. 4.22, 4.23, and 4.24, we search for matching scenes by identifying a small number of discriminative visual elements. For example in Fig. 4.22, the bell tower and large stained-glass window are identified as discriminative of that particular cathedral. We create two Tropel detectors, one for each item. Then we evaluate all images in our GPS tagged dataset of Paris to select the images for which these two detectors fire most strongly. The matching images, as identified by our multi-detector search, are shown on the right side of Fig. 4.22. The top detection is the same cathedral from a different vantage point.

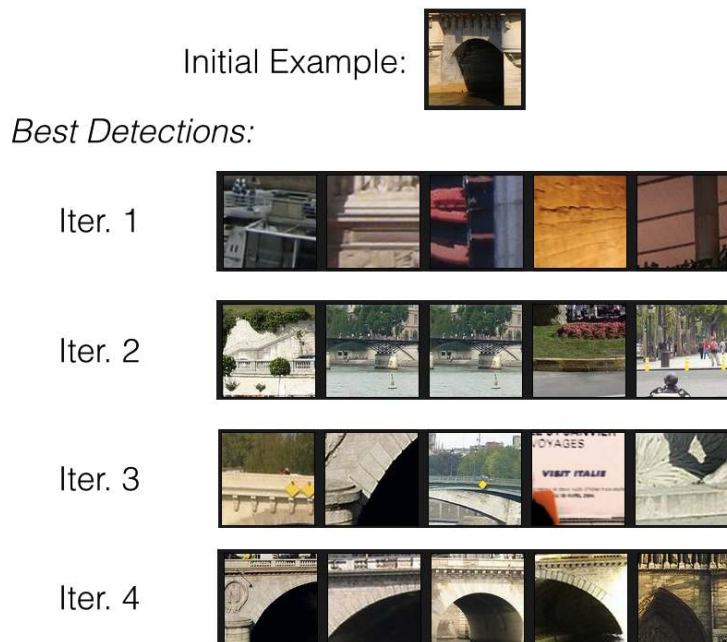


Figure 4.20: *Example detections at all iterations of the pipeline: Architecture.* The top 5 detections in our unlabeled dataset for the architectural element identified in Fig. 4.19. Iterations 2 - 4 are interesting because the crowd corrects the detectors from initially fixating on metal bridges to instead detect stone bridges.

4.9 Omitting the Crowd: Detectors created by End-Users Only

The central question examined by the previous sections was whether it was possible to drastically reduce the supervision required from an end user and still get a viable detector. The one-shot crowd active learning is an alternative to having an end user perform the potentially tedious active learning process themselves. But how much are we losing by relying on the crowd?

This question is difficult to answer because it depends on the particular end-user. A domain expert could probably do better than the crowd. However, the results of a comparison experiment shown in Fig. 4.25 illustrate that it is surprisingly hard to beat the crowd. In this experiment, a volunteer end-user with a reasonable level of bird species knowledge trained 20 detectors by answering all active queries in the training pipeline. The end-user created classifiers are compared to classifiers created by the crowd that are initialized with the same seed patches. The crowd workers are not individually particularly accurate, but Fig. 4.25 shows that the consensus of three or more workers is hard to beat.

In a few cases (Boat-tailed Grackle, Pine Grosbeak, Worm-eating Warbler), the end-user was able to exploit her specialized knowledge to distinguish these birds from visually similar birds to make a

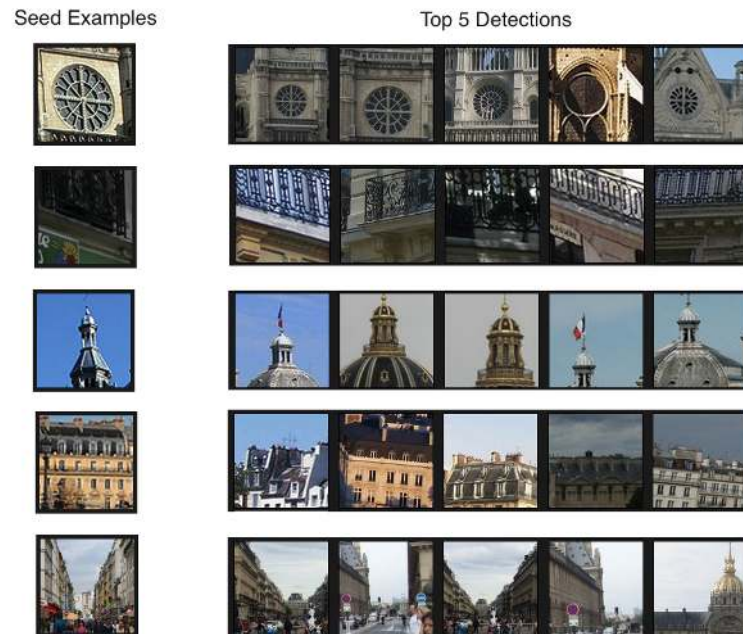


Figure 4.21: *Example detections for 5 architectural concepts.* While these are distinctive architectural elements (from top to bottom: Catherine windows, exterior wrought iron balustrades, cupola lanterns, Mansard-roofed apartment buildings, and Parisian urban canyons) the end user and the crowd didn't use these names or have architectural expertise.

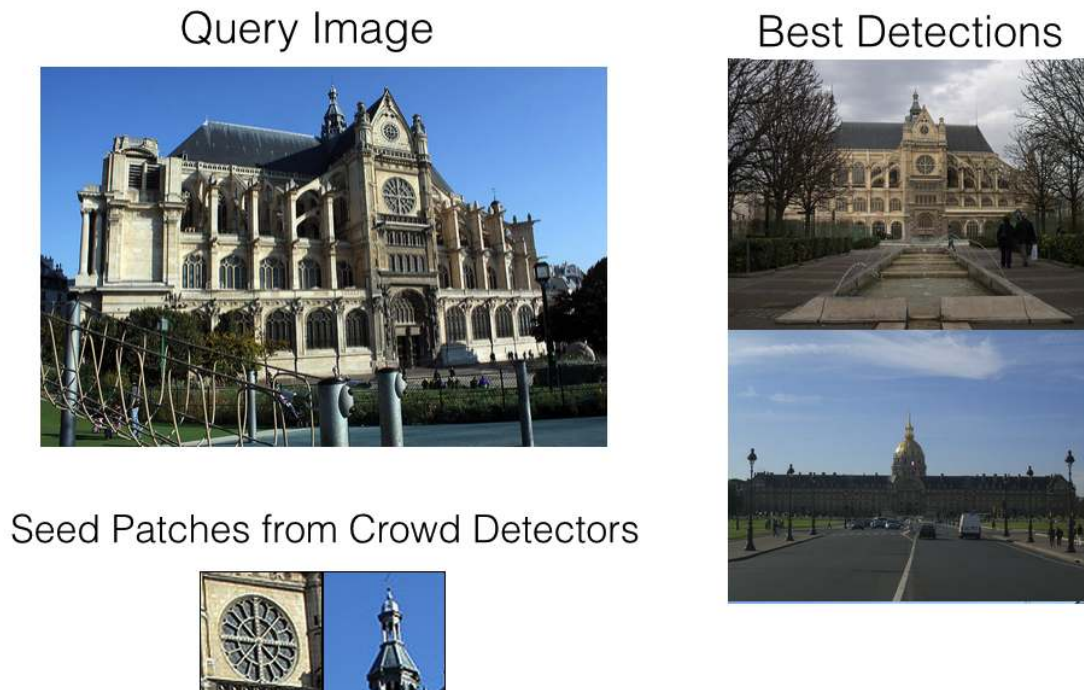


Figure 4.22: *Multi-Detector Search: Cathedral.* In this multi-detector search, we created Tropel detectors for a bell tower and stained-glass window.

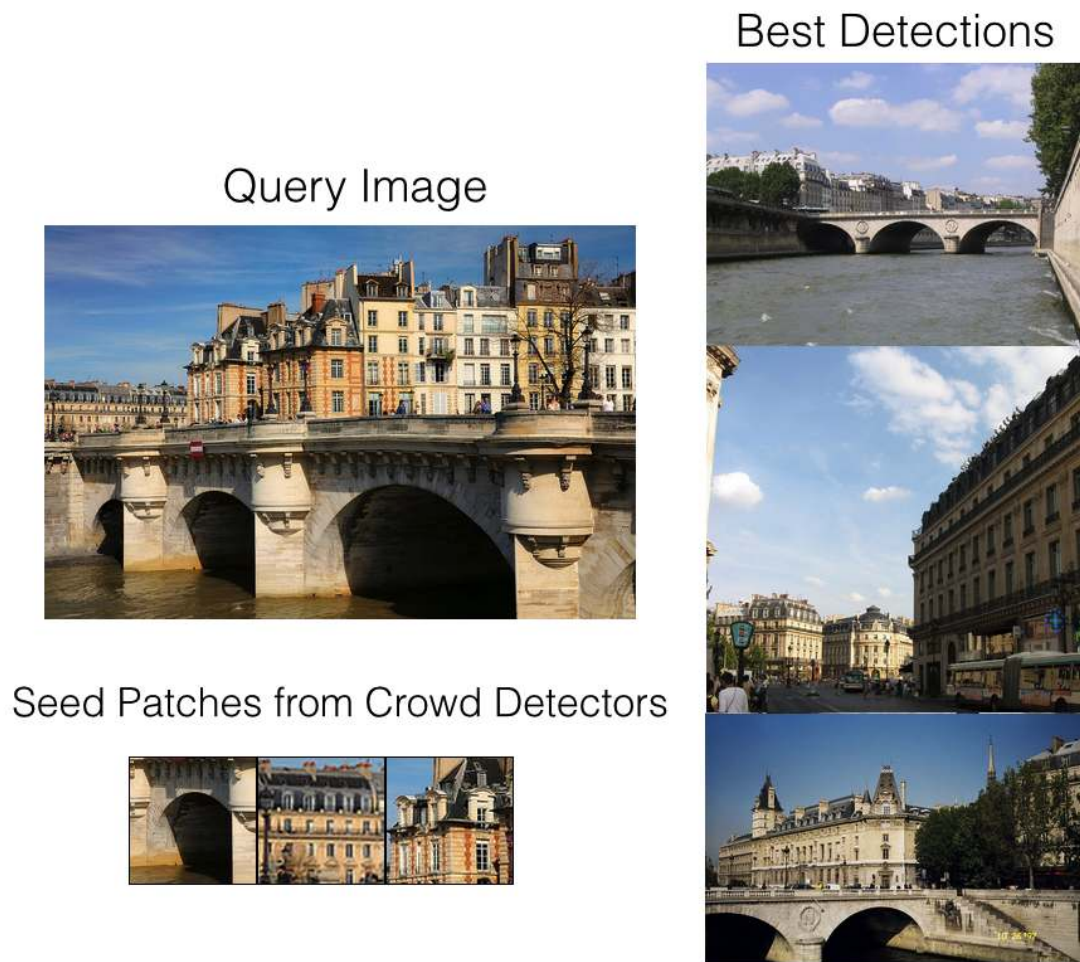


Figure 4.23: *Multi-Detector Search: Bridge*. In this multi-detector search, we created Tropel detectors for a bridge arch and two buildings with mansard roofs.

Query Image



Seed Patches from Crowd Detectors



Best Detections



Figure 4.24: *Multi-Detector Search: Fruit Stand*. In this multi-detector search, we created Tropel detectors for a produce stand, a restaurant black board, and a wrought iron balcony railing.

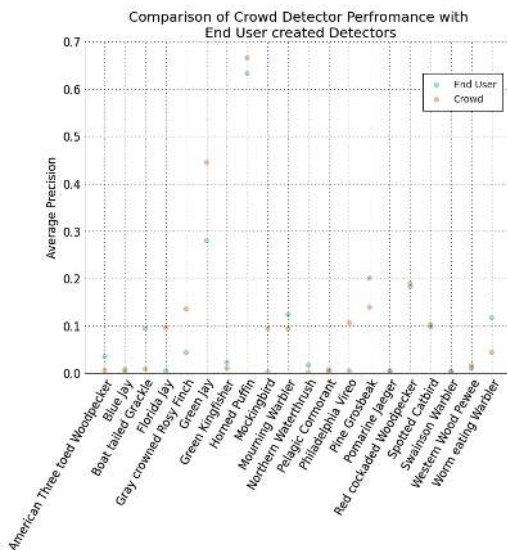


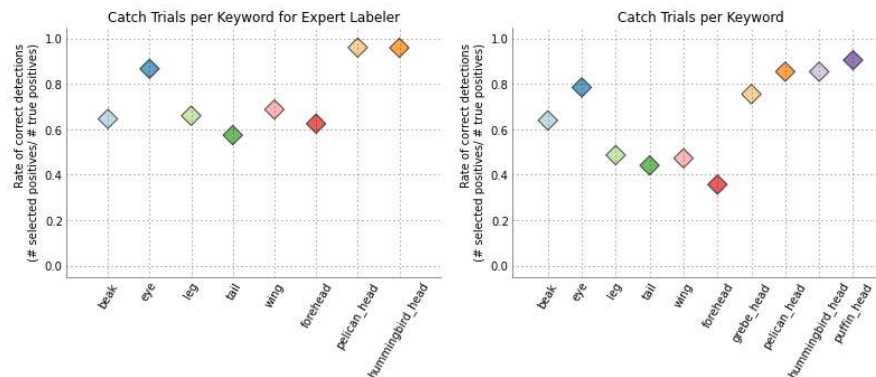
Figure 4.25: *Detection AP*. Comparison of the detection AP for classifiers created by the crowd and by a single end user. The average AP among these birds is 0.095 for end user only detectors and 0.11 for crowd trained classifiers.

better classifier. These birds are particularly tricky to distinguish. They are evenly colored black, red, and speckled brown respectively, but otherwise have similar shape and feathering. These species each have several other bird species that they can be easily confused with. However, the crowd was able to create equally or better performing classifiers for most of the other 17 randomly selected species.

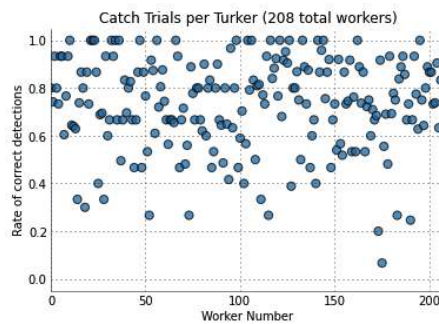
4.10 Assessing Worker Performance

We implement catch trials for our bird and fashion attribute classifiers. Our catch trials were generated by the author of this dissertation. She labeled dozens of birds and fashion items with ground truth labels that were secretly added to the active query items in hundreds of Tropel HITs. For each HIT that included a catch trial, five positive items and five negative items were randomly placed in the display of query patches. The negative patches were randomly selected from all of the patches in the dataset. Figs. 4.26 and 4.27 show the catch trial performance the AMT workers for fashion attributes. Overall, the annotators are more reliable and accurate for the fashion concepts. This could be because these attributes are more commonly found in daily life.

During our experiments with Tropel, we compared weighting worker query responses in order to obtain more accurate consensus responses. Reducing worker labeling noise is difficult in the Tropel context because ground truth labels are unknown and workers are ephemeral. For our experiments involving the

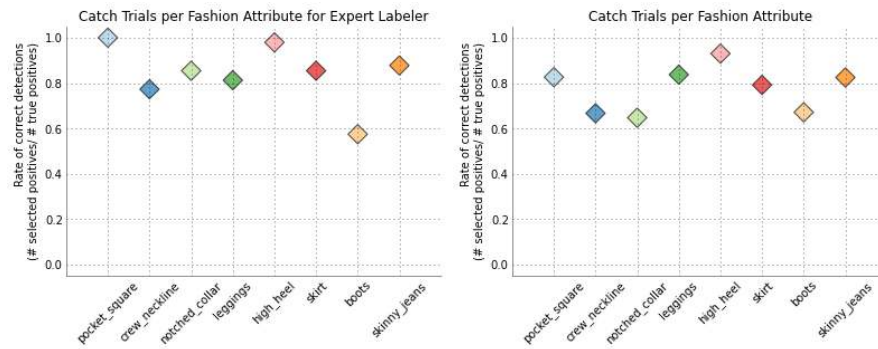


(a) Labeling Accuracy of Expert per Detector. (b) Labeling Accuracy of Crowd per Detector.

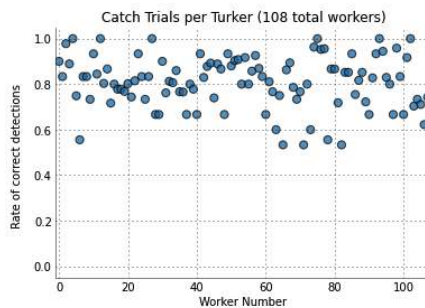


(c) Overall Labeling Accuracy of each Crowd Worker.

Figure 4.26: *Bird Annotator Accuracy*. The plots above show that the accuracy of the crowd annotators is similar to that of the expert, except in the cases of 'leg', 'tail', 'wing', and 'forehead'. Despite the lower crowd accuracy, the classifiers for those attributes perform similarly to or better than the expert trained version in Fig. 4.26a. In Fig. 4.26c the large majority have similar or slightly less accuracy than an expert annotator. The 'Expert' worker for this experiment was an undergraduate researcher at Brown University.



(a) Labeling Accuracy of Expert per Detector. (b) Labeling Accuracy of Crowd per Detector.



(c) Overall Labeling Accuracy of each Crowd Worker.

Figure 4.27: *Fashion Annotator Accuracy*. The plots above show that the accuracy of the crowd annotators is higher and similar across more categories than for the bird-related detectors. The ‘Expert’ worker for this experiment was an undergraduate researcher at Brown University.

CUB dataset, we were able to measure the recall for the workers who participated in HITs for Tropel. Recall is highly important in this active learning setting because positive instances of a given visual event are rare. In Fig. 4.28 we compare how different worker meta-data are correlated to a worker’s recall. Ultimately, none of the meta-data variables – self-reported confidence, HIT completion time, or number of HITs submitted – were strongly correlated with recall, which reflects our earlier evaluation in this chapter that consensus weighting strategies did not strongly improve Tropel detector AP scores.

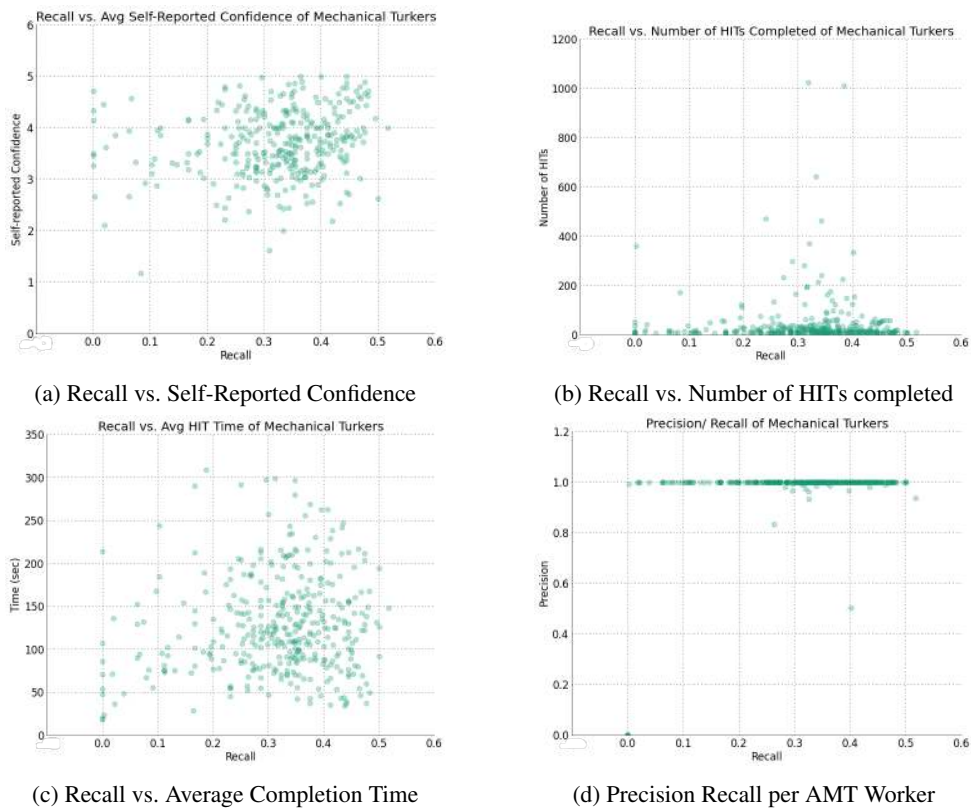


Figure 4.28: *Annotator Recall*. Data for approximately 250 randomly selected workers are shown in the above graphs. Each green dot in each graph represents a unique Tropel worker.

4.11 Conclusions from building the Tropel System

We have shown that the crowd is surprisingly capable of training detectors for specific, fine-grained visual phenomena. An end user can spend minimal effort to provide one or five training examples which, by themselves, are not capable of generating an accurate detector. The crowd can learn from these few examples to build a high precision detector. Our investigations into the hierarchical similarity of the Tropel detectors' output showed how fine-grained a detector it is possible to obtain with limited instructions. The more limited the instructions to the crowd, the more the crowd generalizes to high level concepts.

Tropel is biased towards high-precision rather than complete recall. It may be possible to change the active query strategy to achieve better recall. In this chapter, we opted not to investigate alternative active query strategies to limit the scope of our experimentation. However, improving recall is an important next step.

The linear SVM classifier and CNN image features selected for these experiments likely underperform domain specific detection strategies and fine-tuned deep networks. Still, our active learning system can create successful detectors in three disparate visual domains. Our work shows that it is possible to bootstrap classifiers from a single visual example that approach the performance of traditional baselines yet are still inexpensive in both time and capital investment.

Chapter 5

Conclusion

At the beginning of this dissertation, we set out to show the utility of incorporating a crowd workforce into the computer vision pipeline. In Chapter 2 the crowd helped us to discover discriminative visual attributes and label a large dataset. In Chapter 3 we changed the role of the crowd from annotation engine to annotation-bootstrapping pipeline. The crowd helped us both to label visual events and to figure out which visual events were most likely to occur in an image, thus decreasing the number of items to label. We were able to show the crowd to be a useful tool for raw knowledge acquisition and for intelligent annotation protocol.

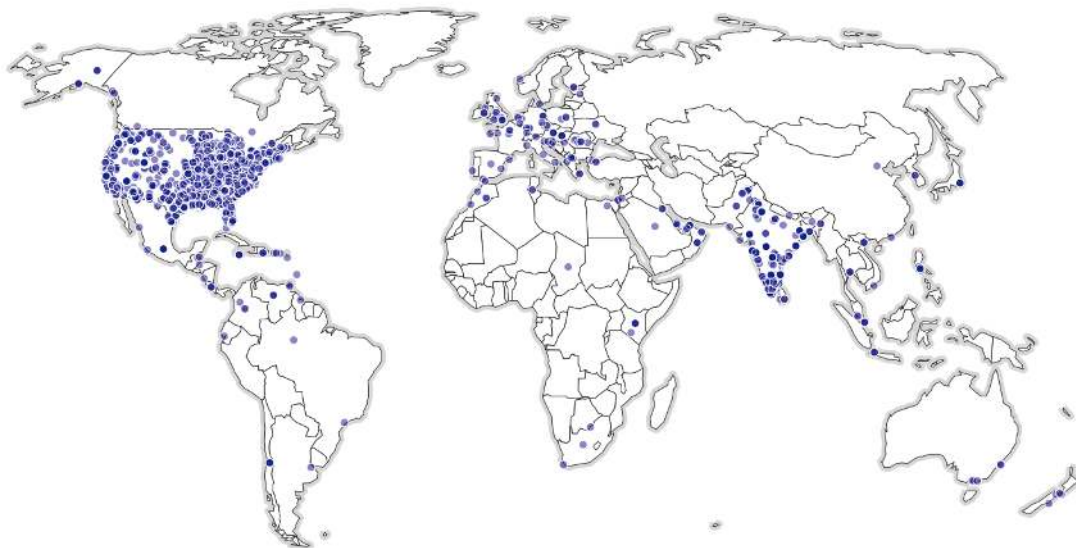
Chapter 4 demonstrated the usefulness of the crowd in a much different role. In Chapters 2 and 3 the crowd provided training data for automated systems. In Chapter 4 the crowd was an integral component in the computer vision pipeline. The systems presented in Chapter 4, the Tropel system in particular, show how the crowd could be used to enable end-users to have more control over creating fine-grained object detectors.

Ultimately, this dissertation is proof of the usefulness of the crowd for computer vision. Each one of the experiments contained in this document could be extended or executed on a larger scale, revealing even more about the nature of the crowd workforce as a computational aid. There is still an unknown number of ways the crowd could help bring computer vision closer to natural image understanding and bring computer vision into the daily lives of users.

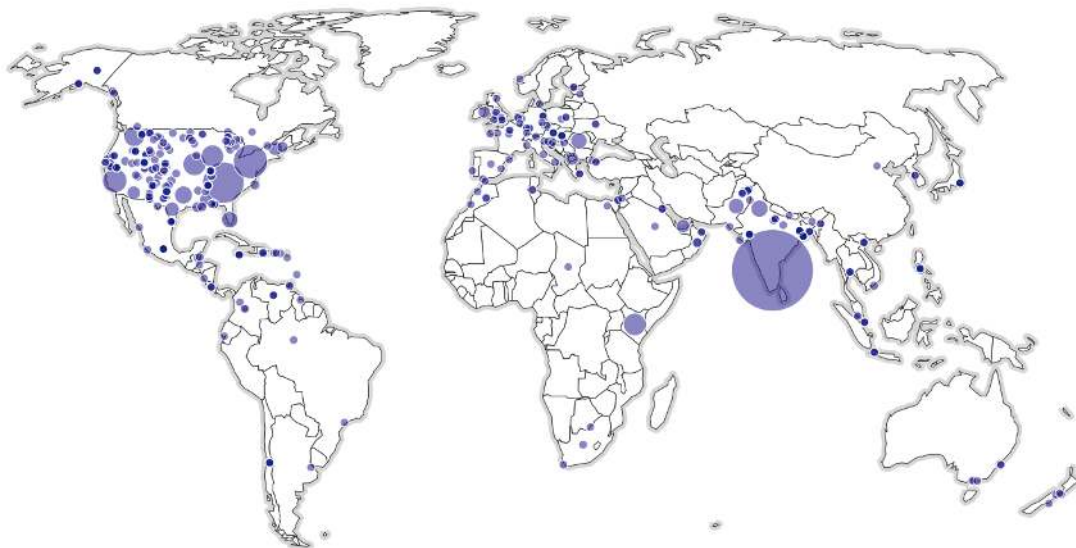
Appendix A

Nationalities of the Mechanical Turk Workforce

Thousands of Amazon Mechanical Turk (AMT) workers contributed to the experiments presented in this dissertation. The following figures provide a broad description of the geolocaition and work product of our AMT workforce. We recorded IP addresses for all of the workers for the MS COCO Attributes Dataset project (Chapter 3) and the Tropel project (Chapter 4). Figures A.1 and A.2 plot the locations of workers on the globe, while Tables A.1 and A.2 list that data in table format.



(a) *Geolocation for MS COCO Attributes Workers*

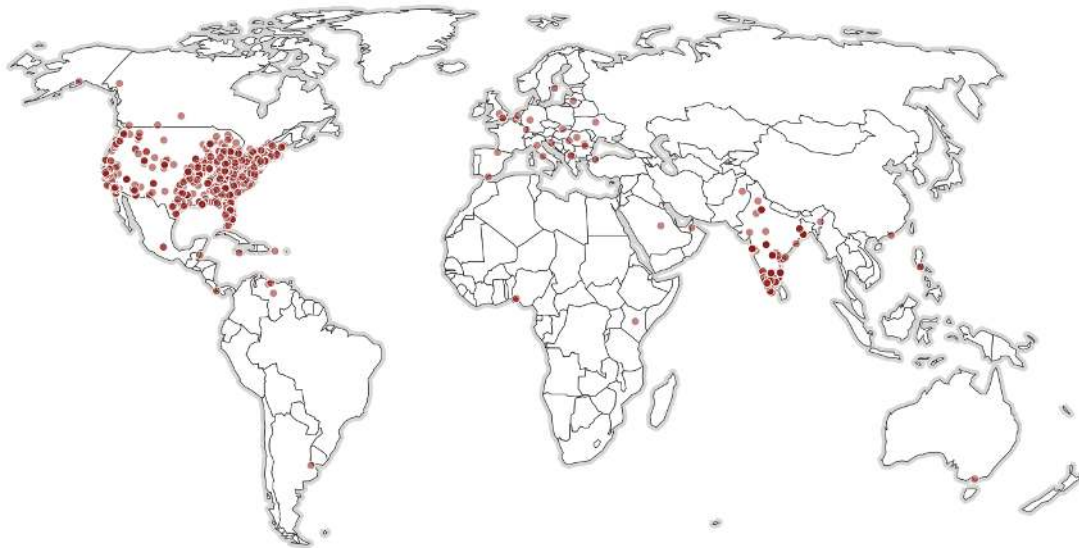


(b) *Number of HITs submitted by Region for MS COCO Attributes Workers*

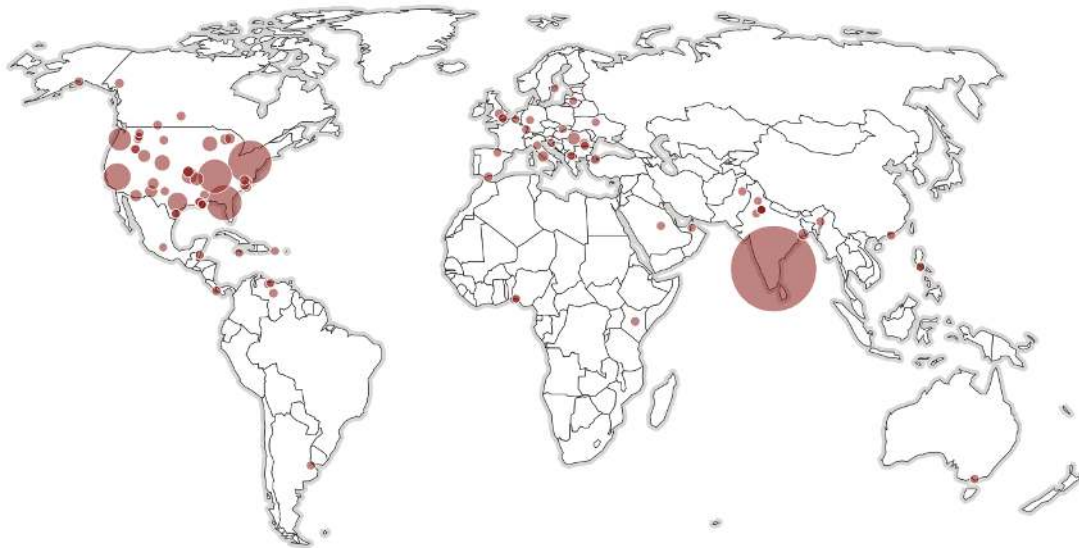
Figure A.1: MS COCO Attributes Worker Information The two figures above characterize the COCO Attributes workforce by GPS location and HITs submitted. The first sub-figure represents each worker with a dot for the location corresponding to their IP address. There is one dot for each worker. The second sub-figure aggregates the number of HITs submitted in a particular region. Larger dots correspond to more HITs submitted by workers in the municipality where the dot is centered.

Table A.1: COCO Attributes Worker Nationality

Country	Number of HITs Submitted	Number of Workers	Country	Number of HITs Submitted	Number of Workers
India	109872	831	Ukraine	6	2
United States	60442	4794	Singapore	6	3
Kenya	4129	5	Indonesia	6	3
Romania	1307	6	Hungary	6	4
Pakistan	1174	7	Turkey	6	3
United Arab Emirates	568	6	Czech Republic	5	2
Macedonia	518	6	Brazil	5	2
Ireland	325	3	Korea, Republic of	5	2
Oman	235	5	Thailand	5	3
Dominican Republic	108	3	Poland	5	3
Canada	103	18	France	5	5
Philippines	79	15	Netherlands	4	2
Australia	55	5	Sri Lanka	4	3
Mexico	32	11	Spain	4	4
United Kingdom	28	14	Saint Kitts and Nevis	3	1
Jamaica	25	4	Virgin Islands, U.S.	3	1
Kuwait	24	2	Finland	3	2
Italy	18	4	Nicaragua	2	1
Viet Nam	17	4	Honduras	2	1
Morocco	17	4	Guyana	2	1
Chile	16	5	Denmark	2	1
Belize	15	2	Norway	2	1
Tunisia	15	2	Barbados	2	1
South Africa	13	2	Malaysia	2	2
New Zealand	13	4	Albania	1	1
Bangladesh	13	3	Botswana	1	1
Japan	13	4	Jordan	1	1
Germany	13	9	China	1	1
Puerto Rico	11	3	Saudi Arabia	1	1
Austria	11	4	Estonia	1	1
Serbia	10	1	Portugal	1	1
Costa Rica	10	3	Egypt	1	1
Greece	9	2	Belgium	1	1
Qatar	8	3	Gibraltar	1	1
Croatia	8	3	Slovenia	1	1
Trinidad and Tobago	8	2	Ecuador	1	1
Israel	8	2	Argentina	1	1
Colombia	8	3	Nepal	1	1
Venezuela	7	3	Chad	1	1
			Hong Kong	1	1



(a) *Geolocation for Tropel Workers*



(b) *Number of HITs submitted by Region for Tropel Workers*

Figure A.2: Tropel Worker Information The two figures above characterize the Tropel workforce by GPS location and HITs submitted. The first sub-figure represents each worker with a dot for the location corresponding to their IP address. There is one dot for each worker. The second sub-figure aggregates the number of HITs submitted in a particular region. Larger dots correspond to more HITs submitted by workers in the municipality where the dot is centered.

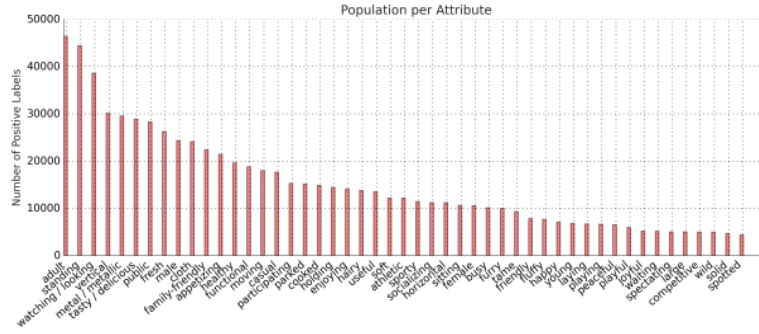
Table A.2: Tropel Worker Nationality

Country	Number of HITs Submitted	Number of Workers
India	10502	341
United States	6604	1045
Romania	48	4
Canada	32	6
Italy	20	2
United Kingdom	12	3
Venezuela, Bolivarian Republic of	9	3
Spain	8	2
Philippines	6	2
Turkey	5	1
Kenya	4	1
Australia	3	1
France	3	1
Panama	3	1
Nigeria	3	2
United Arab Emirates	2	1
Argentina	2	1
Hong Kong	2	1
Pakistan	2	1
Belize	2	1
Saudi Arabia	2	1
Macedonia, Republic of	2	2
Mexico	2	2
Slovakia	1	1
Ukraine	1	1
Jamaica	1	1
Belgium	1	1
Lithuania	1	1
Puerto Rico	1	1
Croatia	1	1
Sweden	1	1
Germany	1	1
Netherlands	1	1

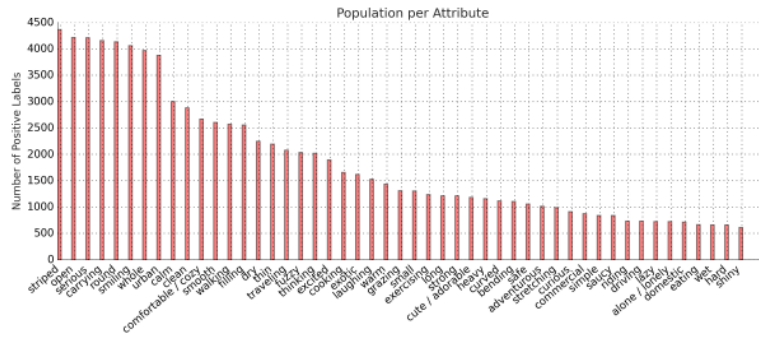
Appendix B

Extended Results

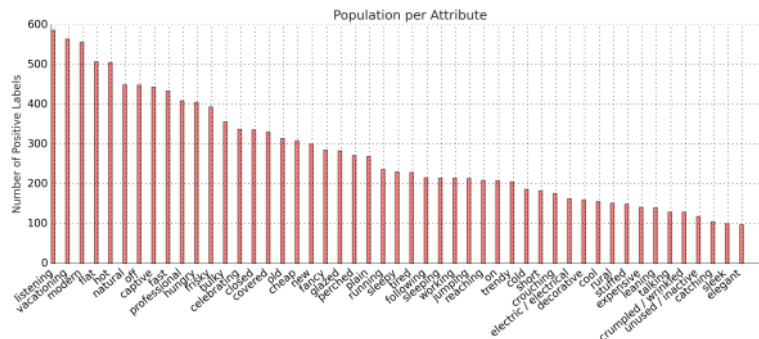
The three figures included in this appendix extend or better explain several of the figures in Chapter 3. These supplemental figures include a plot of the population for each attribute, the recall of the ELA method for each attribute, and the results of a multilabel classification experiment that tests classification AP for all MS COCO attributes.



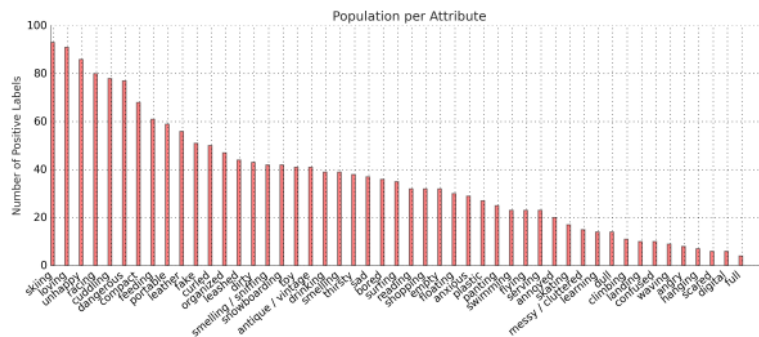
(a) Attributes 1-49



(b) Attributes 50-98



(c) Attributes 99-147



(d) Attributes 148-196

Figure B.1: *Attribute Population* In the figure above, we plot the population of each attribute in the MS COCO Attributes dataset. The population number is counted across object categories, thus the population of the ‘standing’ attribute comes from the number of times ‘standing’ occurred in all instances of people, animals, etc. Both plots have a log-scale y-axis. The attributes are sorted by population. This figure highlights both the amount of attribute instances and the wide range from common to rare attributes present in this dataset.

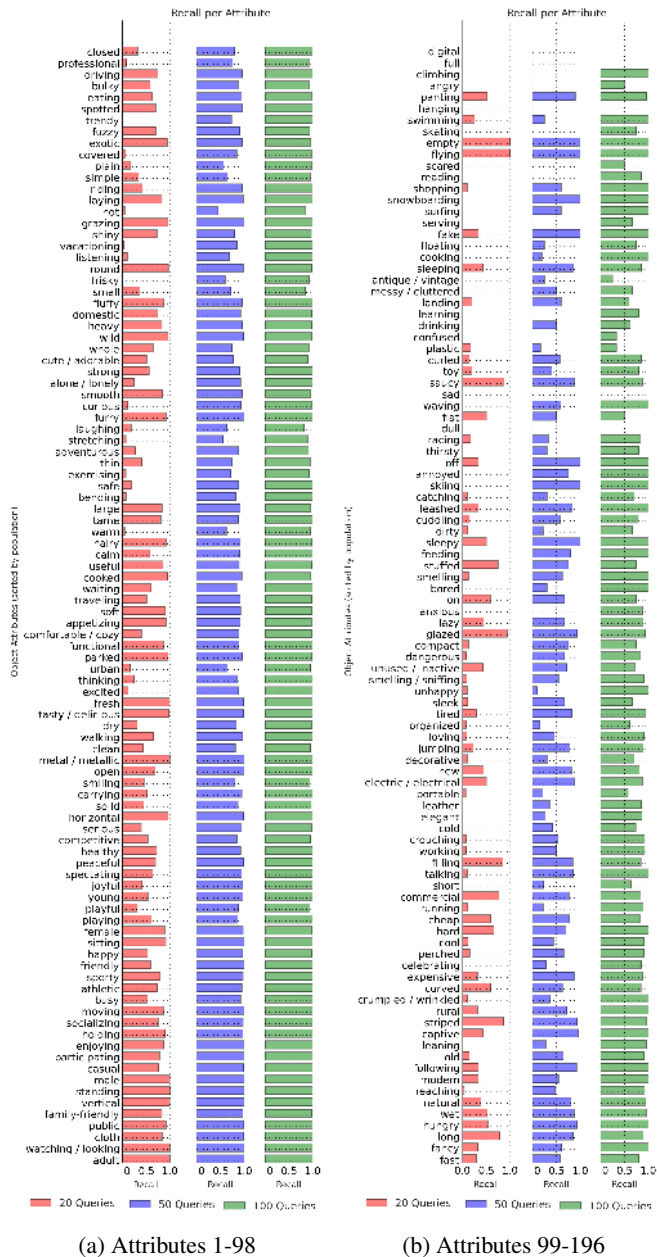


Figure B.2: *Mean Recall Across all Categories for all Attributes*. This plot shows the recall of the ELA-Distance method for annotating all attributes from the full set of 196. The recall is calculated across all instances of the exhaustively labeled test set. The attributes are sorted by their popularity in the exhaustively labeled training dataset. This figure is an expansion of Fig. 3.3 in Chapter 3. This figure shows that generally, more popular attributes are recovered in the first 20 iterations of the ELA method, but rare attributes are also frequently recovered. At a higher number of iterations, 50 or 100 queries which is still only 25% and 50% of the entire dataset respectively, the ELA method has high recall with rare attributes. Only a few attributes are almost never recovered by the ELA, for example ‘digital’. These attributes were among the rarest in the dataset.

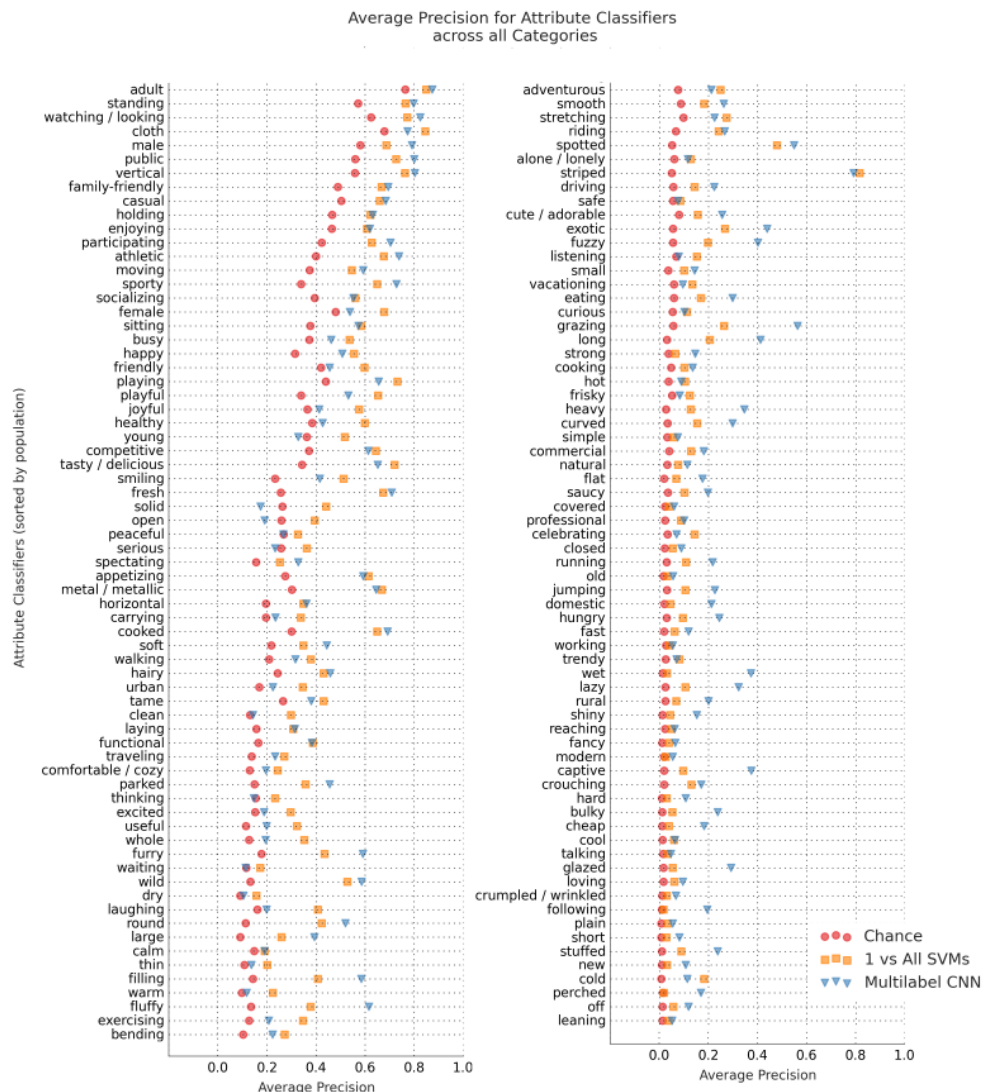


Figure B.3: *Average Precision vs. Chance*. This figure is the full length version of Fig. 3.7 in Chapter 3. Performance is shown for all 196 MS COCO attributes. Attributes are sorted in descending order by their population in the dataset. Each yellow square represents an SVM that was trained to recognize that particular attribute. All SVM training and test sets are composed of consensus AMT labels. A positive attribute label occurs when 2/3 AMT workers voted ‘true’ for the presence on a given attribute and 0 workers voted for false. Thus, the training and test sets for each attribute SVM are a different size. Each blue triangle represents the AP for that attribute calculated on the full multi-label test set predictions of our CNN. The objects in the training set for all classifiers shown are members of the MS COCO ‘train2014’ set, and test instances are members of the ‘val2014’ set. The ratio of train to test for each SVM is approximately 2-to-1. For the multi-label CNN, the train/test set sizes were 27k/18k.

Appendix C

Choosing a Title

Picking a thesis title can be a bit of a fussy problem. Selecting the perfect set of words to describe the content of a thesis at a high level can be additionally difficult for the person who has had their nose in a project for the last several years. We decided instead to let the crowd create our thesis title for us.

Our thesis title creation process began by asking a personally engaged crowd, our Facebook friends, to suggest titles. Our friends took zealously to the task, and over the course of two hours submitted the list of titles in Table C.1. These titles were ingenious and delightful, but it was too hard for us to pick the best one. We decided to crowdsource that task as well.

We created an AMT HIT to vote for the best title and suggest changes and main title, subtitle combinations. The HIT consisted of a one paragraph description of the research in this dissertation, examples titles from successful PhD thesis in Computer Vision, and instructions to vote on the list of existing titles and suggest new ones. The AMT votes for the original Facebook titles are also listed in Table C.1. The suggested improved titles from AMT are listed in Table C.3. Ultimately, the winner was a combination of an AMT submission for title and a Facebook submission for subtitle. We picked the winner by our own personal preference.

Because the AMT workers are essentially anonymous, we asked the workers to report their level of education. The population of the different degree levels is reported in Table C.2. Many of the workers had completed a Bachelor's degree, which we found to be a surprisingly high level of education for such an economical workforce. Table C.3 lists the education of the worker along with their submitted title. The quality of the submitted title did not appear to be correlated with the worker's level of education.

The largest percentage of workers, 20%, spent between 40sec - 1min to complete the voting and

Table C.1: Titles contributed on Facebook and the corresponding number of votes for each title from AMT.

Facebook Submitted Titles	Votes from AMT Workers
The Democracy of Vision	10
Peering into the Crowd	9
Mob Mentality and Pixel Perception	6
Crowd-driven Image Understanding	6
Groupsight	5
To See or Not to See	4
Crowdconomics	3
The In Crowd: It's All Relative	3
Hive Mind	2
Multitudes of Insight	2
Crowded	1
Crowdocracy	1
Crowdrophemia	1
The Human Crowd Affair	1
Crowds Will Be Crowds	1
The Colloquial Hive	1
Crowd-Sourced Internet Colloquialism	0
Peer Pressure	0
Crowdsourcing Images	0
Crowding	0
Crowd Control	0
From the Crowd	0

Table C.2: Self-reported education level of AMT workers in title generating experiment.

Education Level	Number of AMT Workers
Some High School	2
Bachelors degree	26
Graduate degree (Masters, Doctorate, etc.)	4
Some college, no degree	18
Associates degree	6

new title suggestion HIT. An other 13% took up to 1.5mins, and 18% more took up to 2.5mins. This demonstrates that the workers put a reasonable amount of thought into their suggestions. The worker who submitted the winning title spent 15mins on their hit. They suggested a set of four distinct titles, shown in the final row of Table C.3.

Table C.3 lists all suggested titles submitted by the AMT workers. Of the 56 workers that participated in this endeavor, 54 submitted title. Several workers submitted multiple titles. Although many of the titles were not accurate representations of the contents of this dissertation, many were clever and interesting. In the end, we were very happy using this process to generate our title. The total cost to create the final title “Collective Insight: Crowd-driven Image Understanding”, was \$0.55 per HIT (AMT fee included) for 56 unique HITs, which comes to \$30.80.

Table C.3: All Titles Submitted via AMT

AMT Submitted Titles	Elapsed Time (mins:secs)	Education Level of Worker
Do you like what you see?	1:02	Some college, no degree
Perceptual Democratization	3:08	Graduate degree (Masters, Doctorate, etc.)
Hive Thrive	1:52	Some college, no degree
The Vision of the Crowd	0:51	Bachelors degree
See the People	1:54	Bachelors degree
Crowdamaniacs	0:14	Bachelors degree
Eye of the Crowd	2:19	Some college, no degree
Through the eyes of the 'crowd' - Understanding Images	5:43	Graduate degree (Masters, Doctorate, etc.)
Crowdsight	2:33	Bachelors degree
Understanding Images From The Human Crowd's Point of View	0:41	Some college, no degree
Insights From Many Angles	4:32	Some college, no degree
Crowding Images from Clouds	5:17	Bachelors degree
Crowds: Going beyond what you see.	1:30	Associates degree
AllSight	37:08	Graduate degree (Masters, Doctorate, etc.)
The Science of GroupThink	29:20	Bachelors degree
Hive Mind and Vision, Peering Peers	3:05	Bachelors degree
Peering into Pixels	4:46	Some college, no degree
Human Vision	1:02	Bachelors degree
Seeing from the Human Point of View	5:49	Bachelors degree
Crowdidentification	2:07	Bachelors degree
Seeing it all	1:25	Some college, no degree
Visuals and Humans	1:04	Bachelors degree
Many Eyes	0:55	Some college, no degree
Compelling the Eye	1:34	Bachelors degree
Rose Tinted Crowds. Crowded shaded image perception	1:20	Bachelors degree
human minds connect	2:39	Some High School
Matrixity	4:09	Associates degree
A Democratic Vision	1:25	Some college, no degree
Crowd Hive	0:46	Bachelors degree
Mass perception of images	3:16	Bachelors degree
peoples perception	4:18	Bachelors degree
human perceptive of computer imagery	15:22	Bachelors degree
Peer Control	1:16	Some college, no degree
The More the Merrier	2:09	Associates degree
Crowdvision	0:55	Bachelors degree
You could shorten it to "Peer into the crowd" or "From the eye of the crowd"	0:58	Some college, no degree
Games of Crowds	5:33	Bachelors degree
The Crowd Mind	1:26	Some college, no degree
Open mind	13:34	Bachelors degree
In the Public Eye	3:35	Graduate degree (Masters, Doctorate, etc.)
Collecting the Crowd	1:59	Some college, no degree
Crowdology	0:54	Bachelors degree
Crowdology	4:01	Bachelors degree
CrowdNow	25:00	Bachelors degree
Crowded Vision	2:07	Some college, no degree
Crowdsourced 3D Scene Matching	4:39	Bachelors degree
Behind The Crowd: What Appeals To You?	2:13	Some High School
Group In-'sight'	1:04	Some college, no degree
Crowd Vision	0:47	Associates degree
Ode to the Rolling Stones Their Title- I said, "Hey (hey), you (you). Get off of my crowd."	2:43	Associates degree
Peering into the Crowd: How Group Perceptions on Beauty Help to Understand Compelling Imagery	46:36	Some college, no degree
See through the eyes of the crowd; by the crowd, for the people; Crowdgazing: how we see what we all see	4:52	Associates degree
Mob Mentality and Pixel Perception: A Deeper Look at the Visually Compelling	3:19	Bachelors degree
Crowd-based Identification, Collective Insight, Multi-human Visual Processing Unit, Identification Through Group Consensus	15:04	Bachelors degree

Bibliography

- [1] Yotam Abramson and Yoav Freund. Active learning for visual object recognition. Technical report, Technical report, UCSD, 2004.
- [2] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2425–2433, 2015.
- [3] T. Berg, A. Berg, and J. Shih. Automatic Attribute Discovery and Characterization from Noisy Web Data. *ECCV*, 2010.
- [4] Lubomir Bourdev and Jitendra Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *International Conference on Computer Vision (ICCV)*, 2009.
- [5] Steve Branson, Catherine Wah, Florian Schroff, Boris Babenko, Peter Welinder, Pietro Perona, and Serge Belongie. Visual recognition with humans in the loop. In *Computer Vision–ECCV 2010*, pages 438–451. Springer, 2010.
- [6] D.L. Chen and W.B. Dolan. Building a persistent workforce on mechanical turk for multilingual data collection. *The 3rd Human Computation Workshop (HCOMP)*, 2011.
- [7] Xinlei Chen, Abhinav Shrivastava, and Abhinav Gupta. NEIL: Extracting Visual Knowledge from Web Data. In *International Conference on Computer Vision (ICCV)*, 2013.
- [8] Lydia B Chilton, Greg Little, Darren Edge, Daniel S Weld, and James A Landay. Cascade: crowdsourcing taxonomy creation. In *Proceedings of the 2013 ACM annual conference on Human factors in computing systems*, pages 1999–2008. ACM, 2013.
- [9] Brendan Collins, Jia Deng, Kai Li, and Li Fei-Fei. Towards scalable dataset construction: An active learning approach. In *Computer Vision–ECCV 2008*, pages 86–98. Springer, 2008.

- [10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. *CVPR*, 2009.
- [11] J. Deng, K. Li, M. Do, H. Su, and L. Fei-Fei. Construction and Analysis of a Large Scale Image Ontology. Vision Sciences Society, 2009.
- [12] Jia Deng, Nan Ding, Yangqing Jia, Andrea Frome, Kevin Murphy, Samy Bengio, Yuan Li, Hartmut Neven, and Hartwig Adam. Large-scale object classification using label relation graphs. In *Computer Vision—ECCV 2014*, pages 48–64. Springer, 2014.
- [13] Jia Deng, Jonathan Krause, and Li Fei-Fei. Fine-grained crowdsourcing for fine-grained recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013.
- [14] Jia Deng, Olga Russakovsky, Jonathan Krause, Michael S Bernstein, Alex Berg, and Li Fei-Fei. Scalable multi-label annotation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 3099–3102. ACM, 2014.
- [15] Santosh Divvala, Ali Farhadi, and Carlos Guestrin. Learning everything about anything: Webly-supervised visual concept learning. 2014.
- [16] Carl Doersch, Saurabh Singh, Abhinav Gupta, Josef Sivic, and Alexei A Efros. What makes paris look like paris? *ACM Transactions on Graphics (TOG)*, 31(4):101, 2012.
- [17] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. *arXiv preprint arXiv:1310.1531*, 2013.
- [18] Steven Dow, Anand Kulkarni, Scott Klemmer, and Björn Hartmann. Shepherding the crowd yields better work. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*, pages 1013–1022. ACM, 2012.
- [19] K. A. Ehinger, J. Xiao, A. Torralba, and A. Oliva. Estimating scene typicality from human ratings and image features. *33rd Annual Conference of the Cognitive Science Society*, 2011.
- [20] I. Endres, A. Farhadi, D. Hoiem, and D. Forsyth. The Benefits and Challenges of Collecting Richer Object Annotations. *Advancing Computer Vision with Humans in the Loop (ACVHL) (in conjunction with CVPR)*, 2010.

- [21] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010.
- [22] A. Farhadi, I. Endres, and D. Hoiem. Attribute-centric recognition for cross-category generalization. *CVPR*, 2010.
- [23] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.
- [24] Vittorio Ferrari and Andrew Zisserman. Learning visual attributes. *NIPS*, 2008.
- [25] Donald Geman, Stuart Geman, Neil Hallonquist, and Laurent Younes. Visual turing test for computer vision systems. *Proceedings of the National Academy of Sciences*, 112(12):3618–3623, 2015.
- [26] Yotam Gingold, Ariel Shamir, and Daniel Cohen-Or. Micro perceptual human computation for visual tasks. *ACM Transactions on Graphics (TOG)*, 31(5):119, 2012.
- [27] R. Gomes, P. Welinder, A. Krause, and P. Perona. Crowdclustering. In *Neural Information Processing Systems (NIPS)*, 2011.
- [28] Yunchao Gong, Yangqing Jia, Thomas Leung, Alexander Toshev, and Sergey Ioffe. Deep convolutional ranking for multilabel image annotation. *arXiv preprint arXiv:1312.4894*, 2013.
- [29] M.R. Greene and A. Oliva. Recognition of natural scenes from global properties: Seeing the forest without representing the trees. *Cognitive Psychology*, 58(2):137–176, 2009.
- [30] Matthieu Guillaumin, Thomas Mensink, Jakob Verbeek, and Cordelia Schmid. Tagprop: Discriminative metric learning in nearest neighbor models for image auto-annotation. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 309–316. IEEE, 2009.
- [31] Derek Hoiem, Yodsawalai Chodpathumwan, and Qieyun Dai. Diagnosing error in object detectors. In *Computer Vision–ECCV 2012*, pages 340–353. Springer, 2012.
- [32] Panagiotis G Ipeirotis, Foster Provost, and Jing Wang. Quality management on amazon mechanical turk. In *Proceedings of the ACM SIGKDD workshop on human computation*, pages 64–67. ACM, 2010.

- [33] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the ACM International Conference on Multimedia*, pages 675–678. ACM, 2014.
- [34] M. Juneja, A. Vedaldi, C. V. Jawahar, and A. Zisserman. Blocks that shout: Distinctive parts for scene classification. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2013.
- [35] Adriana Kovashka and Kristen Grauman. Attribute adaptation for personalized image search. *ICCV*, 2013.
- [36] Adriana Kovashka, Devi Parikh, and Kristen Grauman. Whittlesearch: Image Search with Relative Attribute Feedback. *CVPR*, 2012.
- [37] Adriana Kovashka, Devi Parikh, and Kristen Grauman. Whittlesearch: Image search with relative attribute feedback. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2973–2980. IEEE, 2012.
- [38] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *arXiv preprint arXiv:1602.07332*, 2016.
- [39] N. Kumar, A.C. Berg, P.N. Belhumeur, and S.K. Nayar. Attribute and simile classifiers for face verification. *ICCV*, 2009.
- [40] C. H. Lampert, H. Nickisch, and S. Harmeling. Learning To Detect Unseen Object Classes by Between-Class Attribute Transfer. *CVPR*, 2009.
- [41] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Attribute-based classification for zero-shot visual object categorization. *PAMI*, pages 453–465, 2014.
- [42] Lasecki, Murray, White, Miller, and Bigham. Real-time Crowd Control of Existing Interfaces. *UIST*, 2011.
- [43] S. Lazebnik, C. Schmid, and J. Ponce. Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. *CVPR*, 2006.

- [44] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2169–2178. IEEE, 2006.
- [45] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision—ECCV 2014*, pages 740–755. Springer, 2014.
- [46] Greg Little, Lydia B Chilton, Max Goldman, and Robert C Miller. Turkit: tools for iterative tasks on mechanical turk. In *Proceedings of the ACM SIGKDD workshop on human computation*, pages 29–30. ACM, 2009.
- [47] Greg Little, Lydia B Chilton, Max Goldman, and Robert C Miller. Exploring iterative and parallel human computation processes. In *Proceedings of the ACM SIGKDD workshop on human computation*, pages 68–76. ACM, 2010.
- [48] Jingen Liu, Benjamin Kuipers, and Silvio Savarese. Recognizing Human Actions by Attributes. *CVPR*, 2011.
- [49] George Lucas. Star wars: Episode iv-a new hope [film]. *G. Kurtz (Producer), Star Wars. USA: Twentieth Century Fox Film Corporation*, 1977.
- [50] Rebecca Mason and Eugene Charniak. Nonparametric method for data-driven image captioning. *Proceedings of the ACL*, 2014.
- [51] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *IJCV*, 42(3):145–175, 2001.
- [52] A. Oliva and A. Torralba. Scene-Centered Description from Spatial Envelope Properties. *2nd Workshop on Biologically Motivated Computer Vision (BMCV)*, 2002.
- [53] Mark Palatucci, Dean Pomerleau, Geoffrey E Hinton, and Tom M Mitchell. Zero-shot learning with semantic output codes. *NIPS*, 2009.
- [54] Devi Parikh and Kristen Grauman. Interactively Building a Discriminative Vocabulary of Nameable Attributes. *CVPR*, 2011.
- [55] Devi Parikh and Kristen Grauman. Relative Attributes. *ICCV*, 2011.

- [56] Genevieve Patterson and James Hays. Sun attribute database: Discovering, annotating, and recognizing scene attributes. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2751–2758. IEEE, 2012.
- [57] Genevieve Patterson and James Hays. Sun attribute database: Discovering, annotating, and recognizing scene attributes. *CVPR*, 2012.
- [58] Genevieve Patterson, Tsung-Yi Lin, and James Hays. Using humans to build mid-level features. In *Computer Vision and Pattern Recognition (CVPR), Scene Understanding Workshop*, 2013.
- [59] Genevieve Patterson, Grant Van Horn, Serge Belongie, Pietro Perona, and James Hays. Crowd in the loop active learning. In *Neural Information Processing Systems (NIPS), Crowd Workshop*, 2013.
- [60] Genevieve Patterson, Grant Van Horn, Serge Belongie, Pietro Perona, and James Hays. Tropel: Crowdsourcing detectors with minimal training. In *Third AAAI Conference on Human Computation and Crowdsourcing*, 2015.
- [61] Genevieve Patterson, Chen Xu, Hang Su, and James Hays. The sun attribute database: Beyond categories for deeper scene understanding. *IJCV*, 108(1-2):59–81, 2014.
- [62] Pietro Perona. Vision of a visipedia. *Proceedings of the IEEE*, pages 1526–1534, 2010.
- [63] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. *arXiv preprint arXiv:1403.6382*, 2014.
- [64] Marcus Rohrbach, Michael Stark, and Bernt Schiele. Evaluating knowledge transfer and zero-shot learning in a large-scale setting. *CVPR*, 2011.
- [65] Marcus Rohrbach, Michael Stark, György Szarvas, Iryna Gurevych, and Bernt Schiele. What helps where—and why? semantic relatedness for knowledge transfer. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 910–917. IEEE, 2010.
- [66] O. Russakovsky and L. Fei-Fei. Attribute learning in largescale datasets. *ECCV Workshop on Parts and Attributes*, 2010.
- [67] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

- [68] Bryan C Russell, Antonio Torralba, Kevin P Murphy, and William T Freeman. Labelme: a database and web-based tool for image annotation. *ICCV*.
- [69] Evan Sandhaus. The new york times annotated corpus. *Linguistic Data Consortium, Philadelphia*, 6(12):e26752, 2008.
- [70] Burr Settles. Active learning literature survey. *University of Wisconsin, Madison*, 2010.
- [71] Sukrit Shankar, Vikas K Garg, and Roberto Cipolla. Deep-carving: Discovering visual attributes by carving deep neural nets. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3403–3412, 2015.
- [72] Saurabh Singh, Abhinav Gupta, and Alexei A Efros. Unsupervised discovery of mid-level discriminative patches. In *Computer Vision—ECCV 2012*, pages 73–86. Springer, 2012.
- [73] A. Sorokin and D. Forsyth. Utility data annotation with amazon mechanical turk. *First IEEE Workshop on Internet Vision at CVPR*, 2008.
- [74] Yu Su, Moray Allan, and Frdric Jurie. Improving Object Classification using Semantic Attributes. *BMVC*, 2010.
- [75] A. Torralba, R. Fergus, and W. T. Freeman. 80 million tiny images: a large dataset for non-parametric object and scene recognition. *PAMI*, 30(11), 2008.
- [76] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research (JMLR)*, 9(2579-2605):85, 2008.
- [77] Grant Van Horn, Steve Branson, Ryan Farrell, Scott Haber, Jessie Barry, Panos Ipeirotis, Pietro Perona, and Serge Belongie. Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 595–604, 2015.
- [78] Andrea Vedaldi, Siddharth Mahendran, Stavros Tsogkas, Subhansu Maji, Ross Girshick, Juho Kannala, Esa Rahtu, Iasonas Kokkinos, Matthew B Blaschko, David Weiss, et al. Understanding objects in detail with fine-grained attributes. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 3622–3629. IEEE, 2014.

- [79] Sudheendra Vijayanarasimhan and Kristen Grauman. Large-scale live active learning: Training object detectors with crawled data and crowds. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1449–1456. IEEE, 2011.
- [80] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.
- [81] Shuo Wang, Jungseock Joo, Yizhou Wang, and Song-Chun Zhu. Weakly supervised learning for attribute localization in outdoor scenes. *CVPR*, 2013.
- [82] Peter Welinder, Steve Branson, Pietro Perona, and Serge J Belongie. The multidimensional wisdom of crowds. In *Advances in Neural Information Processing Systems*, pages 2424–2432, 2010.
- [83] J. Xiao, J. Hays, K.A. Ehinger, A. Oliva, and A. Torralba. SUN database: Large-scale scene recognition from abbey to zoo. *CVPR*, 2010.
- [84] Jianxiong Xiao, James Hays, Bryan C Russell, Genevieve Patterson, Krista A Ehinger, Antonio Torralba, and Aude Oliva. Basic level scene understanding: categories, attributes and structures. *Frontiers in psychology*, 4, 2013.
- [85] Bangpeng Yao, Xiaoye Jiang, Aditya Khosla, Andy Lai Lin, Leonidas Guibas, and Li Fei-Fei. Human Action Recognition by Learning Bases of Action Attributes and Parts. *ICCV*, 2011.
- [86] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. *NIPS*, 2014.
- [87] Bolei Zhou, Liu Liu, Aude Oliva, and Antonio Torralba. Recognizing city identity via attribute analysis of geo-tagged images. *ECCV*, 2014.