ABSTRACT

Title of dissertation: Recognizing Visual Categories by Commonality and Diversity

Jonghyun Choi, Doctor of Philosophy, 2015

Dissertation directed by: Professor Larry S. Davis
Department of Computer Science,
Department of Electrical and Computer Engineering

Visual categories refer to categories of objects or scenes in the computer vision literature. Building a well-performing classifier for visual categories is challenging as it requires a high level of generalization as the categories have large within class variability. We present several methods to build generalizable classifiers for visual categories by exploiting commonality and diversity of labeled samples and the category definitions to improve category classification accuracy.

First, we describe a method to discover and add unlabeled samples from auxiliary sources to categories of interest for building better classifiers. In the literature, given a pool of unlabeled samples, the samples to be added are usually discovered based on low level visual signatures such as edge statistics or shape or color by an unsupervised or semi-supervised learning framework. This method is inexpensive as it does not require human intervention, but generally does not provide useful information for accuracy improvement as the selected samples are visually similar to the existing set of samples. The samples added by active learning, on the other

hand, provide different visual aspects to categories and contribute to learning a better classifier, but are expensive as they need human labeling. To obtain high quality samples with less annotation cost, we present a method to discover and add samples from unlabeled image pools that are visually diverse but coherent to category definition by using higher level visual aspects, captured by a set of learned attributes. The method significantly improves the classification accuracy over the baselines without human intervention.

Second, we describe now to learn an ensemble of classifiers that captures both commonly shared information and diversity among the training samples. To learn such ensemble classifiers, we first discover discriminative sub-categories of the labeled samples for diversity. We then learn an ensemble of discriminative classifiers with a constraint that minimizes the rank of the stacked matrix of classifiers. The resulting set of classifiers both share the category-wide commonality and preserve diversity of subcategories. The proposed ensemble classifier improves recognition accuracy significantly over the baselines and state-of-the-art subcategory based ensemble classifiers, especially for the challenging categories.

Third, we explore the commonality and diversity of semantic relationships of category definitions to improve classification accuracy in an efficient manner. Specifically, our classification model identifies the most helpful relational semantic queries to discriminatively refine the model by a small amount of semantic feedback in interactive iterations. We improve the classification accuracy on challenging categories that have very small numbers of training samples via transferred knowledge from other related categories that have a lager number of training samples by solving a

semantically constrained transfer learning optimization problem.

Finally, we summarize ideas presented and discuss possible future work.

Recognizing Visual Categories by Commonality and Diversity

by

Jonghyun Choi

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2015

Advisory Committee:
Professor Larry S. Davis, Chair/Advisor
Professor Min Wu (Dean's Representative)
Professor Rama Chellappa
Professor David W. Jacobs
Assistant Professor Ali Farhadi (University of Washington)

Dedication

To my beloved wife, Daehee Lee, and my selfless parents.

Acknowledgments

First and foremost I would like to thank my advisor, Professor Larry S. Davis for many years of generous support and comprehensive guidance. Larry's broad and deep insight of the literature saved me significant amount of time to wander the jungle of the computer vision research and made possible for me to complete this thesis. His acute questions, timeless knowledge and rich sense of humor made him a great advisor and very happy to work with. I deeply appreciate him for giving me almost infinite amount of freedom to explore the ideas and for having extreme amount of patience in listening to the crazy ideas and hardships. He has positively influenced me in almost every corner of my life from the research perspective to a general sense of value. I feel really fortunate to work with him, as an academic advisor as well as a life mentor.

I am also deeply grateful to Professor Ali Farhadi in the University of Washington. His profound insights on the topic and valuable comments have been very important ingredients of this thesis. His deep intuitions helped me to think the problems in various aspects within a very short period of time.

I also would like to thank the committee members for taking time to read the thesis and excuse your busy schedule for the defense. I thank Professor Rama Chellappa for his insightful advice on career and broad knowledge I learned from his class, which forms a cornerstone of my research. I also thank Professor David W. Jacobs for his acute intuitions and friendly discussions on various topics. I am also grateful to Professor Min Wu to help me from the very beginning when I start

the PhD program as my initial academic advisor to the valuable comments in my thesis proposal.

I am also fortunate to work with great mentors throughout numerous internships. I thank to Dr. Shuowen (Sean) Hu and Dr. Susan S. Young in US Army Research Lab for providing me a very supportive and friendly research environment. I also thank to Dr. Hailin Jin, Dr. Zhe Lin and Dr. Jianchao Yang in Adobe Research. I have learned the modern deep learning technique from the theory to the practice in a very short time with their thoughtful guidance in a beautiful bay area's sunshine. I also thank to Dr. Leonid Sigal in Disney Research for his knowledgable comments and friendly cares. Last but not least, I thank to Dr. Sudipta N. Sinha and Krishnan Ramnath in Microsoft Research for their generous support and vibrant discussions on a challenging problem during the amazing time in summer Seattle.

I would also like to thank to my brilliant lab colleagues for their insightful discussions and enjoyable moments in graduate school, Mohammad Rastegari, Abhishek Sharma, William Schwartz, Sameh Khamis, Ruiping Wang, Huimin Guo, Arpit Jain, Vlad I. Morariu, Brandyn White, Zhuolin Jiang, Yang Hu, Stephen Chen, Ejaz Ahmed, Sravanthi Bondugul, Guangxiao Zhang, Yangmuzi Zhang, Joe Ng, Hyungtae Lee, Hyunjong Cho, Vishal Patel, Shobeir Fakhraei, Angjoo Kanazawa, Austin Myers, Raviteja Vermulapalli, Jaishanker Pillai, Kota Hara, Mahyar Najibi, Bharat Singh, Yaming Wang, Yao Yao and Ang Li.

I also am grateful to Prof. Bohyung Han for me to begin the study in Maryland and tons of careful advices for research and life here. And I thank to all my friends for a pleasant and memorable moments in Maryland, Young Wook Kim, Hyun Soo

Kim, Kyowon Kim, Kangmook Lim, Doohyun Sung, Hyun Jung, Sungmin Eum, Eunhui Park, Jaehwan Lee, Jinseong Jeon, Youngsam Park, Uran Oh, Kyunghun Lee, Beomjoon Kwon, Geunmin Ryu, Jounghoon Beh, Inkeun Cho, Shinkyu Park and Takuya Omura.

I owe deepest thanks to my family - my wife, mother, father and family in law. Words cannot express all my gratitude to my wife Daehee for her relentless patience all the time especially during the hard ones. And last, I thank God being with me.

Table of Contents

List of Figures

Chapter 1: Introduction

Recognizing visual categories is an important computer vision problem with high applicability to systems requiring visual perception. Such systems include autonomous vehicle, robots and web services with automatic image tagging, search and management, to name a few. The recognition problem is usually formulated as a generalizable classifier learning problem with state-of-the-art feature descriptors. Building a generalizable classifier for visual category recognition, however, is difficult due to high intra-class variations of visual features because of the high level of visual diversity of samples within a category. In other words, visual categories may contain highly diverse samples in terms of appearance. To address the intra-class variations in learning a classifier, there are two typical solutions. One is to use a large number of labeled samples. The other is to learn a sophisticated classifier. The first solution requires expensive human labeling efforts to obtain quality labeled samples and the second solution often requires high computational cost. Avoiding the issues of expensive human labeling and high computational costs, we present three methods to improve visual category recognition accuracy.

The first method is to add unlabeled samples to categories by learned attributes. Using attributes learned from auxiliary data, we can add high quality

samples without requiring humans in the loop, unlike active learning. Using the attribute representation, we can identify high quality samples without explicitly modeling the sample distribution, unlike semi-supervised learning (SSL). We add samples by two criteria: commonality and specificity of the initially given labeled samples. The samples added by the two criteria are visually diverse but maintaining the characteristics of the category of interest, thus improving classification accuracy.

The second method is to build an ensemble of classifiers for addressing both diverse appearances of subcategories and commonality. With a given set of labeled samples, we discover the subcategories that are discriminative to other categories and learn a set of subcategory classifiers that share the commonality of them. The new ensemble classifier improves accuracy compared to state-of-the-art methods.

The third method is a novel learning framework for visual category categorization by exploiting the commonality and diversity of category definitions with an efficient interactive semantic feedback. In this framework, a discriminative categorization model is improved through iterative semantic feedback. Specifically, the model identifies the most helpful relational semantic queries to discriminatively refine the model. The semantic feedbacks on whether the pattern is valid or not is incorporated back into the model, in the form of regularization, and the process iterates. We validate the proposed model in a few-shot multi-class classification scenario, where we measure classification performance on a set of 'target' classes, with few training instances, by leveraging and transferring knowledge from 'anchor' classes, that contain larger sets of labeled instances.

The dissertation consists of the following chapters. Chapter 2 describes the

method to add unlabeled samples to categories by learned attributes. Chapter 3 describes the method to build an ensemble of classifiers to address the visual variations of samples without adding unlabeled samples. Chapter 4 presents the transfer learning framework with interactive semantics. We then conclude the dissertation with a future plan in Chapter 5.

Chapter 2: Adding Unlabeled Sample to Categories by Learned Attributes

2.1 Introduction

Designing generalizable classifiers for visual categories is an active research area and has led to the development of many sophisticated classifiers in vision and machine learning [70]. Building a good training set with minimal supervision is a core problem in training visual category recognition algorithms [6].

A good training set should span the appearance variability of its category. While the internet provides a nearly boundless set of potentially useful images for training many categories, a challenge is to select the relevant ones – those that help to change the decision boundary of a classifier to be closer to the best achievable. So, given a relatively small initial set of labeled samples from a category, we want to mine a large pool of unlabeled samples to identify *visually different* examples without human intervention.

This problem has been studied by two research communities: active learning and semi-supervised learning. In active learning, the goal is to add visually different samples using human intervention, but to minimize human effort and cost by choos-

ing informative samples for people to label [27, 52, 58]. Even though the amount of human intervention is minimized and its cost is getting cheaper via crowd sourcing, *e.g.*, Amazon Mechanical Turk, it is still preferable to not have humans in the loop because of issues like quality control and time [58].

Semi-supervised learning (SSL) aims at labeling unlabeled images based on their underlying distribution shared with a few labeled samples [19, 59, 73]. In SSL, it is assumed that the unlabeled images that are distributed around the labeled samples are highly likely to be members of the labeled category. However, if we need to dramatically change the decision boundary of a category to achieve good classification performance, it is unlikely that this can be done just by adding samples that are similar in the space in which the original classifier is constructed.

To expand the boundary of a category to an *unseen* region, we propose a method that selects unlabeled samples based on their attributes. The selected unlabeled samples are not always instances from the same category, but they can still improve category recognition accuracy, similar to [31, 41]. We use two types of attributes: category-wide attributes and example-specific attributes. The category-wide attributes find samples that share a large number of discriminative attributes with the preponderance of training data. The example-specific attributes find samples that are highly predictive of the *hard* examples from a category - the ones poorly predicted by a leave one out protocol.

We demonstrate that our augmented training set can significantly improve the recognition accuracy over a very small initial labeled training set, where the unlabeled samples are selected from a very large unlabeled image pool, *e.g.*, ImageNet.

Our contributions are summarized as follows:

1. We show the effectiveness of using attributes learned with auxiliary data to label unlabeled images without annotated attributes.
2. We propose a framework that jointly identifies the unlabeled images and category wide attributes through an optimization that seeks high classification accuracy in both the original feature space and the attribute space.
3. We propose a method to learn example specific attributes with a small sized training set, used with the proposed framework. We then combine the category wide and the example specific attributes to further improve the quality of image selection by diversifying the variations of selected images.

The rest of the chapter is organized as follows: Section 2.2 reviews related work. Section 2.3 presents the overview of our approach. Section 2.4 describes our optimization framework for discovering category wide attributes and the unlabeled images as well as a method to capture exemplar specific attributes. Section 2.5 describes the details of the dataset configurations used in our experiments. Experimental results that demonstrate the effectiveness of our method is presented in Section 3.4. Section 3.5 concludes the chapter.

2.2 Related Work

Our work is related to active learning, semi-supervised learning, transfer learning and recent work about borrowing examples from other categories.

Active Learning The goal of active learning is to add examples with minimal

human supervision [27]. [58] provides a comprehensive survey. Recently, Parkash *et al.* proposed a novel active learning framework based on interactive communication between learners and supervisors (teachers) via attributes [52]. It requires fairly extensive human supervision with rich information.

Semi-Supervised Learning Semi-supervised learning (SSL) adds unlabeled examples to a training set by modeling the distribution of features without supervision. [73] is a detailed review of the SSL literature. Fergus *et al.* proposed a computationally efficient SSL technique for large datasets [19]. Our approach also uses a large dataset and scales linearly in the size of that dataset; it differs from conventional SSL approaches because we do not use the distribution of sample in the original feature space, but in an attribute space. Recently, Shrivastava *et al.* proposed a SSL based scene category recognition framework using attributes, constrained by a category ontology [59]. They leverage the inter-class relationships as constraints for SSL using semantic attributes given by a category ontology as a priori. Our approach is similar to their work in terms of using attributes, but aims to discover attributes without any structured semantic prior.

Transfer Learning and Borrowing Examples Our work is related to recent work on transfer learning [50] and borrowing examples [31, 41, 57].

Ruslan *et al.* [57] proposed building a hierarchical model from categories to borrow images of a useful category for detection and classification. They assume that the images in a category are not diverse and adding all images from some selected category will help to build a better model for the target category. The assumption, however, is bound to be violated by visually diverse categories.

Instead, Lim *et al.* [41] propose a max-margin formulation to borrow some samples from other categories based on a symmetric borrowing constraints.

Kim and Grauman [31] propose a shape sharing method to improve segmentation based on the insight that shapes are often shared between objects of different categories.

Attributes Research on attributes recently has been drawing a lot of attention in the computer vision community because of their robustness to visual variations [17, 35, 37]. Attributes can, in principle, be used to construct models of new objects without training data - zero shot learning [37]. Recently, Rastegari *et al.* [54] propose discovering implicit attributes that are not necessarily semantic for category recognition. The discovered attributes preserve category-specific traits as well as their visual similarity by an iterative algorithm that learns discriminative hyperplanes with max-margin and locality sensitive hashing criteria.

2.3 Approach Overview

Given a handful of labeled training examples per category, it is difficult to build a generalizable visual model of a category even with sophisticated classifiers [70]. To address the lack of variations of the few labeled examples, we expand the visual boundary of a category by adding unlabeled samples based on their attributes. The attribute description allows us to find examples that are visually different but similar in traits or characteristics [17, 35, 37].

Based on recent work on automatic discovery of attributes [54] and large scale

category-labeled image datasets [12], we discover a rich set of attributes. These attributes are leaned using an auxiliary category-labeled dataset to avoid biasing the attribute models towards the few labeled examples. The motivation here is similar to what underlies the successful Classemes representation [63] which achieved good category recognition performance by representing samples by external data that consists of a large number of samples from various categories.

Across the original visual feature space and the attribute space, we propose a framework that jointly selects the unlabeled images to be assigned to each category and the discriminative attribute representations of the categories based on either a category wide or exemplar based ranking criteria. Sec. 2.4.1 presents the optimization framework for category wide addition of unlabeled samples to categories. This adds samples that share many discriminative attributes amongst themselves and the given labeled training data. The same framework can be applied to identify relevant unlabeled samples based on their attribute similarity to specific instances of the training data. This only involves a simple change to one term of the optimization, and is based on how ranks of unlabeled samples change as labeled samples are left out, one at a time, from the attribute based classifier. So, the optimization runs twice - one to identify samples that share large numbers of discriminative attributes within class and a second to find samples that share strong attribute similarity with specific members of the class, and the two sets of samples are then combined to create the augmented training set for the class. We refer to the first as a categorical analysis and the second as an exemplar analysis.

2.4 Joint Discovery of Discriminative Attributes and Unlabeled Samples

2.4.1 Categorical Analysis

We simultaneously discover discriminative attributes and images from the unlabeled data set in a joint optimization framework formulated in both visual feature space and attribute space with a max margin criterion for discriminativity. Unlike [59], we do not require a label taxonomy to find the shared properties. Also unlike [41], we do not need to learn the distributions of the unlabeled images in the original feature space.

For each category c, we will construct a classifier in visual feature space, w_c^v, using the set $X = \{x_i | i \in \{1, \ldots, l, l+1, \ldots, n\}\}$ that consists of the initially given labeled training images $\{x_i | i \in \{1, \ldots, l\}\} \subset X$ and the selected images from the unlabeled image pool $\{x_i | i \in \{l+1, \ldots, n\}\} \subset X$. The subset of images from the unlabeled set is assigned to a category based on identifying discriminative attribute models. Since the problems of determining the discriminative attributes and selecting the subset of unlabeled data to assign to a category are coupled, we learn them jointly. Additionally, we want to mitigate against unlabeled samples being assigned to multiple categories, so a term $M(\cdot)$ is added to the optimization

criteria to enforce that. The joint optimization function is:

$$\min_{I_c \in I, w_c^v, w_c^a} \sum_c \Big(\alpha J_c^v(I_c, w_c^v) + \beta J_c^a(I_c, w_c^a) \Big) + M(I)$$

subject to

$$\begin{aligned}
& J_c^v(I_c, w_c^v) = \|w_c^v\|_2^2 + \lambda_v \sum_{i=1}^{n} \xi_{c,i} \\
& I_{c,i} \cdot y_{c,i}(w_c^v x_i) \geq 1 - \xi_{c,i}, \quad \forall i \in \{1, \ldots, n\} \\
& J_c^a(I_c, w_c^a) = \|w_c^a\|_2^2 + \lambda_a \sum_{j=1}^{n} \zeta_{c,j} - \sum_{k=l+1}^{n} I_{c,k} \Big(w_c^a \phi(x_k) \Big) \\
& I_{c,j} \cdot y_{c,j}(w_c^a \phi(x_j)) \geq 1 - \zeta_{c,j}, \quad \forall j \in \{1, \ldots, n\} \\
& \sum_{k=l+1}^{n} I_{c,k} \leq \gamma, \quad I_{c,k} = 1, \ \forall k \in \{1, \ldots, l\} \\
& M(I) = \sum_{c1 \neq c2} \sum I_{c1} \cdot I_{c2},
\end{aligned} \tag{2.1}$$

$I_c \in \{0, 1\}$ is the sample selection vector for category c, and indicates which unlabeled samples are selected for assignment to the training set of category c. $I_{c,i} = 1$ when the i^{th} sample is selected for category c. $x_i \in \mathbb{R}^D$ is the visual feature vector of image i. $y_{c,i} \in \{+1, -1\}$ indicates whether the label assigned to x_i is c $(+1)$ or not (-1). $\phi(\cdot) : \mathbb{R}^D \to \mathbb{R}^A$ is a mapping function of visual feature to the attribute space that is learned from auxiliary data, where $\mathbb{R}^D$ and $\mathbb{R}^A$ denote visual feature space and attribute space, respectively. α and β are hyper-parameters for balancing the max margin objective terms for both the visual feature and attribute based classifiers. γ is a hyper-parameter for specifying the number of selected images.

$J_c^v(I_c, w_c^v)$ and the second constraint of Eq. 2.1 are a max-margin classification terms in visual feature space. $J_c^a(\cdot)$ and the forth constraint of Eq. 2.1 are a max-margin classifier in the attribute space (T_A) with a selection criterion (T_R); we divide

it as follows:

$$J_c^a(I_c, w_c^a) = \underbrace{\|w_c^a\|_2^2 + \sum_{j=1}^{n} \zeta_{c,j}}_{T_A} \underbrace{- \sum_{k=l+1}^{n} I_{c,k}\Big(w_c^a \phi(x_k)\Big)}_{T_R}. \tag{2.2}$$

T_R essentially chooses the top γ responses of the attribute classifier from the unlabeled set by the fifth constraint of Eq. 2.1. The term $M(I_c)$ penalizes adding the same sample to multiple categories (sixth constraint of Eq. 2.1).

The objective function is obviously not convex due to the interconnection of the two spaces by the example selecting indicator vector I and the attribute mapper $\phi(\cdot)$. However, if the I_c's were known and we fix either $J_c^v(I_c, w_c^v)$ or $J_c^a(I_c, w_c^a)$, the function becomes convex and can be solved with an iterative block coordinate descent algorithm. At each iteration we fix one of the terms and the entire objective function becomes an ordinary max margin classification formulation with a selection criterion. Each iteration of the block coordinate descent algorithm updates the set of indicator vectors I. At the first iteration, the initial value of I is determined by training the attribute classifier w_c^a on the given labeled training set. Then, after the two SVM's in both spaces are updated, we update I. Since there is no proof of convergence for the algorithm, we iterate it a fixed number of times - $1 \sim 3$ in practice. The iterations could be controlled using a held out validation set, but since our premise is that labeled samples are rare we do not do that.

2.4.2 Exemplar Analysis

The discriminative attributes learned in Sec. 2.4.1 capture commonality among all examples in a category. We refer them as *categorical attributes*. Each example,

however, has its own characteristics that may help to expand the visual space of the category by identifying images based on example-specific characteristics. To discover *exemplar attributes*, a straightforward solution would be to learn exemplar-SVMs [45]. The exemplar-SVM, however, requires many negative samples to make the classifier output stable. For our purposes, though, we can accomplish the same thing by analyzing how the ranks of unlabeled samples change when a single sample is eliminated from the training set of the attribute SVM. If an unlabeled sample sees its rank drop sharply from its rank in the full-sample SVM, then the training sample dropped should have strong attribute similarity to the unlabeled sample.

This is illustrated in Figure 2.1. The first row shows the labeled training samples (10 examples). The left most column is a list of unlabeled images ordered by confidence score by w_c^a. Rest of the columns are lists of unlabeled images ordered by each $w_{c,\bar{i}}^a$'s. Note that an image of halved orange in the second column makes the first ranked images in the left most column (by w_c^a) go down because the halved orange was removed in the training set of $w_{c,\bar{i}}^a$. Eliminating the half orange (second sample, top row) from the training set reduces the rank of the globally best unlabeled sample from 1 to 10.

First, let w_c^a be the attribute classifier for the current training set for category c (while the process is initialized based on the labeled training set, after each iteration we use the additional unlabeled samples added to the category to construct a new attribute classifier). Let $w_{c,\bar{j}}^a$ be the attribute classifier learned when the i^{th} sample is removed from the training set. We next describe how we use the ranks of unlabeled samples in these two classifiers to modify T_R in Eq. 2.2. Basically, we are going

to re-rank the unlabeled samples based on their rank changes from w_c^a to $w_{c,\bar{j}}^a$. We want samples whose ranks are lowered dramatically by the elimination of a single sample from the training set to be highly ranked by the re-ranking function. This can be accomplished by computing the following score based on rank changes, and the sorting the unlabeled samples by this score:

$$e_j(x_i) = \frac{\mu}{r_g(x_i)} - \frac{\nu}{r_j(x_i)}, \tag{2.3}$$

where x_i is a sample from the an unlabeled pool, $r_g(\cdot)$ and $r_j(\cdot)$ are the rank functions of w_c^a and $w_{c,\bar{j}}^a$ respectively. μ and ν are the balancing hyper-parameters for two ranks. T_R is then simply determined by first selecting the new top ranked sample from each leave one out SVM, then the second ranked, until a fixed number of samples are selected (skipping over duplicates). This set is then used to re-learn the feature and attribute based SVM's and the entire process iterates.

2.5 Dataset

We construct a dataset from a large scale dataset for category recognition, ImageNet [12] using its standard benchmark subset, ILSVRC 2010 dataset. We will publicly release our dataset for future comparison.[1] It consists of approximately 1 million images of 1,000 categories. The images are downloaded from a photo sharing portal[2]. It provides fine grained category labels such as specific breed of dogs, *e.g.*, Yorkshire Terrier and Australian Terrier.

[1] http://umiacs.umd.edu/∼jhchoi/addingbyattr/

[2] http://www.flickr.com

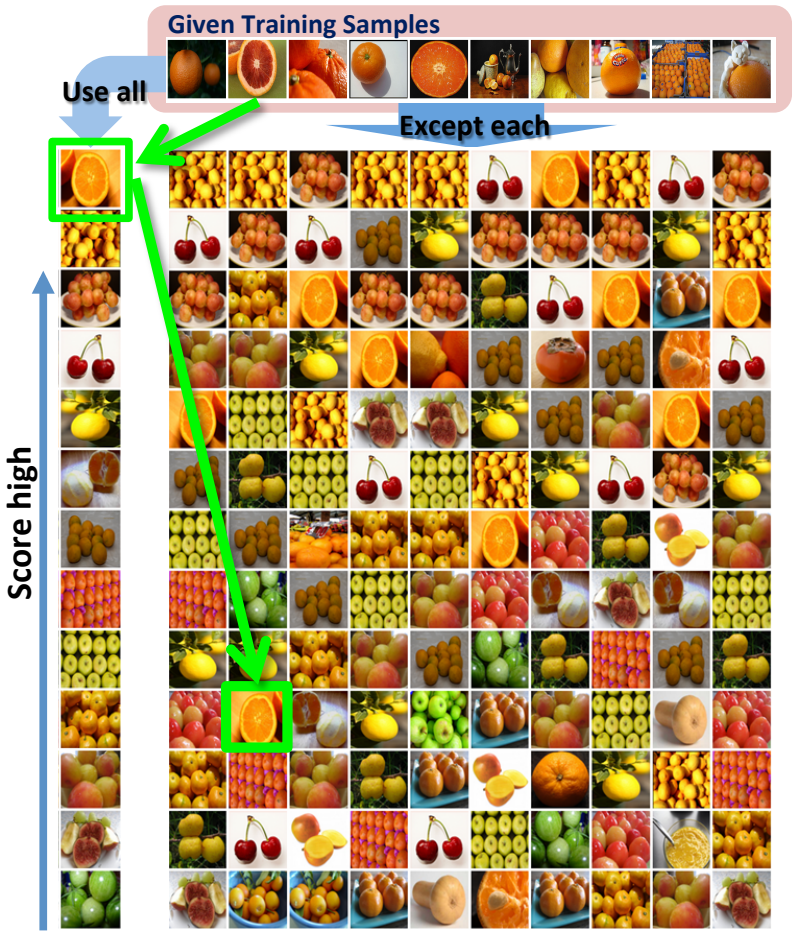


Figure 2.1: Unlabeled images ordered by confidence score by w_c^a and a set of $w_{c,\bar{i}}^a$'s (column wise).

We randomly choose 11 categories among natural objects such as vegetable and dogs as the categories of interest. Those categories have very large appearance variations due to factors including non-rigid deformation, lighting, camera angle, intra-class appearance variability *etc.*. For each category, we randomly choose ten images as an initial labeled training set and 500 images as a testing set. The unlabeled image pool consists of images that are arbitrarily chosen from the entire 1,000 categories in the ILSVRC 2010 benchmark dataset, but includes at least 50

samples from each of the categories to be learned. The size of the image pool varies in the experiments but is much larger (from 5,000 to 50,000) than the initial training set. For learning the attribute space and the mapper, it is expected that the attribute mapper should capture some attribute of the categories of interest. For this purpose, we use 50 labeled samples from 93 categories that are similar to the 11 categories to learn the attribute space.

2.6 Experiments

The main goal of our method is to add unlabeled images to the initial training set in order to classify more test images correctly. We demonstrate the effectiveness of our method by improvements in average precision (AP) of category recognition. We also evaluate our approach under various scenarios including the precision of the unlabeled image pool and the size of the learned attribute space and also the effect of parameters including number of selected examples. Moreover, we evaluate the effect of selecting images that are not from the category of interest.

2.6.1 Experimental Setup

Visual feature descriptors: We use various visual feature descriptors including HOG, GIST and color histograms. Since the feature dimensionality is prohibitively large, we reduce the dimension to 6,416 by PCA.

Attribute discovery: We use the binary attribute discovery method of Rastegari *et al.* [54] as the attribute mapping function, $\phi(\cdot)$ in Eq. 2.1. We learn the mapper

with default hyper-parameter sets as suggested in [54]. We use 400 bits in most of our experiments. We also present performance as a function of the number of bits.

Max margin optimization: We use LibLinear [16] for training all max-margin based objective functions. To address the non-linearity of visual feature space, we use homogeneous kernel mapping [65] on the original features with the linear classifier. For the hinge loss penalty hyper-parameter, we use 0.1.

Parameters: For the parameter in Eq. 2.1, we use $\alpha = 1, \beta = 1$. For categorical attribute only, we mostly use $\gamma = 50$ except ones in Section 2.6.4. For combining exemplar and categorical attributes, we mostly use $\gamma = 20$ and $\gamma_i = 3$ except for Section 2.6.4. We investigate algorithm performance as a function of γ in Section 2.6.4. For the parameters of the scoring function for exemplar-attributes in Eq. 2.3, we use $\mu = 1$ and $\nu = 1$.

2.6.2 Qualitative Results

Our method discovers examples that expand the visual coverage of a category by not only adding the examples from the same category but also examples from other categories. Figure 2.2 illustrates qualitative results on the category *Dalmatian* for both categorical and exemplar attributes analyses. The selected examples based on categorical attributes exhibit characteristics commonly found in the labeled examples such as dotted, four legged animal. The exemplar attributes, on the other hand, select examples that exhibit the characteristic of individual labeled training examples.

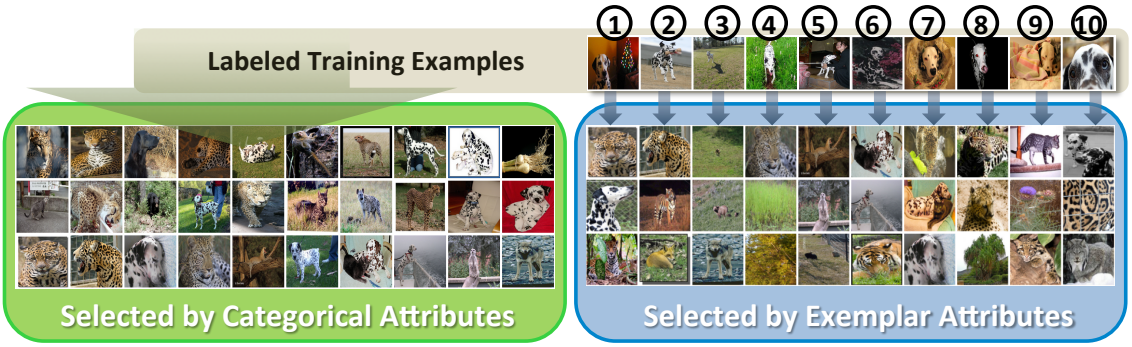


Figure 2.2: Qualitative results of our method. Note that the selected examples by categorical attributes display characteristics commonly found in the labeled training examples such as 'dotted', 'four legged animal'. In contrast, the exemplar attributes select the examples that display the characteristic of individual example.

2.6.3 Comparison with Other Selection Criteria

Given our goal of selecting examples from a large unlabeled data with only a small number of labeled training samples, we do not compare with semi-supervised learning methods because they need more labeled data to model the distribution. Since our method does not involve human intervention, we do not compare to active learning.

We compare to baseline algorithms which are applicable to the large unlabeled data scenario. The first baseline algorithm is to select nearest neighbors. The second baseline selects images by an active criterion that finds examples close to a learned decision hyperplanes [27]. Both baseline algorithms selects images based on analysis in the visual feature space. We summarize the comparison in the Table 2.1. 'Init.' refers to initial labeled training set. 'NN' refers to addition by 'nearest neighbor'

in visual feature space, 'ALC' refers to addition by 'active learning criteria (ALC)' that finds the examples close to the current decision hyperplanes [27]. 'Cat.' refers to our method of select examples using categorical attributes only. 'E+C' refers to addition using categorical and exemplar attributes. The size of the unlabeled dataset is roughly 3,000 from randomly chosen categories out of 1,000 categories.

As shown in Table. 2.1, the two baseline strategies decrease mean average precision (mAP). However, our method identifies useful images in the unlabeled image pool and significantly improves mAP by 7.64%. Except for the category *Greyhound*, we obtain performance gain from 2.77% - 16.36% in all categories. The added examples serve not only as positive samples for each category but also as negative samples for other categories. The quality of the selected set can change the mAP significantly in both ways.

2.6.4 Number of Selected Examples

As we select more examples, controlled by γ in Eq. 2.1, the chances of both selecting useful images and harmful images for a category increase simultaneously. We vary the number of selected examples and observe mean average precision as shown in Figure 2.3. The category wide attributes identify useful unlabeled images. In addition, the exemplar attributes further improve the recognition accuracy.

Category Name	Init.	NN	ALC	Cat.	E+C
Mashed Potato	45.03	34.02	51.15	61.39	63.92
Orange	29.84	16.29	26.97	40.61	41.05
Lemon	32.21	27.58	32.43	35.37	34.23
Green Onion	25.06	16.50	19.66	38.57	40.20
Acorn	13.09	11.05	15.41	19.35	20.10
Coffee bean	58.29	43.89	56.62	64.65	66.54
Golden Retriever	14.54	15.57	12.61	17.54	18.61
Yorkshire Terrier	29.62	13.62	27.63	41.41	45.65
Greyhound	15.24	15.73	15.64	14.75	15.22
Dalmatian	43.84	27.97	37.91	54.42	57.23
Miniature Poodle	26.10	12.50	21.16	28.87	30.21
Average	30.26	21.34	28.84	37.90	39.36

Table 2.1: Comparison of average precision (AP) (%) for each category with 50 added examples by various methods.

2.6.5 Adding Examples from Similar Categories

Among the selected images per category, some examples are true instance of the category. We refer to these as *exact examples* and the rest as *similar examples*. We are interested in how much the similar examples improve category recognition. First, we examine the purity of the selected set in Figure 2.4. In the figure, red bars denote the purity of selected images using category wide attributes only (+ by C

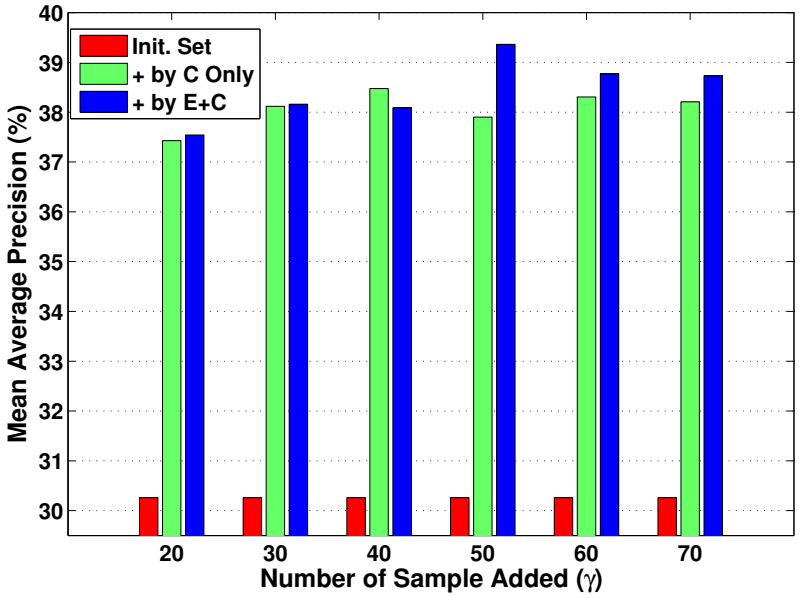


Figure 2.3: Mean average precision (mAP) of 11 category by our method varying the number of unlabeled images selected. The red, green and blue are the mAP using the initial labeled set (Init. Set), the augmented set by our method using category wide attributes only (+ by C only) and categorical+exemplar attributes respectively. (+ by E+C)

only) and the green bars are obtained from categorical+exemplar attributes (+ by E+C). The purity is the percentage of exact samples in the set. Surprisingly, even though the purity values seem low, they still improve classification performance.

We now investigate how much the similar examples improve the average precision (AP) by removing the exact examples from the selected set. The blue bars in Figure 2.5 represent the AP using just the similar examples. It is interesting to note that using only the similar examples still improves the APs over the initial labeled set.

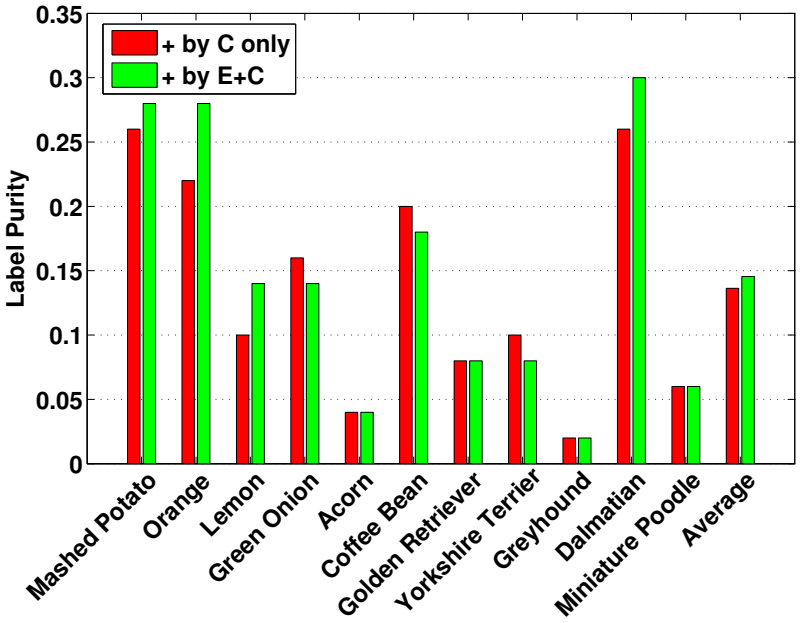


Figure 2.4: Purity of added examples.

In addition, it is also interesting to observe how the performance changes when we add the same number of similar examples as the size of the initially selected image set (50). This is shown as green bars in Figure 2.5. In the figure, the navy colored bars are obtained using the initial labeled set (baseline). The blue bars use only similar examples among the selected 50 examples. The green bars use 50 similar examples to compare with the result of our selected 50 examples (orange bars) including both similar and exact examples. The red bars are obtained using a set of 50 ground truth images, which is the best achievable accuracy (upper bound). Even the similar examples alone improve the category recognition accuracy compared to just using the initial labeled set. All results are obtained using categorical attributes only. (The results using both exemplar and categorical attributes are similar so are omitted).

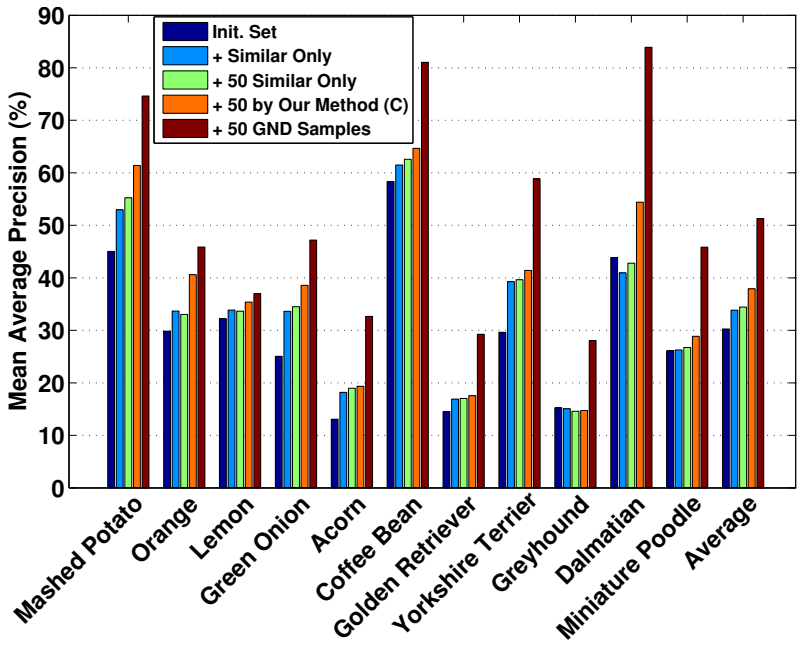


Figure 2.5: Mean average precision (mAP) as a function of the purity of the selected examples.

2.6.6 Precision of Unlabeled Data

The unlabeled data can be composed of images from many categories. The precision of the unlabeled data is defined as the ratio of size of the unlabeled images from extraneous categories to the size of the entire unlabeled image data. The larger the unlabeled data, the lower we expect its precision to be (imagine running a text based image search using the category name and accepting the first k images returned). It is interesting to observe how robust our method is against the precision of unlabeled data.

We start with an unlabeled set (550 images, 50 from each of the 11 categories) of precision 1.0, and reduce precision by adding images from other categories. The

number of the unrelated images ranges from 2,500 to 50,000, which are randomly chosen from the entire 1,000 categories of the ImageNet ILSVRC 2010 dataset.

As shown in Figure 2.6, we observe that the accuracy improvement by our method using categorical attributes is quite stable even when precision is low.

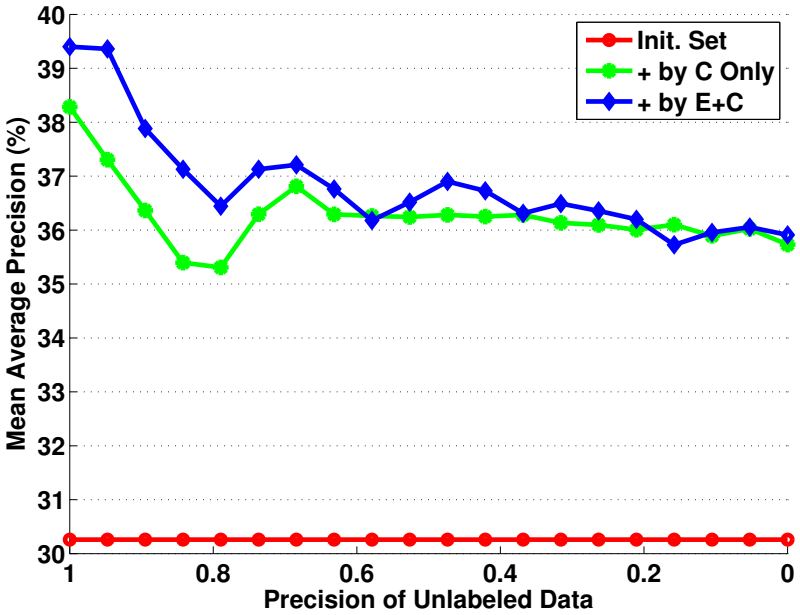


Figure 2.6: Mean average precision (mAP) as a function of precision of unlabeled data. Precision denotes the ratio of size of the unlabeled images from extraneous categories to the size of the entire unlabeled image data (size = 50,000). Although precision decreases, the mean average precisions (mAP) by our method do not decrease much.

2.6.7 Size of Initial Labeled Set

We next explore how the size of the initial labeled set effects accuracy. We systematically vary the size from 5 to 50 and show mAP compared to an SVM

learned on the initial training set - see Figure 2.7. The mAP gain for the smallest initial labeled set (5) is the highest as expected. When the number of samples is larger than 25, our method (+ by C only) does not improve the mAP much, although it still improves by $1.18 - 2.74\%$. Interestingly when there are many samples in the initial training set (*e.g.*, more than 25), the exemplar traits begin to reduce the mAP.

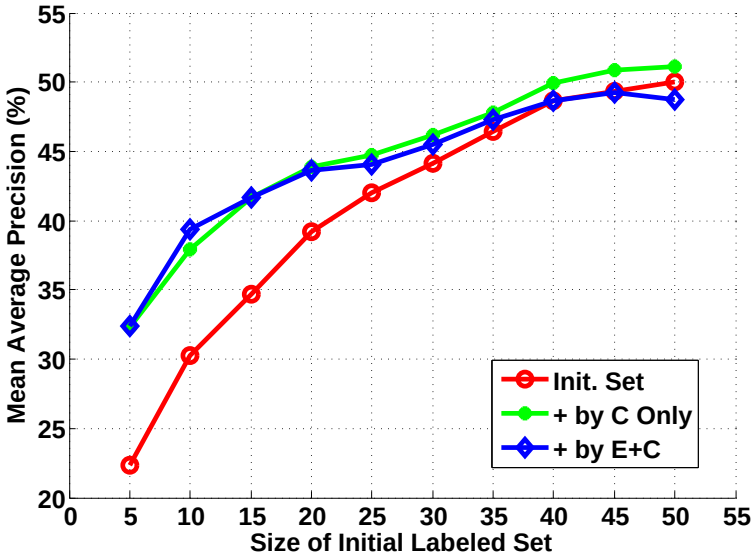


Figure 2.7: Mean average precision (mAP) as a function of the size of the initial labeled set. The number of added samples is 50 in all experiments.

2.6.8 Comparison to Exemplar SVM

We also compare the effectiveness of our proposed exemplar attributes discovery method (Sec. 2.4.2) to a conventional exemplar SVM [45]. It is straightforward to integrate the exemplar SVM into our formulation (Eq. 2.1): by setting label

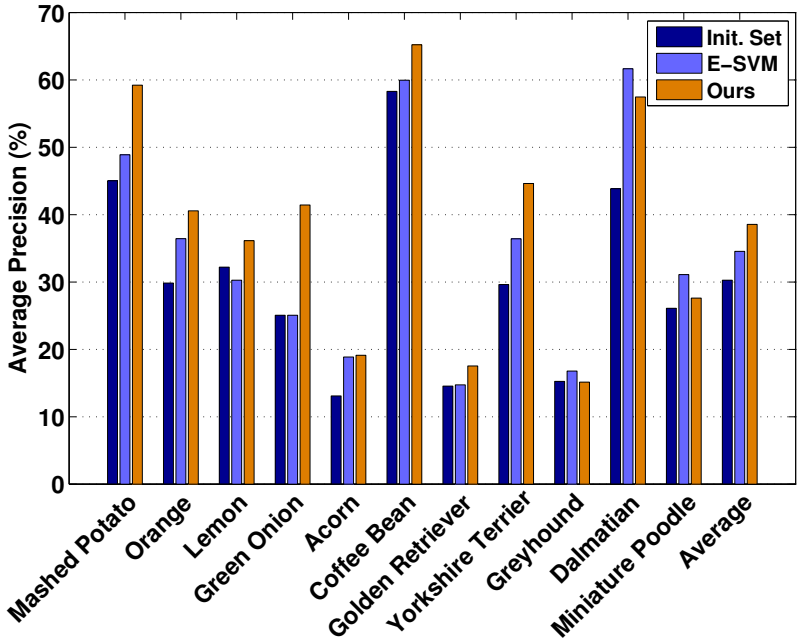


Figure 2.8: Comparison of our exemplar attribute discovery method (Sec. 2.4.2) to exemplar SVM. Our method outperforms the exemplar SVM in terms of category recognition accuracy by APs without the extra large negative example set (size = 50,000).

$y_{c,j}$ to 1 for the j^{th} example, the label coresponding to the examples in the same category to 0 and the rest to 1. To stabilize the exemplar SVM scores, we employ 50,000 external negative samples to learn each exemplar SVM while we use the small original training set for our method. Figure 2.8 shows that our exemplar attribute discovery method outperforms the exemplar SVM by large margins even without the large negative example set.

2.7 Conclusion

We proposed a method to select unlabeled images to learn classifiers based on learned attributes. The unlabeled images selected by our method do not necessarily belong to the category of interest but are similar in attributes. Our method does not require any annotated attribute set a priori but first builds an automatically learned attribute space. We formulate a joint optimization framework to select both images and the attributes for a category and solve it iteratively. In addition to the category wide attributes, we identify example specific attributes to diversify the selected images. For addressing the problem of small size training set to learn the example specific attributes, we propose a method that can be intuitively regarded as an inverse of exemplar SVM.

From a large unlabeled data pool, the selected images improve category recognition accuracy significantly over accuracy obtained using the initial labeled training set.

Chapter 3: Sharing Subcategory Commonality for Learning Generalizable Classifiers

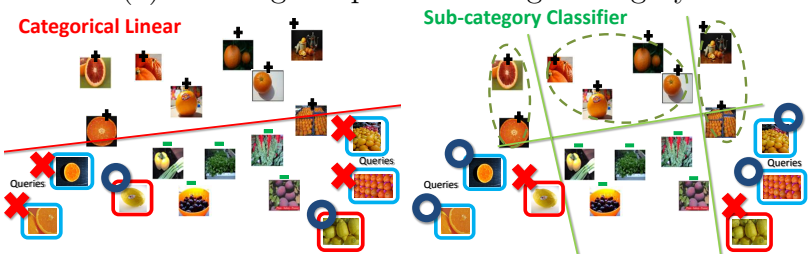
3.1 Introduction

Classifier generalization is an important goal in visual recognition. It is achieved by either using sufficient labeled training data or enforcing prior knowledge as a regularizer to prevent overfitting to the given training samples [23]. The regularizers include geometric properties (*e.g.*, max-margin in Support Vector Machines (SVM) [9] and convex hull in Power-SVM [70]), shrinkage (*e.g.*, ridge regression) and sparsity (*e.g.*, sparse coding [23]).

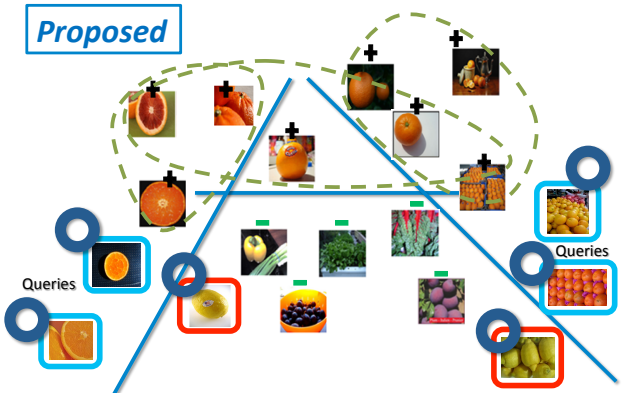
Intra-class variation of visual categories is often high and leads to multi-modal distributions of training samples [20,45]. Even with a good regularization prior, it is challenging to learn a single generalizable visual category classifier. Fig. 3.1 shows an illustrative example of classifying an 'orange' category. Images with + or − sign are the positive/negative training samples respectively. Blue rectangles denotes test samples labeled as orange. Red rectangles denotes test samples labeled as not orange. Navy blue **O** denotes correct classification. Red **X** denotes misclassification. Proposed method learns a set of classifiers on discovered discriminative overlapping

subcategories. The classifiers are forced to be similar by minimizing the rank of the subcategory classifier matrix to share commonalities amongst subcategories.

(a) Training samples of 'Orange' category



(b) A linear classifier (c) Subcategory classifiers



(d) Proposed

Figure 3.1: Classification of 'Orange' category by Various Approaches.

The visual appearances of 'orange' are diverse: a half-cut shape, close-up and zoomed-out as shown in Fig. 3.1-(a). In feature space, orange images would form multiple modes, as depicted in Fig. 3.1-(b)∼(d). A single classifier for the 'orange' category might miss visually inconsistent variations (*e.g.*, half-cut and whole shape) in Fig. 3.1-(b). The ensemble classification approaches [18, 24, 45] address this problem by introducing diversity into the classification model by dividing the

category into several subcategories and learning subcategory classifiers. However, the subcategories only have partial information of a category. The information insufficiency can lead each subcategory classifier to incorrectly classify test samples as marked with red X in Fig. 3.1-(c). Recently, Zhu *et al.* proposed learning overlapping subcategory classifiers to overcome information insufficiency [74]. We also formulate an objective function to learn an ensemble of subcategory classifiers that share information among the subcategories.

In learning such an ensemble to improve classification accuracy, there are two problems to address; 1) identifying a good set of subcategories for better classification and 2) learning a set of subcategory classifiers for better classification.

For discovering subcategories, one may use clustering algorithms (*e.g.*, k-means [44]) or use discriminative subcategorization method (DSC) [24], or treat each sample as a subcategory as an extreme case as in Exemplar-SVM (E-SVM) [45]. The first two of these methods were proposed to obtain visually distinctive clusters but not to improve classification. Instead, we discover a set of subcategories that are good for classification. To do that, we first learn a classifier for each possible subcategory to build a space of subcategory classifiers. Then we explore the subcategory classifier space to select the few subcategories that are most discriminative. The subcategories found are not only chosen for better classification but are also encouraged to overlap, which improves classification accuracy [74].

Once we discover the subcategories, we jointly learn a set of discriminative subcategory classifiers that exploit what is shared among subcategories. The resulting ensemble classifiers maintain not only individual subcategory specificity but

also commonalities amongst the subcategories. One of the advantages of our method is that we do not need to specify or know the optimal number of subcategories for classification. The joint learning of subcategory classifiers is analogous to multi-task learning where each 'task' is learning a subcategory classifier. So, we refer to our approach as *'Multi-Subcategory Learning'*.

We compare our method with baselines and state-of-the-art single category-wide classifiers and subcategory based ensemble classifiers. Our method outperforms other methods by a large margin on three popular visual recognition datasets; Pascal VOC 2007, ImageNet-20 and Caltech-256.

3.2 Related Work

Subcategory Based Methods: To address the intra-class variation of visual category recognition, a number of subcategory based approaches have been proposed [18, 24, 45, 74]. At one extreme, each example can be a subcategory. Malisiewicz *et al.* proposed an ensemble of Exemplar-SVM approach to obtain example-specific characteristics for better detection accuracy [45]. Since each Exemplar-SVM uses only one positive sample, it requires a very large number of negative samples to obtain a good classifier [45]. Recently, Choi *et al.* proposed a way of obtaining exemplar traits by comparing 'leave-one-out' classifiers and 'full' classifiers without requiring a large negative sample set [11].

Building a subcategory classifier for better category-wide classification can be viewed as learning part models for a better classification [18] but in feature space. In

this sense, the Latent-SVM (LSVM) can be used to build subcategory models [24]. Also, Hoai and Zisserman formulated a discriminative subcategory clustering objective function for discovering 'pure' and 'disjoint' subcategories for good clustering performance [24]. Recently, Zhu *et al.* claimed that subcategory overlap is important for better accuracy and proposed an iterative algorithm to assign samples to multiple subcategories for obtaining overlapping subcategories [74]. We also discover overlapping subcategories but in a principled way by formulating an objective function that enforces the subcategory classifiers to share commonality amongst themselves while being discriminative with respect to other categories.

Ensemble Methods: The subcategory based method is an ensemble classification approach that uses multiple classification models to obtain better predictive performance [55]. Existing methods include 'cross-validation committee', 'bagging' and 'boosting' [55]. Each classifier in the ensemble captures traits of a random subset of the training set. The ensemble of classifiers is usually learned independently. Matikainen *et al.* modified AdaBoost to learn a good ensemble of classifiers for recommendation tasks [47]. In recent work, including Multiple-Instance SVM [3] and Latent-SVM [18], ensemble classifiers are learned jointly. We also want to learn an ensemble of classifiers jointly, but force them to share commonality.

Multi-task Learning: Multi-task learning addresses simultaneous learning of multiple prediction tasks that are related to each other [10]. In multi-class classification, classifiers for each class are learned simultaneously by the low-rank embedding of the multi-class structure [2,4,42,43]. Loeff and Farhadi proposed a multi-task learning based scene recognition method [42]. Specifically, they enforced the scene

classifier to share or discover latent structures of scene labels using trace-norm minimization [2]. Bergamo *et al.* proposed a meta-class feature to merge categories based on their visual similarity for learning more discriminative multi-class classifiers [7]. Recently, Harchaoui *et al.* proposed a scalable approach to learn a set of classifiers through a single objective function with trace-norm regularization for multi-class classification [22].

We extend the idea to learning the classifiers for a *single* category for better generalization by utilizing the intrinsic commonalities of subcategories. In other words, our method leverages shared information among the subcategories. Sharing the same motivation, Weston and Blitzer proposed a method to improve ranking quality for a single task by leveraging the structure of already ranked queries by matrix low rank parameterization [67]. Yang *et al.* proposed a ν-SVM based multi-task learning framework for one-class classification for small number of training set scenario [69].

3.3 Multi-Subcategory Learning

The goal of Multi-Subcategory Learning is two-fold. One is to discover visual subcategories that are directly useful in learning a discriminative ensemble classifier. The other is to learn subcategory classifiers that capture both the subcategory traits and their commonalities. Specifically, we want to jointly learn the subcategory classifiers that are not only discriminative, but also share common information among them. Unlike the previous approach for sharing information by overlap-

ping samples [74], we formulate a discriminative learning framework to enforce the information sharing.

3.3.1 Formulation

We are given labeled training samples for a category $S = (x_i, y_i) \in \mathbb{R}^D \times \{1, -1\},\ i \in \{1, \ldots, N\}$ where D is the feature dimension, N is the total number of samples. Positive sample set is denoted by $X_p = \{x_i | y_i = 1\}$ and the negative sample set is denoted by $X_n = \{x_i | y_i = -1\}$.

First, we discover a set of subcategories that are discriminative to other categories. As a direct way of finding a good set of subcategories for classification, we learn subcategory classifiers and choose a few that are discriminative with respect to other categories. In this way, we naturally allow the subcategories to overlap, as is known to be beneficial for better classification [74].

Specifically, we first construct M candidate subcategories; the candidate set of subcategories can be obtained by various methods including exhaustive bagging [55] that considers all possible subsets of the positive training set of a category (2^N) or exhaustive bagging on clusters of the positive training set obtained by any clustering methods, *e.g.*, k-means. This is to reduce the number of candidate subcategories that are exponentially large. For each candidate subcategory, we learn a classifier, $w_m \in \mathbb{R}^D$, discriminative to other categories (but not to other subcategories). We then stack the subcategory classifiers into a matrix, $W = [w_1 \cdots w_M] \in \mathbb{R}^{D \times M}$ and choose a *small number* of subcategories that are most discriminative. We formulate

this by an optimization of a selector vector, I, with a sparsity regularizer and a discriminative loss term as:

$$\begin{aligned} &\min_{I} \ \|I\|_1 + \alpha \sum_{i=1}^{N} \xi_i \\ &\text{subject to} \\ &y_i I^T W^T x_i \geq 1 - \xi_i, \quad \forall i \in \{1, \cdots, N\}, \end{aligned} \tag{3.1}$$

where $I \in \mathbb{B}^M$ is a M-dimensional binary indicator vector (chosen 1, not chosen 0), α is a balancing parameter between the sum of slack variables for the loss term and the sparsity constraint. Small α results in fewer subcategories. Optimization details are presented in the following subsection.

Suppose we discover K subcategories by solving Eq.(3.1); we now learn the subcategory classifiers jointly by enforcing information sharing of subcategory classifiers, $w_k \in \mathbb{R}^D$, where $k \in \{1, \ldots, K\}$. We enforce information sharing by minimizing the rank of the sub-matrix of W denoted by $\widetilde{W} = [w_1 \cdots w_K] \in \mathbb{R}^{D \times K}$ which consists of the selected classifiers. Along with a discriminative loss term, minimizing the rank of $\widetilde{W}$ enforces the classifiers to be similar so that the classifiers share information while they are discriminatively learned. The optimization objective function is as follow:

$$\begin{aligned} &\min_{\widetilde{W}} \ \operatorname{rank}(\widetilde{W}) + \beta \sum_{i=1}^{N} \sum_{k=1}^{K} \zeta_{i,k} \\ &\text{subject to} \\ &Y(i,k) w_k^T x_i \geq 1 - \zeta_{i,k}, \quad \forall i \in \{1, \cdots, N\}, \\ &\qquad\qquad\qquad\qquad\qquad \forall k \in \{1, \cdots, K\}, \end{aligned} \tag{3.2}$$

where β is a balancing parameter between the sum of slack variables for the loss

function and the rank of a matrix, and $Y \in \mathbb{R}^{N \times K}$ is a matrix of subcategory membership labels. β determines the amount of information sharing. $Y(i,k)$ is the subcategory membership label of the i^{th} sample for the k^{th} subcategory (associated with w_k), that is:

$$Y(i,k) = \begin{cases} 1 & \text{if } x_i \in X_{p,k}, \\ 0 & \text{if } x_i \notin X_{p,k}, x_i \in X_p \\ -1 & \text{if } x_i \in X_n, \end{cases} \tag{3.3}$$

where $X_{p,k}$ denotes the set of positive samples that belong to the k^{th} subcategory.

3.3.2 Optimization

Optimizing Eq.(3.1) and Eq.(3.2) is not trivial due to the integer variable I and rank term, respectively. We rewrite the equations to efficiently solve them using popular relaxations.

3.3.2.1 Discovering Subcategories

Solving Eq.(3.1) is a combinatorial optimization problem since the indicator variable I is discrete. To solve it efficiently, we relax the objective function by replacing $I \in \mathbb{B}^M$ with $\tilde{I} \in \mathbb{R}^M$ and binarize $\tilde{I}$ to obtain I as follows:

$$I(m) = \begin{cases} 1, & \text{if } \tilde{I}(m) \neq 0, \\ 0, & \text{if } \tilde{I}(m) = 0. \end{cases} \tag{3.4}$$

Using this relaxation, Eq.(3.1) becomes an L_1-SVM objective function as:

$$\min_{\tilde{I}} \ \|\tilde{I}\|_1 + \alpha \sum_{i=1}^{N} h(y_i \tilde{I}^T W^T x_i), \tag{3.5}$$

where $h(\cdot)$ is the hinge loss function. It can be solved by a stochastic gradient descent algorithm in its primal form as in [16].

Eq.(3.5) finds subcategories (associated with the subcategory classifiers) that are discriminative to other categories. The discovered subcategories can overlap as long as they are discriminative. The overlapping subcategories play an important role for better classification as an ensemble [74], while other subcategory discovery methods find disjoint subcategories, *e.g.*, k-means, Latent-SVM or DSC [24].

3.3.2.2 Sharing Information by Rank Minimization

Minimizing a loss function with a rank constraint is not a convex problem [68], leading to an NP-hard optimization. The trace norm (or nuclear norm), denoted by $\|\cdot\|_\Sigma$, is a convex surrogate for the rank function and is frequently used as an alternative regularization term for efficient optimization [49]. Using the trace norm for minimizing the rank of $\widetilde{W}$, which is the sub-matrix of W selected by I in Eq.(3.1), we reduce Eq.(3.2) to:

$$\min_{\widetilde{W}} \ \|\widetilde{W}\|_\Sigma + \beta \sum_{i=1}^{l} \sum_{k}^{K} h(Y(i,k) I(k) w_k^T x_i), \tag{3.6}$$

where $h(\cdot)$ is the hinge loss function.

Eq.(3.6) is convex since both the trace norm and hinge loss function are convex and can be optimally solved by gradient descent algorithms. Although Eq.(3.6)

is convex, each term has a discontinuous point which is not differentiable. We approximate hinge loss by its smooth proxies as in [2,42] and use a proximal gradient method to optimize the regularized convex problem.

3.3.3 Aggregation of Ensemble Classifier Scores

As opposed to a single category-wide classifier that gives a scalar valued classification score (or confidence value) for a sample, an ensemble of classifiers for a category gives a vector of classifier outputs for a sample. To consolidate the scores of ensemble classifiers we use 'max' aggregation same as [18, 74] as:

$$f(x) = \max_{k} w_k \cdot x, \qquad k \in \{1, \ldots, K\}, \tag{3.7}$$

where k denotes a subcategory mixture component. K is the set of all mixture components. w_k is the template for the k^{th} subcategory, and x is the image feature vector.

3.4 Experimental Evaluation

For empirical validation, we use three datasets; Pascal VOC 2007 [15], Caltech-256 [21] and ImageNet-20, which is a subset of ImageNet dataset, that is similar in size to Pascal VOC 2007 but contains a more visually diverse set of images of fine-grained categories (10 vegetable types and 10 dog breeds) as shown in Fig. 3.2. The categories in ImageNet-20 dataset are randomly selected 10 vegetables and 10 dog breeds including 'Mashed Potato'(MP), 'Crab Apple'(CA), 'Black Berry'(BB), 'Orange'(OG), 'Lemon'(LM), 'Plum'(PM), 'Chard'(CD), 'Green Onion'(GO), 'Acorn'(AC),

'Coffee Bean'(CB), 'Vizsla'(VS), 'Brittany Spaniel'(BS), 'Golden Retriever'(GR), 'Flat-coated Retriever'(FR), 'Yorkshire Terrier'(YT), 'Greyhound'(GH), 'Dalmatian'(DM), 'Corgi'(CG), 'Miniature Poodle'(MPl) and 'Griffon'(GRf). We will publicly release the dataset and the code for future comparison (Link).

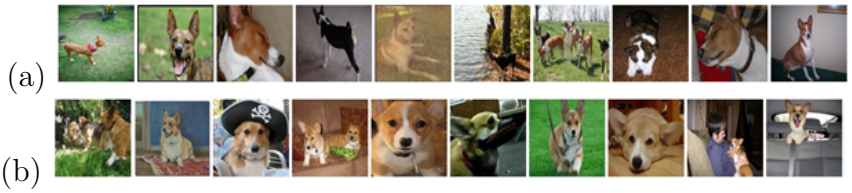


Figure 3.2: **Fine-grained dog breeds in ImageNet-20 dataset.** (a) Basenji (BS) (b) Corgi (CG).

3.4.1 Experimental Setup

Dataset Size: Each set consists of training, validation (for hyper-parameter tuning) and testing set. Pascal VOC 2007 contains 2,501 images for training, 2,510 for validation and 5,011 for testing of 20 categories. With Caltech-256, we use a challenging set-up that has 20 training samples per category (5,120 images for training), 6,400 for validation and 6,400 for testing of 256 categories. The ImageNet-20 dataset is slightly larger than Pascal (20 categories, 3,000 images for training, 5,000 images for validation, 5,000 images for testing).

Visual Feature Descriptors: For Caltech-256 and ImageNet-20 dataset, we use color GIST, BoW of SIFT, Pyramid HOG and Pyramid self-similarity as in [63] and reduce the dimensionality by PCA (and discriminative binary codes (DBC) [54] for Caltech-256 only) to 400 dimensions. For Pascal VOC 2007 dataset, we use OverFeat

features by a deep convolutional neural net (output of FC7 layer of AlexNet [33]) learned on the ImageNet challenge 2012 dataset [28] using the OverFeat implementation by [**?**].

Optimization: For sparse selection of the subcategories (Eq.(3.1)), we use LibLinear [16] which implements an L_1-constrained hinge loss optimization. For optimizing Eq. (3.2), we use MALSAR library with our objective function [72]; it uses the accelerated gradient method (AGM) by computing the proximal operator for trace-norm regularizer. For Latent-SVM and DSC, we use the code provided by the authors of [24].

Parameters: All hyper-parameters are tuned by validation set accuracy. For building the candidate subcategory classifiers, $W \in \mathbb{R}^{D \times M}$, we first cluster the training set into 10 candidate subcategories using k-means (if the number of positive samples is less than or equal to 10, we skip this (Sec.3.4.5)) and generate 1,024 candidates subcategories ($M = 1024$) for all experiments. For all three datasets, we swept broad range of hyper-parameters and picked the best value in terms of accuracy on a held-out validation set.

For baseline SVM, we sweep the balancing parameters between the hinge loss and the regularization terms, usually denoted by $C \in \{10^{-4}, 10^{-3}, \ldots, 10^{3}, 10^{4}\}$. For Power-SVM, we sweep balancing parameter of cross-overs, referred as $D \in \{10^{-4}, 10^{-3}, \ldots, 10^{3}, 10^{4}\}$ as in [70].

For each of E-SVM's and LOO-SVM's, we sweep $C \in \{10^{-4}, 10^{-3}, \ldots, 10^{3}, 10^{4}\}$. For LSVM [18] and DSC [24], we sweep $C \in \{10^{-4}, 10^{-3}, \ldots, 10^{3}, 10^{4}\}$ and numbers

of subcategories (1∼10) as it strongly influences the final classification accuracy.

For the parameters of our method, we explore $\alpha \in \{0.01, 0.1, 0.5\}$ and $\beta \in \{1, 2, \ldots, 25\}$. To construct a candidate classifier space, a matrix of W in Eq.(1) of main paper, we learn linear SVM with $C = 1$ of all combinations of samples or 10-clusters of samples. In other words, constructing the candidate subcategories, we use the exhaustive bagging method for small sample experiments (Sec. 4.5) or exhaustive bagging on 10-clusters obtained by k-means [44].

For kernel based non-linear classifier, we use kernel-SVM with two popular kernels; radial basis and polynomial function. For radial basis function (RBF) kernels, we explore $\sigma \in \{10^{-2}, 10^{-1}, \ldots, 10, 10^2\}$ with $C \in \{10^{-2}, 10^{-1}, \ldots, 10, 10^2\}$. For Polynomial kernel-SVMs, we explore 2,3 and 4 degree polynomials with $g \in \{10^{-2}, 10^{-1}, \ldots, 10, 10^2\}$ with $C \in \{10^{-2}, 10^{-1}, \ldots, 10, 10^2\}$. All multi-class classifications are done in one-vs-all manner.

3.4.2 Recognition Accuracy

For quantitative analysis, we summarize mean average precision (mAP) of ours and compared methods in Table 3.1. P-SVM denotes Power-SVM, which is a state-of-the-art linear classifier [70]. E-SVM denotes an ensemble of Exemplar-SVM's, which is a baseline ensemble classifier [45]. LOO-SVM denotes an ensemble of Leave-One-Out-SVM's, which are maximally overlapping subcategory classifiers [11]. LSVM and DSC denote an ensemble of Latent-SVM's and its modified version [18].

Compared to baselines and state-of-the-art classification methods of both single classifier approaches and ensemble approaches, our method shows the best overall performance.

Dataset	SVM	P-SVM	E-SVM	LOO-SVM	LSVM	DSC	**Ours**
Imgnet20	41.03	41.93	21.95	41.02	39.56	39.06	**43.87**
C256-tr20	27.65	26.59	16.77	27.51	22.90	22.90	**29.85**
VOC2007	73.92	73.56	63.44	74.14	71.15	71.06	**74.70**

Table 3.1: **Recognition accuracy in mean average precision (mAP,%).** On ImageNet-20 (Imgnet20), Pascal VOC 2007 (VOC2007) and Caltech-256 (C256-tr20). Comparison with state-of-the-art subcategory based approaches (LSVM and DSC), baseline ensemble approach (E-SVM) and its complementary version (LOO-SVM), state-of-the-art linear classifier (P-SVM) and baseline (SVM).

For a detailed analysis, we present class-wise average precision (AP) on ImageNet-20 and Pascal VOC 2007 in Table 3.2. In the table, 'SVM' denotes a class-wide Linear SVM, 'P-SVM' denotes a class-wide Power-SVM, 'E-SVM' denotes an ensemble of exemplar-SVM's, 'LOO-SVM' denotes an ensemble of 'Leave-One-Out' SVM's that are learned on maximally overlapping subcategories, 'LSVM' denotes latent-SVM used in the deformable parts model, 'DSC' denotes discriminative subcategorization method and 'Ours' denotes the proposed method. **Bold** indicates the best accuracy across the methods. All hyper-parameters are determined using a held-out validation set. Our method significantly outperforms other methods on the categories on which the baseline SVM performs poorly (see Sec. 3.4.3 for further

Imgnet-20	Vegetables										Dogs										Avg.
Method\Cat.	MP	CA	BB	OG	LM	PM	CD	GO	AC	CB	VS	BS	GR	FR	YT	GH	DM	CG	MPl	GRf	
SVM	77.00	**44.41**	62.06	43.70	37.25	37.19	**53.06**	51.02	17.31	64.65	18.97	27.95	18.91	50.10	36.61	17.50	**74.75**	24.45	40.39	23.27	41.03
P-SVM [70]	74.02	42.14	61.75	40.04	40.57	40.74	47.18	51.12	20.51	60.41	22.63	31.01	23.59	**55.00**	37.14	**20.70**	73.13	28.83	37.78	30.29	41.93
E-SVM [45]	33.85	21.74	26.89	25.88	21.92	17.62	23.60	34.69	10.02	36.29	12.59	13.52	13.08	24.08	18.54	12.44	35.65	12.98	23.09	18.75	21.86
LOO-SVM [11]	**77.02**	44.20	62.00	43.51	37.31	37.05	52.93	51.14	17.18	**64.75**	18.79	28.65	18.75	49.57	36.55	17.64	74.66	25.20	40.28	23.03	41.01
LSVM [18]	70.82	43.29	60.39	36.91	37.97	39.82	43.30	46.81	18.18	62.35	20.92	21.63	19.89	53.14	35.54	18.84	69.87	27.37	35.41	28.79	39.56
DSC [24]	70.82	43.29	60.39	39.19	37.97	39.82	43.30	35.07	17.04	62.39	20.56	29.89	21.98	53.14	31.25	18.84	69.87	27.37	30.16	28.79	39.06
Ours	75.61	43.07	**62.53**	**44.90**	**42.33**	**45.21**	52.49	**54.28**	**22.17**	62.04	**23.07**	**32.52**	**25.01**	53.34	**37.94**	20.36	74.06	**30.48**	**43.03**	**33.00**	**43.87**

VOC2007	Plane	Bicycle	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Table	Dog	Horse	Mbike	Person	Plant	Sheep	Sofa	Train	TV	Avg.
SVM	91.12	82.82	86.85	84.67	36.67	**74.93**	**88.08**	**84.27**	52.73	58.40	63.10	81.35	83.55	81.33	**91.78**	51.19	70.87	**59.92**	90.81	68.21	73.92
P-SVM	90.90	82.87	86.17	85.03	38.02	73.46	86.60	83.37	52.66	**58.63**	62.39	79.60	**83.68**	80.82	91.34	49.43	71.09	58.70	90.11	66.47	73.56
E-SVM	87.59	73.31	80.78	76.58	23.61	63.39	82.40	72.27	38.74	40.10	47.28	69.54	77.91	70.47	85.40	36.51	62.19	34.18	86.83	59.64	63.44
LOO-SVM	91.15	82.86	86.84	84.70	36.73	74.84	88.07	84.28	52.70	58.35	63.16	**81.37**	83.62	81.32	91.80	51.23	71.07	59.65	90.83	**68.25**	74.14
L-SVM	**91.30**	81.00	**85.12**	83.72	34.47	67.37	86.01	81.09	53.01	54.94	**63.88**	77.03	79.99	78.62	91.71	41.36	65.19	57.06	89.47	60.60	71.15
DSC	**91.30**	81.00	86.20	83.72	32.82	67.17	86.01	81.09	53.01	54.94	57.58	78.07	79.99	78.11	91.71	43.35	67.56	57.06	89.85	60.60	71.06
Ours	91.20	**83.34**	87.12	**85.55**	**40.52**	74.49	87.96	84.13	**55.79**	57.87	63.55	81.35	82.71	**81.78**	**91.07**	**54.52**	**72.24**	59.56	**91.07**	68.16	**74.70**

Table 3.2: **Class-wise average precision (%) on ImageNet-20 (upper) and Pascal VOC 2007 dataset (lower).**

discussion), *e.g.*, *Plum* (PM), *Acorn* (AC), *Golden Retriever* (GR) and *Corgi* (CG) in ImageNet-20 and *Dining Table* (Table) and *Sofa* in Pascal VOC 2007.

Surprisingly, LSVM and DSC overall perform poorly, even though we performed multiple optimizations and picked the best performing classifiers (the optimization of LSVM and DSC uses a stochastic gradient decent and finds a local minimum). But they outperform the linear SVM in 65 categories of Caltech-256, 10 categories of ImageNet-20 and 3 categories of Pascal VOC 2007. Like our method, when the baseline SVM performs poorly, they improve the accuracy on average (see Sec. 3.4.3).

For qualitative analysis, we present test images sorted by (consolidated) classifier confidence, in descending order, obtained by various methods including ours in Fig. 3.3. In the figure, two vegetable categories and two dog breed categories

on ImageNet-20 dataset. Red rectangle indicates mis-classified samples. Note that our method not only classifies more samples correctly but also find the subcategory traits (*e.g.*, the bundled green onions, half cut oranges, a zoomed-out greyhound in the center and frontal face of a griffon) and semantically similar images (*e.g.*, no dogs in 'Green Onion' and no vegetables in 'Greyhound'). The red boxes around images indicate misclassified samples. Our method not only correctly classifies more samples in top-retrieval rank (low-recall region) but also identifies samples that exhibit subcategory traits, *e.g.*, the bundled green onions, half cut oranges, a zoomed-out greyhound in the center and frontal face of a griffon (refer to **'Ours'** row). In addition, the retrieved images by our method are semantically more consistent than ones by other methods, *e.g.*, no dogs in 'Green Onion' and no vegetables in 'Greyhound'.

We also compare to kernel based non-linear classifiers which require expensive resources such as large memory space and computational burden at test time. Since it is not trivial to choose a proper kernel for each classification task, we tried various kernels including polynomial kernels with various degrees (2, 3 and 4) and Radial Basis Functions (RBF) with extensive hyper-parameter tuning. Our method is not only fast (0.91 sec, compared to 230.53 sec) but also shows comparable performance to kernel based methods (1$\sim$2% overall accuracy difference) and even outperforms them in a small-training set scenario (see Sec. 3.4.5).

Generalization by Training Error and Testing Accuracy. Our ensemble classification model adjusts the model complexity by information sharing among classifiers. The level of overfitting as a function of model complexity is best shown

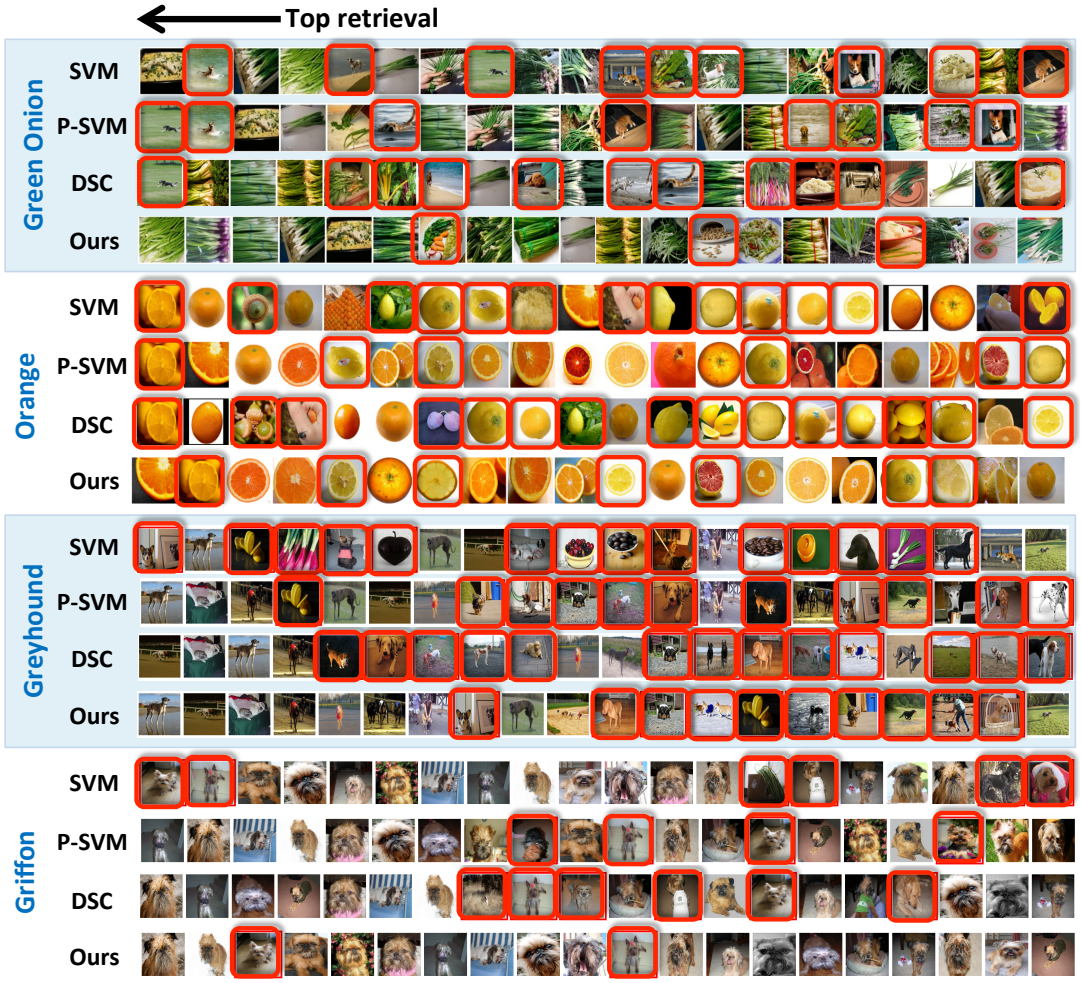


Figure 3.3: **Qualitative comparison of our method to other methods.**

by both training error and test error as shown in Fig.2.11 in Hastie *et al.* [23]. Thus, it is interesting to investigate the training errors of the methods to observe the overfitting and generalization. Table 3.3 shows training error alongside with testing accuracy of each method on each dataset.

Even with higher training error, our method achieves better test accuracy, which implies that our method learns a well-balanced set of classifiers between under- and overfitted ensembles. Baseline SVM achieves decent training error rate but not competitive test accuracy, which implies its classification boundary is not gener-

Dataset	Type	SVM	P-SVM	E-SVM	LOO-SVM	LSVM	DSC	**Ours**
Imgnet20	Tr.Err.	19.23	**32.46**	0.0	18.7	25.14	22.52	30.00
	Ts.Acc.	41.03	41.93	21.86	41.01	39.56	39.06	**43.87**
VOC2007	Tr.Err.	2.66	0.24	0.0	2.62	1.02	0.95	**8.71**
	Ts.Acc.	73.92	73.56	63.44	74.14	71.15	71.06	**74.70**
C256-tr20	Tr.Err.	0.01	**71.75**	1.92	0.02	0.68	0.41	0.75
	Ts.Acc.	27.65	26.59	16.77	27.51	22.90	22.90	**29.85**

Table 3.3: **Average training errors and testing accuracy (mAP,%).** When both the training error (Tr.Err.) and the test accuracy (Ts.Acc) are high, classifier is less overfitted and more generalizable.

alizable. Power-SVM achieves the highest training error but the second best test accuracy, which implies the underfitting by enforcing too much regularization by the exemplar uncertainty. As an E-SVM overfits to each example, it shows zero or very small training error but shows poor generalization performance. LOO-SVM show similar training error and testing accuracy to SVM as each LOO-SVM is trained with only one fewer training sample than SVM. LSVM and DSC show higher training error than SVM but do not generalize well.

3.4.3 When Do We Need Subcategory Based Methods?

Intuitively, subcategory based methods are for the category that linear method cannot perform well. [74] shows that subcategory based methods perform bet-

ter when the baseline classifier suffers (possibly due to underfitting). We also observe the same trend; compare the improvement in 'Acorn'(AC), 'Viszla'(VS), 'Golden Retriever'(GR) 'Greyhound'(GH), 'Corgi'(CG) and 'Griffon'(GRf) categories in ImageNet-20 and 'Chair' and 'Table' in Pascal VOC 2007, on which the average precision (AP) of SVM are less than 25% as shown in Table 3.2.

We summarize the mean average precision (mAP) of *difficult* categories where linear SVM performs worse than 25% (Imgnet-20 and VOC2007) or the 50-worst categories of linear SVM performance (C256-tr20) and the mAP of *easy* categories where linear SVM performs better than 70% (Imgnet-20 and VOC2007) or the top-50 categories of linear SVM performance (C256-tr20) in Table 3.4. In the table, *Difficult* categories denote the ones on which Linear SVM performs worse than 25% (Imgnet-20), 40% (VOC2007) or the 50-worst categories of linear SVM performance (C256-tr20) and *easy* categories denote the ones on which Linear SVM performs better than 70% (Imgnet-20 and VOC2007) or the top-50 categories of linear SVM performance (C256-tr20).

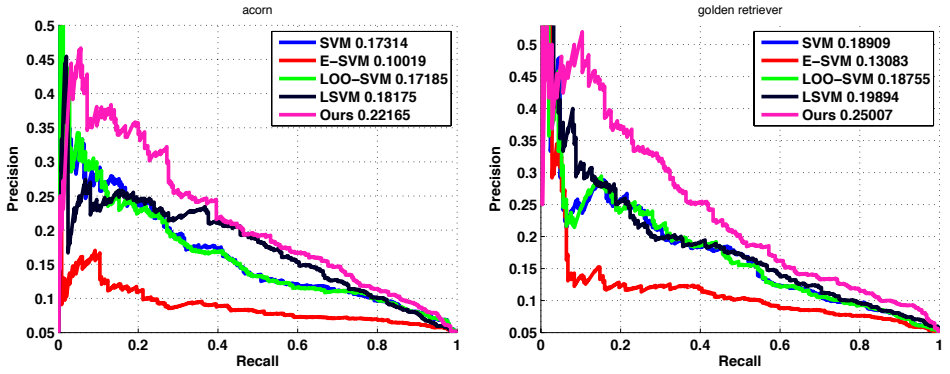
In the *difficult* categories, subcategory-based methods such as LSVM, DSC and Ours outperform SVM significantly ($\sim 4.19\%$ overall). In the *easy* categories, subcategory methods outperform linear methods but with a small margin ($0.2 \sim 1.54\%$ overall). For detailed analysis, we show examples of precision-recall curves of two difficult categories and two less difficult categories in Fig. 3.4.

		Linear		Subcat. Ensemble				
mAP(%)	# cat	SVM	P-SVM	E-SVM	LOO-SVM	LSVM	DSC	**Ours**
Difficult Categories								
Imgnet-20	6	18.12	**21.65**	11.22	18.09	19.38	19.76	**22.07**
VOC2007	3	46.86	46.70	32.95	46.89	42.95	43.06	**50.28**
C256-tr20	50	4.44	**5.52**	1.83	4.37	**5.10**	**5.15**	**6.33**
Easy Categories								
Imgnet-20	2	72.91	71.73	33.23	**73.11**	70.97	67.19	72.60
VOC2007	13	84.03	83.46	76.05	84.06	81.36	81.67	**84.16**
C256-tr20	50	66.78	62.50	50.43	66.70	58.37	58.28	**68.32**

Table 3.4: **Average recognition accuracy (mAP,%) of *difficult* (upper) and *easy* (lower) categories.**

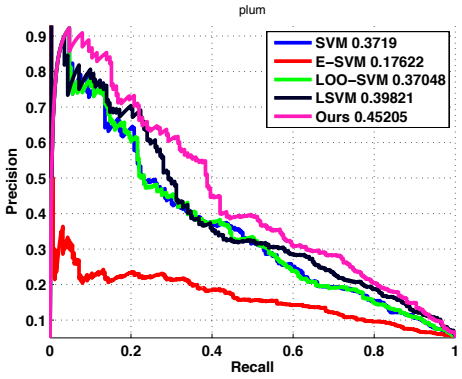
3.4.4 Subcategory Configuration for Better Classification

It is not obvious what set of subcategories are good for better classification by an ensemble. Specifically, for better classification, it is not known 1) what is the optimal number of subcategories 2) how to find the subcategories that are good for ensemble classification. We seek the answers to these questions in the following subsections.

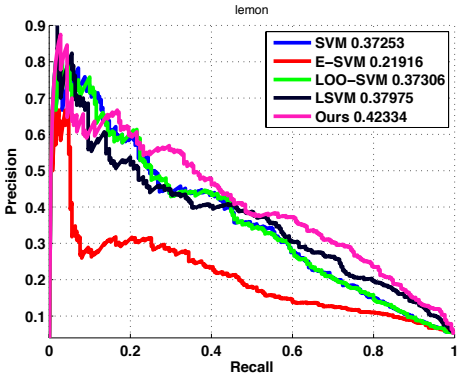


(a) Acorn (AC)

(b) Golden Retriever (GR)



(c) Plum (PM)



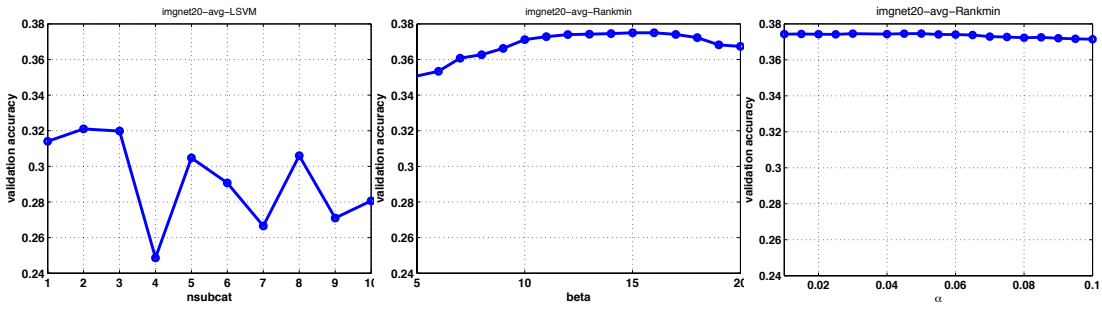
(d) Lemon (LM)

Figure 3.4: **Precision-recall curves of *difficult* categories (a,b) and *easy* categories in ImageNet-20 (c,d).** Numbers in legend indicate average precision (AP) of each method.

3.4.4.1 Optimal Number of Subcategories

All previous methods require to specify the number of subcategories as a hyper-parameter [18, 24] and classification accuracy is sensitive to the number of subcategories as shown in Fig. 3.5-(a). It is, however, not obvious how many subcategories are optimal for better classification. Our method does not require to specify the number subcategories but implicitly determines the appropriate number of subcat-

egories through the balancing hyper-parameter between the rank of the classifier matrix and the loss function, β, and the accuracy is not sensitive to the choice of β as shown in Fig. 3.5-(b). In addition, the accuracy by our method can be affected by candidate subcategory configurations which are controlled by α in Eq.(3.1) (The larger α results in more candidate subcategories). But the accuracy is also not sensitive to the choice of α as shown in Fig. 3.5-(c). (see the related discussion about Table 3.6).



(a) (b) (c)

Figure 3.5: **Sensitivity to hyper-parameters for subcategory Discovery.** Average validation accuracy (mAP) on ImageNet-20 dataset as a function of a hyper-parameter. Note that scale of the y-axis is the same in all figures for easy comparison. (a) Accuracy by LSVM as a function of number of subcategories specified (b) Accuracy of our method as a function of β with fixed α (c) Accuracy of our method as a function of α with fixed β.

3.4.4.2 Information Sharing Amongst Subcategories

As argued in [74], the overlap of subcategories plays an important role in obtaining better consolidated classification accuracy in an ensemble. The overlap

of the subcategories is equivalent to shared information among subcategories. Our method implements information sharing by discovering overlapping subcategories (Eq. (3.1)) and learning the subcategory classifier jointly by enforcing similarity amongst them (Eq. (3.2)). Fig. 3.6 shows the number of samples per subcategory that are discovered by DSC and our method (Eq.(3.1)). The larger the number of big subcategories, the more the subcategories overlap. Our method finds more large subcategories than DSC.

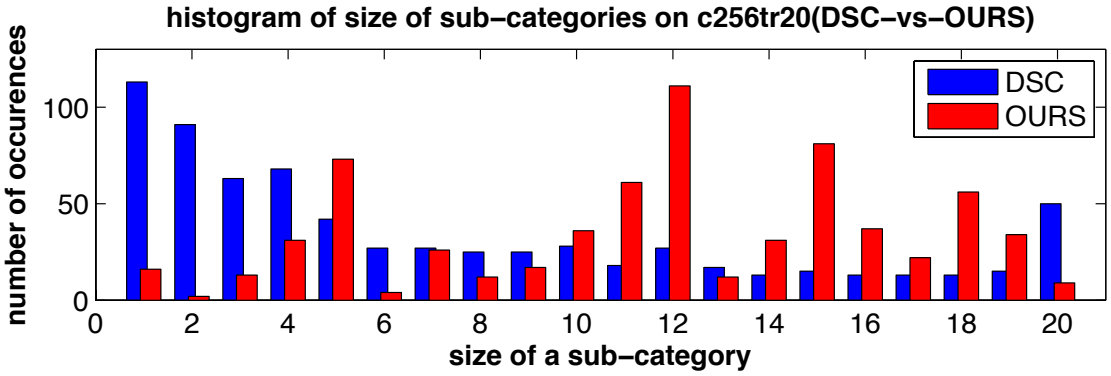


Figure 3.6: **Histogram of size of subcategories discovered by DSC and our method on Caltech-256 dataset.** More number of large subcategories implies that many subcategories overlap.

Fig. 3.7 shows an example of subcategories ('Green Onion'(GO) category of ImageNet-20 dataset) discovered by DSC and our method. The subcategories discovered by DSC are visually distinctive with respect to each other and do not overlap, whereas the ones selected by our method are not visually distinctive but are highly overlapped and have *discriminative diversity* for better classification.

With the discovered subcategories, we enforce the subcategory classifiers to be similar to each other so that the underlying subcategories are further overlapped

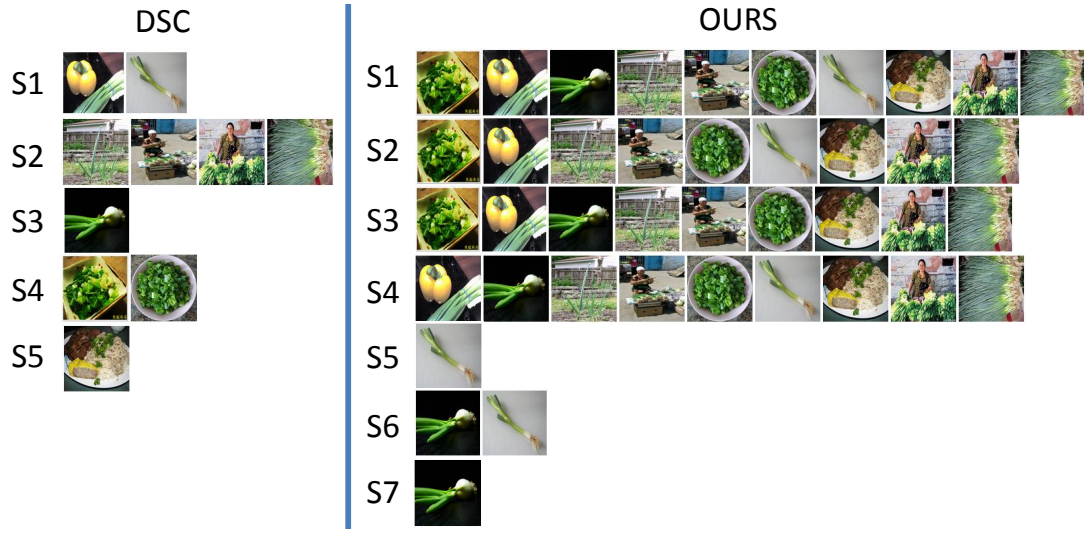


Figure 3.7: **Subcategory discovered by DSC and our method.** On 'Green Onion'(GO) category in ImageNet-20.

by optimizing Eq. (3.2). In Fig. 3.8, we plot the accuracy on validation set and the trace norm of the classifier matrix W as a function of iteration of gradient decent and proximal projection. The lower the trace norm, the more information is shared. As the iteration progresses, the trace norm of the classifier matrix decreases and the accuracy increases.

Although the joint learning of the ensemble classifiers (Eq.(3.2)) depends on

Dataset	Indep.	**Ours**
Imgnet20	41.70%	**43.80%**
VOC2007	72.26%	**74.70%**
C256-tr20	24.94%	**29.85%**

Table 3.5: **Information sharing and accuracy.** Average test accuracy by ensemble classifier learned independently (Indep.) and by our method (Ours).

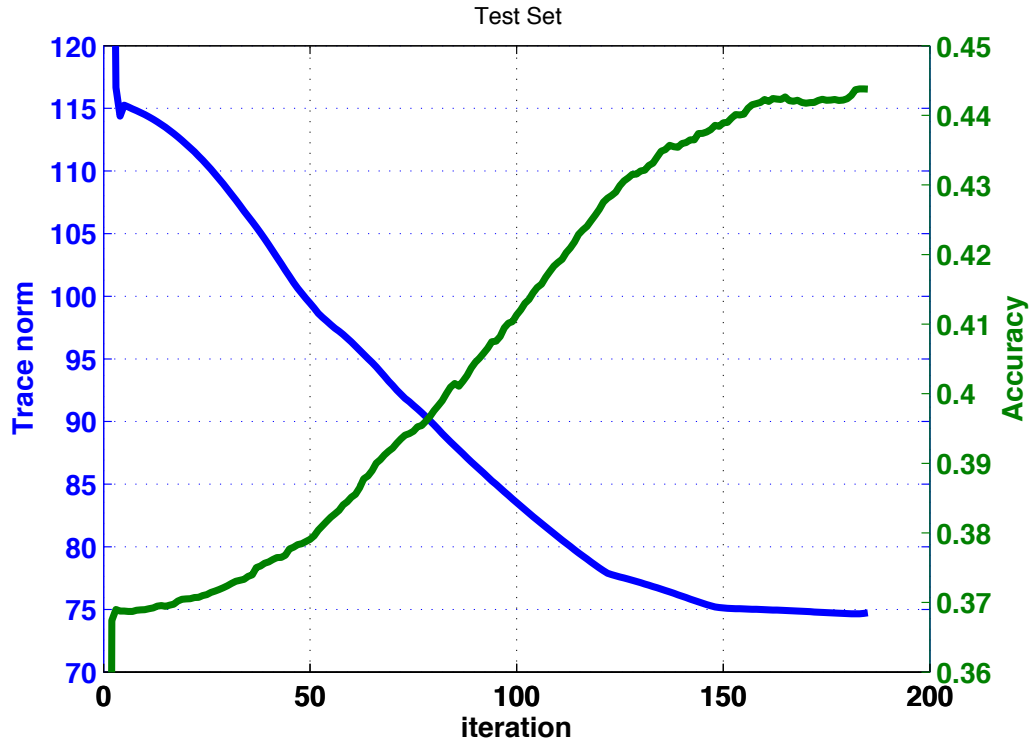


Figure 3.8: **Trace-norm of $\widetilde{W}$ and accuracy. (Left)** Validation accuracy on 'Crab Apple' category in ImageNet-20 dataset. As optimization iteration proceeds, trace norm decreases and accuracy increases.

the discovered subcategory structure via Y, we find that the structure of the subcategories does not significantly affect the classification accuracy. Table 3.6 shows the accuracy of the rank minimized classifiers using different subcategory discovery methods. 'DSC' refers to the subcategories discovered by DSC. 'DSC-ov' refers to overlapping DSC subcategories using [74]. The accuracy of rank minimized classifier on different subcategory configurations are similar even with exemplar subcategories. But our proposal of discovering the subcategory candidates still shows the best results as it directly finds a good set of subcategories for classification.

	K-Means	Exemplar	LOO	DSC	DSC-ov	**Ours**
mAP(%)	43.29	43.48	43.50	43.52	43.72	**43.87**

Table 3.6: **Mean average precision (mAP,%) on rank minimized ensemble classifier learned on different subcategory discovery methods.** On ImageNet-20 dataset. LOO refers to 'Leave-One-Out' scheme for a set of subcategories that are maximally overlapping. 'DSC-ov' refers to overlapping categories by [74] on subcategories found by DSC. All hyper-parameters are determined by a validation set.

3.4.5 Generalization in Small Training Set Scenario

When there are few training samples given, classification generalization is more important but challenging [57]. As the number of labeled training samples decreases, we expect that the accuracy improvement by our method also decreases. Fig. 3.9 shows mAP of various methods as a function of training set size. Notably, even with only 10 training samples per category, our method still outperforms the linear baseline and performs comparably to other methods.

It is interesting to note that in the small training set scenario (10 training samples per category), our method outperforms kernel based non-linear classifiers (K-SVM) as shown in Table 3.7.

For the choice of kernel, we pick the best kernel among 2,3 and 4 degree polynomial and RBF kernels with extensive hyper-parameter tuning on a held-out validation set. We believe that the kernel method overfits to the small training data so the test accuracy is lower than regularized ensemble classifiers.

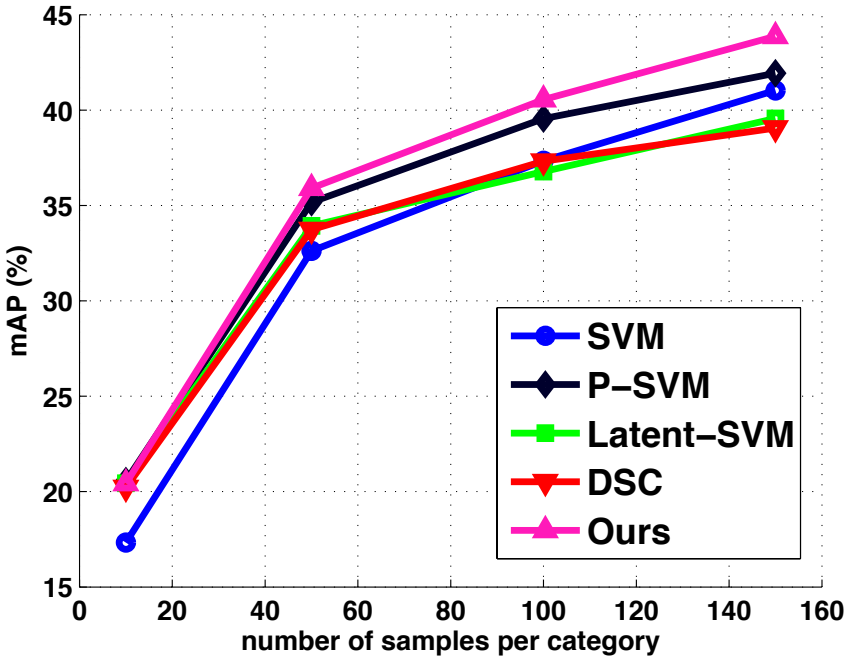


Figure 3.9: **Mean average precision (mAP,%) as a function of size of training set on ImageNet-20 dataset.** Even on a small training set (10 samples per category), improvement by our method is still noticeable.

3.5 Conclusion

We presented an approach to generalizing visual category classification. To achieve this goal, we learn a set of classifiers that not only preserve unique traits of subcategories but also share commonalities amongst them. To learn such a set, we discover the discriminative overlapping subcategories in a classifier space and jointly learn a set of subcategory classifiers that share subcategory commonalities by minimizing the rank of the matrix of subcategory classifiers.

Our method outperforms category-wide single classifier approaches including baseline SVM, the state-of-the-art Power-SVM [70] and subcategory based classifi-

mAP(%)	#tr/cat	SVM	K-SVM	LSVM	DSC	**Ours**
Imgnet20	10	17.33	20.56	**20.82**	20.78	20.52
VOC2007	10	43.47	44.26	43.17	43.49	**46.06**
C256	10	20.88	22.19	18.49	18.56	**23.70**

Table 3.7: **Mean average precision (mAP,%) in a Small-Training Set Scenario.** Our method outperforms a kernel based non-linear classifier (K-SVM).

cation approaches such as the ensemble of Exemplar-SVM's [45], Latent-SVM [18], the state-of-the-art discriminative subcategory method (DSC) [24] on three visual category recognition datasets.

3.6 Hyper-parameters for Experiments

For all three datasets, we swept broad range of hyper-parameters and picked the best value in terms of accuracy on a held-out validation set.

For baseline SVM, we sweep the balancing parameters between the hinge loss and the regularization terms, usually denoted by $C \in \{10^{-4}, 10^{-3}, \ldots, 10^{3}, 10^{4}\}$. For Power-SVM, we sweep balancing parameter of cross-overs, referred as $D \in \{10^{-4}, 10^{-3}, \ldots, 10^{3}, 10^{4}\}$ as in [70].

For each of E-SVM's and LOO-SVM's, we sweep $C \in \{10^{-4}, 10^{-3}, \ldots, 10^{3}, 10^{4}\}$. For LSVM [18] and DSC [24], we sweep $C \in \{10^{-4}, 10^{-3}, \ldots, 10^{3}, 10^{4}\}$ and numbers of subcategories (1$\sim$10) as it strongly influences the final classification accuracy.

For the parameters of our method, we explore $\alpha \in \{0.01, 0.1, 0.5\}$ and $\beta \in \{1, 2, \ldots, 25\}$. To construct a candidate classifier space, a matrix of W in Eq.(1)

of main paper, we learn linear SVM with $C = 1$ of all combinations of samples or 10-clusters of samples. In other words, constructing the candidate subcategories, we use the exhaustive bagging method for small sample experiments (Sec. 4.5) or exhaustive bagging on 10-clusters obtained by k-means [44].

For kernel based non-linear classifier, we use kernel-SVM with two popular kernels; radial basis and polynomial function. For radial basis function (RBF) kernels, we explore $\sigma \in \{10^{-2}, 10^{-1}, \ldots, 10, 10^2\}$ with $C \in \{10^{-2}, 10^{-1}, \ldots, 10, 10^2\}$. For Polynomial kernel-SVMs, we explore 2,3 and 4 degree polynomials with $g \in \{10^{-2}, 10^{-1}, \ldots, 10, 10^2\}$ with $C \in \{10^{-2}, 10^{-1}, \ldots, 10, 10^2\}$. All multi-class classifications are done in one-vs-all manner.

Chapter 4: Interactive Semantics for Knowledge Transfer

4.1 Introduction

In recent years, semantic information has been exploited extensively to improve object category recognition accuracy, since object categories are essentially semantic entities that are human-defined. Various types of semantic sources have been exploited such as attributes [1, 26], taxonomies [66, 71], and analogies [25], as auxiliary information to aid categorization. However, exhaustive top-down construction of such knowledge bases could be expensive as it takes large amount of human effort to obtain, and such knowledge base might be largely unavailable for non-generic set of object categories, *e.g.*, recognizing the specific year/model of a vehicle, or cartoon characters from animation database.

Further, not all knowledge is equally useful in the discriminative classification sense. For example, knowing that an *apple* is more similar to a *pear* than a *dragon*, while semantically meaningful, may not be useful to distinguish it from other fruits. Thus, finding and using a proper set of semantic information (*e.g.*, that an *apple* is more similar to a *pear* than a *mellon*) can greatly and efficiently enhance recognition accuracy. In addition, it is difficult to construct the optimal vocabulary of semantic information without prior knowledge about the object itself and/or other categories

that are potentially confusing. This is evident from the game of 20-questions. Every question asked (and answered) has a profound effect on the distribution of questions a player may ask next. It is clear that if a player needed to ask all the questions at once, upfront, he may need considerably more than 20 questions to identify the object in question.

To address such challenges, we propose a method to obtain and leverage a focused set of semantic queries by examining a discriminatively learned model for object categorization in an interactive learning framework. Starting from a base model with no semantic information, we want to iteratively improve it by generating semantic queries for human(s) to answer, then in turn update the existing model with feedback. We expect such an interactive learning system to help transferring knowledge effectively from *anchor* categories, that are well learned, to the *target* categories that have very small number of labeled training samples.

Our contributions are threefold: (1) We propose an interactive learning framework that can be incrementally improved, by asking for verification of semantic queries from humans and taking that feedback into account. (2) As part of the learning framework, we present an active selection method that automatically generates semantic queries from a learned model by detecting relational regularities, and ranking them by their expected impact on the recognition performance. (3) We empirically validate that our method can transfer knowledge for better classification via relational semantics to *target* categories, and thus improve classification performance on them. Figure 4.1 shows the overview of the approach. Our model is a discriminative manifold with embedded semantic entities. The discriminative

categorization model is refined by iteratively generating semantic questions and user feedback. Thumbnail images denote category prototypes in the embedding space. The categories are partitioned into two sets: *Anchor* classes, that have reasonable number of samples per class, and *Target* classes, that have few labeled instances, to which the semantic knowledge is transferred. From the semantic embedding space, we detect relational hypotheses based on classification confusion among target and anchor classes. Approach consists of three steps: (1) finding confusing classes in the target set and confident classes in the anchor set and generating triplet-based relationships (*e.g.*, target class *Chimpanzee* is closer to anchor class *Gorilla* than to anchor class *Deer*); (2) translating the detected relational hypotheses into ranked list of semantic questions to obtain human judgement concerning their validity; (3) translating validated geometric relations into regularizers for the objective function and retraining the model.

4.2 Related Work

Encoding Semantics for object recognition: The most popular semantic information that has been explored for improving recognition accuracy is attributes and taxonomies [1, 26, 46, 71]. While most previous work leverages taxonomies and attributes by focusing on shallow properties such as similarity between the semantic entities, some recent works focus on the their geometrical relationships. [48] showed that there exist regularity between word vectors trained on the skip-gram models (*e.g.*, the words form analogies). The same analogical relations were explored

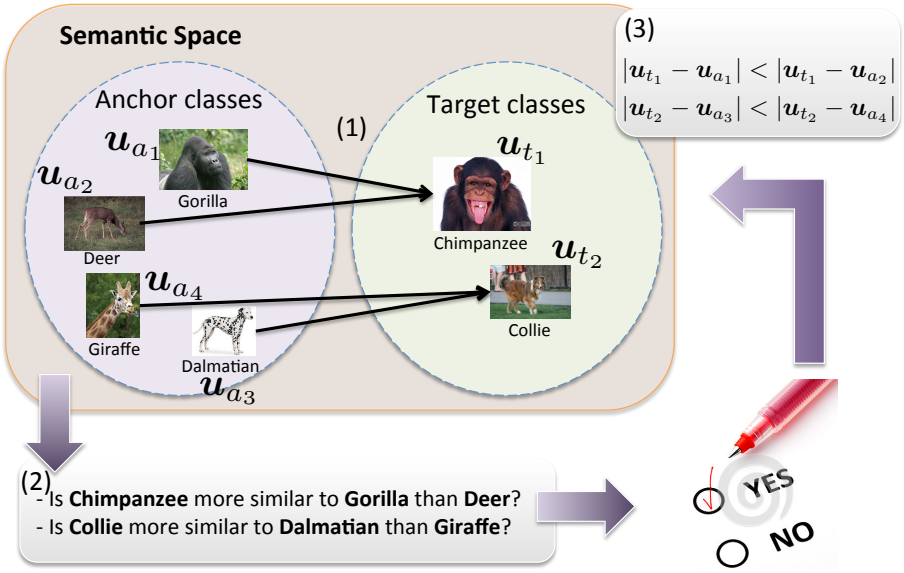


Figure 4.1: **System Overview.**

in [25] and [39] to regularize the geometry of the learned category embeddings for categorization, such that category embeddings associated with an analogy form a parallelogram. More similar to our design, [64] took advantage of relative closeness, encoded by triplets of entities.

The limitation of all these methods is that they require a pre-constructed knowledge base, which often takes a lot of human effort and expertise to create. Such knowledge bases may not be readily available for atypical classes, *e.g.*, specific dog breeds or exotic car models. Our method does not require a predefined knowledge base, and, in fact, is designed to ascertain the most informative, from the model's point of view, knowledge relationships from human users/expert(s).

Active/Interactive/Self-Paced Learning: Our method, which actively selects a few important relational patterns to validate through user feedback, is an instance of ac-

tive/interactive learning. The generic active learning focuses on selecting instances based on possible contribution that the selected instances can make towards improving the classification model and asking for corresponding category labels from a human annotator. Recently, non-class label type queries have been also explored for active-learning, such as in [32], which presents an active learning algorithm that can either ask for attribute or category labels, while learning a joint object categorization model. Pairwise similarity, which forms our relational patterns, has also been explored in [29]. However, in [29], the queries are selected to better search for the target in a fixed metric space, while our method iteratively retrains the metric space with the answered queries.

The closest work to ours, in terms of motivation, is [13], which generates active queries considering the geometry of the manifold, and retrains the model with the newly annotated samples. However, [13] focuses on the instance(sample)-level geometry while we focus on *semantically* important geometrical patterns among category prototypes. [8] makes use of the graph structure to select instances from groups for queries; instances whose collective label prediction disagrees with instance label prediction are preferred. Our method also makes use of structural relationships, but focuses on the geometry of category prototypes rather than instances.

[51] also closely share our motivation of building a semantic model by iteratively selecting semantically meaningful hypotheses from a pool of candidates. They generate discriminative visual attribute hypotheses and then present human subjects a set of images with and without such attributes, and ask them to name the attributes that differentiate between the two, only if the difference is nameable.

The model with such semantic refinement was shown to outperform the non-semantic initial variant.

Self-paced learning, or curriculum learning, [34, 40] is a learning paradigm that incrementally learns from subsets of labeled instances instead of learning in a batch. Self-paced learning iteratively builds the model using samples that are discovered adaptively, based on the model at the previous iteration. Our approach is an instance of self-paced learning, but discovers semantic constraints rather than instances. Further, since semantics are latent, and do not directly correlate with recognition performance, the criterion for iterative selection of such entities is much more difficult to formalize.

Lifelong learning: Lifelong learning [61] is a learning paradigm that continuously learns from a stream of incoming inputs, while transferring knowledge obtained from earlier stages to later ones. Lifetime learning has gained popularity due to its scalability and applications that deal with long streams of inputs, *e.g.*, in web-scale data, wearable cameras and autonomous vehicles. Since the inception of the idea by the seminal work of [61], many researchers have worked on such continuous learning systems. Recent work includes [14], which learns the shared basis for all tasks in an online learning framework. The model was later expanded, in [56], to allow active selection of tasks at each iteration. We hope that our interactive learning paradigm, that learns semantic information online, can serve as a module in such lifelong learning frameworks. In doing so, it would allow mitigation of *semantic drift* through intermittent, but focused, human feedback.

Knowledge Transfer When little labeled data is available for certain categories, transferring knowledge from related categories can be helpful [53, 62]. [62] adapt classifiers for classes with small number of training instances by utilizing information from classifiers of classes with sufficiently large number of training instances. However, they transfer information in a batch, where as our method focuses on incremental transfer and improvements. [53] similarly use cross-category knowledge to improve the image classification accuracy in a batch.

4.3 Approach

Given a labeled dataset $D = \{(\boldsymbol{x}_i, y_i) \in (\mathbb{R}^d, \mathcal{Y})\}_{i=1}^N$, where $\boldsymbol{x}_i$ is a d-dimensional feature vector of i^{th} example, y_i is its class label and N is the number of examples, we learn a model that minimizes classification error for new, unknown, example $\boldsymbol{x}^*$ at test time. We adopt an efficient and scalable discriminative embedding approach [5] to classification, where both the samples, $\boldsymbol{x}_i$, and their labels, y_i, are projected into a common low dimensional space $\in \mathbb{R}^m$, where $m \ll d$. We denote the projected version of $\boldsymbol{x}_i$ as $\boldsymbol{z}_i = f(\boldsymbol{x}_i)$ and class label $y_i = c \in \mathcal{Y}$ as $\boldsymbol{u}_c$. The goal is then to learn both the embedding function $f(\cdot)$ and the location of the prototypes $\boldsymbol{u}_c$ for all classes such that the projected version of the test instance $f(\boldsymbol{x}^*)$ would be closer to the correct class prototype than to others.

If one assumes existence of semantic information, the above model can be further improved [25, 39] through graph-based regularization, *i.e.*, semantic relationships as constraints on the placement of prototypes in the embedding space.

However, as the number of entities increase, the number of possible relationships between them increases rapidly, making it difficult to annotate all semantic relationships offline. Further, even if one has a complete set of semantic information, not only using all of semantic relationships lead to an unjustifiable computational expense, but also not all semantics would be equally useful for discriminative classification, which suggests that encoding all of the semantics may even degrade the classification performance. One often needs to trade off discriminative classification accuracy for the ability to encode all the semantics entities in the knowledge set with a fixed dimensional manifold. To address this, we aim to actively identify a compact subset of semantic relations that are most helpful in learning a discriminative classification model.

We make use of semantics in the form of relative distance: "class a is more similar to class b than to class c." However, the total number of such triplet relationships is cubic in the number of category labels. To alleviate prohibitive cost of attaining a complete semantic knowledge base, we propose an interactive approach to require subset of them. Specifically, we repeat the following three steps. 1) detect geometric patterns that constitute potential semantic triplet queries with respect to the current model, 2) obtaining 'yes' or 'no' answers to these semantic questions from a human user and 3) retraining the model by imposing structural regularizers based on the obtained semantic knowledge. We summarize the overall procedure of our method in Algorithm 1 and describe detailed steps in the following subsections.

Algorithm 1 Interactive Learning with Semantic Feedback

Input: $(x_i, y_i) \in \mathbb{R}^d \times \mathcal{Y}, \quad \forall i \in \{1, \dots N\}$.

Output: $\boldsymbol{W} \in \mathbb{R}^{m \times d}, \boldsymbol{U} \in \mathbb{R}^{m \times C}$.

1: $\mathcal{R} \leftarrow \emptyset$

2: Initialize $\boldsymbol{W}_{prev}, \boldsymbol{U}_{prev}$ with random matrices

3: $\boldsymbol{W}^A$ and $\boldsymbol{U}^A \leftarrow$ Solve Eq.(4.1)

4: $\delta \boldsymbol{W} = \boldsymbol{W}^A - \boldsymbol{W}_{prev}$, $\delta \boldsymbol{U} = \boldsymbol{U}^A - \boldsymbol{U}_{prev}$

5: **while** $\delta \boldsymbol{W} > \epsilon$ and $\delta \boldsymbol{U} > \epsilon$ **do**

6: $\quad \boldsymbol{W}$ and $\boldsymbol{U} \leftarrow$ Solve Eq.(4.2) with $\mathcal{R}, \boldsymbol{W}^A$

7: $\quad \mathcal{P} \leftarrow GenerateOrderedQueries(\boldsymbol{W}, \boldsymbol{U}, \mathcal{R})$ (Sec. 4.3.2)

8: $\quad R \leftarrow Feedback(\mathcal{P})$ (Sec. 4.3.3)

9: $\quad \mathcal{R} \leftarrow \mathcal{R} \cup R$

10: $\quad \delta \boldsymbol{W} = \boldsymbol{W} - \boldsymbol{W}_{prev}$, $\delta \boldsymbol{U} = \boldsymbol{U} - \boldsymbol{U}_{prev}$

11: $\quad \boldsymbol{U}_{prev} = \boldsymbol{U}$, $\boldsymbol{W}_{prev} = \boldsymbol{W}$

12: **end while**

4.3.1 Discriminative Semantic Embedding

To detect patterns that can be translated into semantic queries, we use a manifold embedding approach, where both the data points (features) and the semantic entities (category labels) are embedded as points on a manifold. The semantic queries are asked and the answers refine the manifold. Both the detection of the relations and categorization will be done on this manifold [66]. With the relational semantics, the manifold is discriminatively learned on a large margin loss function.

Formally, we want to embed both the image features $\boldsymbol{x}_i$ and corresponding

class labels y_i into a common low-dimensional space such that the projection of $\boldsymbol{x}_i$, denoted as $\boldsymbol{z}_i$, is closer to the corresponding category embedding $\boldsymbol{u}_{y_i}$ than the embeddings for all the other categories. This is accomplished by constructing a linear projection $\boldsymbol{W} \in \mathbb{R}^{m \times d}$ such that $\boldsymbol{z}_i = \boldsymbol{W}\boldsymbol{x}_i$, and $\|\boldsymbol{W}\boldsymbol{x}_i - \boldsymbol{u}_{y_i}\|_2^2 + 1 \leq \|\boldsymbol{W}\boldsymbol{x}_i - \boldsymbol{u}_c\|_2^2$, $\forall c \neq y_i$.

For knowledge transfer, we first build a reference model with well-defined *anchor* classes. Then we build a model on the *target* classes by transferring semantic information from the *anchor* classes.

4.3.1.1 Semantic embedding for Anchor classes

The desired objective for categorizing semantic embeddings in the *anchor* classes can be expressed as minimization of the large-margin constraints above for all anchor class instances indexed by $i \in \{1, \ldots, N^A\}$ with respect to $\boldsymbol{W}^A$ and prototypes $\boldsymbol{u}_c$:

$$\begin{aligned} &\min_{\boldsymbol{W}^A, \boldsymbol{U}^A} \sum_{i=1}^{N^A} \sum_{c \in \mathcal{C}^A} \mathcal{L}\left(\boldsymbol{W}^A, \boldsymbol{x}_i, \boldsymbol{u}_c\right) + \lambda_1 \|\boldsymbol{W}^A\|_F^2 + \lambda_2 \|\boldsymbol{U}^A\|_F^2, \\ &\text{s.t.} \quad \mathcal{L}(\boldsymbol{W}^A, \boldsymbol{x}_i, \boldsymbol{u}_c) \\ &\qquad = \max(\|\boldsymbol{W}^A\boldsymbol{x}_i - \boldsymbol{u}_{y_i}\|_2^2 - \|\boldsymbol{W}^A\boldsymbol{x}_i - \boldsymbol{u}_c\|_2^2 + 1, 0), \\ &\qquad \forall i, \forall c \neq y_i, \end{aligned} \tag{4.1}$$

where N^A is number of training samples in anchor classes ($\mathcal{C}^A$), $\boldsymbol{U}^A$ is a column stacked matrix of label prototypes $\{\boldsymbol{u}_c\}$ of the anchor classes and λ_1 and λ_2 are hyperparameters for scale regularization terms; $\|\cdot\|_F$ refers to a Frobenius norm.

4.3.1.2 Knowledge Transfer via Relational Semantics

From the learned *anchor* class categorization model with $\boldsymbol{W}^A$ and $\boldsymbol{U}^A$, we transfer the knowledge to the *target* classes that have only a few training samples. Specifically, we use interactively provided semantic relationship $R \in \mathcal{R}$ to regularize the objective function. Formally, learning the discriminative embeddings of target classes can be achieved by solving the following regularized optimization problem:

$$\begin{aligned} &\min_{\boldsymbol{W},\boldsymbol{U}} \sum_{i=1}^{N^T} \sum_{c \in \mathcal{C}^T} \mathcal{L}\left(\boldsymbol{W}, \boldsymbol{x}_i, \boldsymbol{u}_c\right) + \lambda_1 \|\boldsymbol{W}\|_F^2 + \lambda_2 \|\boldsymbol{U}\|_F^2 + \lambda_3 \|\boldsymbol{W} - \boldsymbol{W}^A\|_F^2 + \gamma \sum_j \Omega\left(R_j, \boldsymbol{U}\right), \\ &\quad \text{s.t. } R_j \subset \mathcal{R}, \\ &\quad \mathcal{L}(\boldsymbol{W}, \boldsymbol{x}_i, \boldsymbol{u}_c) = \max(\|\boldsymbol{W}\boldsymbol{x}_i - \boldsymbol{u}_{y_i}\|_2^2 - \|\boldsymbol{W}\boldsymbol{x}_i - \boldsymbol{u}_c\|_2^2 + 1, 0), \forall i, \forall c \neq y_i, \end{aligned} \tag{4.2}$$

where N^T is number of training samples in target classes ($\mathcal{C}^T$), R_j is a subset of $\mathcal{R}$ (the set containing all semantic constraints), and $\boldsymbol{U} = [\boldsymbol{U}^A, \boldsymbol{U}^T]$ is a concatenation of all class prototypes. We regularize the data embeddingt $\boldsymbol{W}$ with $\boldsymbol{W}^A$, and semantic embedding with $\Omega(R_j, \boldsymbol{U})$, which is a regularizer defined on the relationship R_j.

4.3.1.3 Encoding Relational-Semantics by Geometric Topologies

The semantic relationships are used to regularize the embedding space for better classification generalization [25, 39]. As mentioned previously, we use the *triplet-based relationships* in which human feedback is of the form of 'object a is more similar to b than to c'. Triplet-based relationships have many desired properties such as less need to reconcile feedback scale since it is a relative relationship [30,60]. Even though the relationships are local with respect to the associated entities, solving the optimization using the relationships, Eq.(4.2), changes the topology of the class

prototype embeddings globally, which results in a semantically more meaningful model overall.

Triplet-based Relationship. Suppose an target entity, $\boldsymbol{u}_t$, is semantically closer to the anchor entity $\boldsymbol{u}_{a_1}$ than to another anchor entity $\boldsymbol{u}_{a_2}$; we denote such relationship by $R = (t, (a_1, a_2))$ and define its geometric regularizer as a hinge loss type of regularizer that encourages moving $\boldsymbol{u}_t$ closer to $\boldsymbol{u}_{a_1}$ and farther from $\boldsymbol{u}_{a_2}$:

$$\max\left(1 - \|\boldsymbol{u}_{a_2} - \boldsymbol{u}_t\|_2^2 / \|\boldsymbol{u}_{a_1} - \boldsymbol{u}_t\|_2^2, 0\right). \tag{4.3}$$

Eq.(4.3), however, is neither differentiable nor convex in terms of $\boldsymbol{u}_*$'s thus makes the optimization difficult if it is used as a regularization term. So, we relax the regularizer by introducing a scaling constant σ_1 as a proxy of $\|\boldsymbol{u}_{a_1} - \boldsymbol{u}_t\|_2^2$ by a distance between the sample mean of class a_1 and t. In addition, the $\max(x, 0)$ is not continuous at $x = 0$ thus not differentiable. So, we use a differentiable smooth proxy of the $\max(x, 0)$ function,$h_\rho(\cdot)$, to make the regularizer differentiable everywhere:

$$\Omega(R, \boldsymbol{U}) = \sigma_1 h_\rho\left(\|\boldsymbol{u}_{a_1} - \boldsymbol{u}_t\|_2^2 - \|\boldsymbol{u}_{a_2} - \boldsymbol{u}_t\|_2^2\right), \tag{4.4}$$

where the $h_\rho(x)$ is a differentiable proxy for $\max(x, 0)$ as in [2]. A detailed description of $h_\rho(\cdot)$ is as following:

In order to use the gradient descent optimization method at the peak points, we approximate them by smoothed versions as shown by the blue curves in Fig. 4.5 as in [2].

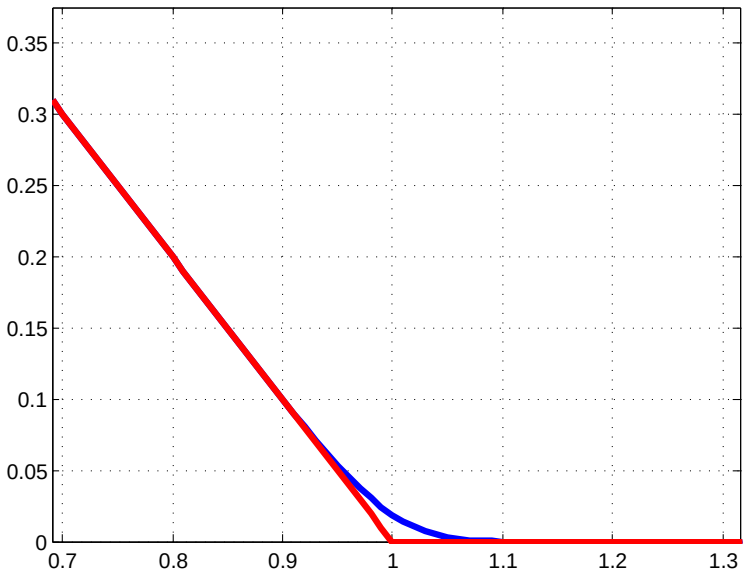


Figure 4.2: **Smoothed hinge loss.**

$h_\rho(\cdot)$ is the approximate hinge loss function that has no discontinuity:

$$h_\rho(z) = \begin{cases} 1 - z & z < 1 - \rho \\ \frac{-(1-z)^4}{16\rho^3} + \frac{3(1-z)^2}{8\rho} + \frac{(1-z)}{2} + \frac{3\rho}{16} & |1 - z| \leq \rho, \\ 0 & z > 1 + \rho. \end{cases} \tag{4.5}$$

and its derivative with respect to z is

$$\frac{\partial h_\rho(z)}{\partial z} = \begin{cases} -1 & z < 1 - \rho \\ \frac{(1-z)^3}{4\rho^3} - \frac{3(1-z)}{4\rho} - \frac{1}{2} & |1 - z| \leq \rho. \\ 0 & z > 1 + \rho. \end{cases} \tag{4.6}$$

In our experiments we use $\rho = \sigma = 10^{-7}$.

4.3.1.4 Numerical Optimization

The optimization problems in Eq. (4.1) and Eq. (4.2) are not jointly convex on $\boldsymbol{W}$ and $\boldsymbol{U}$, but are bi-convex in terms of each variable. We use alternating optimization to solve the problem, where we alternate between the optimization of $\boldsymbol{W}$ and $\boldsymbol{U}$ while fixing the other. We use stochastic sub-gradient method to optimize for each variable.

4.3.2 What Questions to Ask First?

To reduce the number of semantic relationships in the regularizer, while aiming for better classification, we discover candidate semantic questions that are helpful for improving classification accuracy.

4.3.2.1 Generating a Pool of Queries

We first generate a pool of candidate triplet-based semantic relationships; $\mathcal{R} = \{R|R = (t, (a_1, a_2))\}$. R has three entities; target, $\boldsymbol{u}_t$, and two anchors $(\boldsymbol{u}_{a_1}, \boldsymbol{u}_{a_2})$. We want to improve the classification of the target entity by transferring knowledge from the anchor entities, that are highly confident in classification. To generate the pool of triplets, we find the target entities that are highly confused (*i.e.*, classification accuracy in the current model is low) and the anchor entities that are highly confident (*i.e.*, classification accuracy in the current model is high).

Specifically, for each $R = (t, (a_1, a_2))$, we define a scoring function, $S(R, \boldsymbol{U})$, for mining semantic relationship by favoring the most confusing (the least confident)

target entity and the least confusing (the most confident) anchor entities. For the measure of confusion of each entity, we regard each entity as a random variable for class label and use its entropy, $H(\boldsymbol{u}_c)$. The higher the entropy, the higher the confusion. We then define the scoring function as the conditional entropy of a target entity, $\boldsymbol{u}_t$, given anchor entities $(\boldsymbol{u}_{a_1}, \boldsymbol{u}_{a_2})$ as:

$$\begin{aligned} S(R, \boldsymbol{U}) &= H(\boldsymbol{u}_t | \boldsymbol{u}_{a_1}, \boldsymbol{u}_{a_2}) \\ &= H(\boldsymbol{u}_{t_1}, \boldsymbol{u}_{a_1}, \boldsymbol{u}_{a_2}) - H(\boldsymbol{u}_{a_1}, \boldsymbol{u}_{a_2}), \end{aligned} \tag{4.7}$$

Given the label of the target entity $\boldsymbol{u}_t$ of the candidate relationship R, we want the anchor entities to be even more certain. In other words, we assume the uncertainty of anchor entities given the target entity label, $H(\boldsymbol{u}_{a_1}, \boldsymbol{u}_{a_2} | \boldsymbol{u}_t)$, is 0. Then, we can reduce (4.7) to:

$$S(R, \boldsymbol{U}) = H(\boldsymbol{u}_t) - H(\boldsymbol{u}_{a_1}, \boldsymbol{u}_{a_2}). \tag{4.8}$$

Intuitively, the score favors choosing target entities that have high classification confusion and the anchor entities that have low classification confusion. Detailed descriptions of how to compute the probability mass function for the entropy, and the derivation of the conditional entropy function are described in the following subsections.

4.3.2.2 Probability Mass Function

To compute the score by the entropy (Sec.4.3.2.5), we define each entity's probability mass function by its classification confusion on validation set. Specifically,

the probability of a label entity $\boldsymbol{u}_i$ to be a class label j is definedas:

$$P_{\boldsymbol{u}_i}(j) = \frac{\sum_{\boldsymbol{x}_k \in \mathcal{V}} \mathbb{1}(g(\boldsymbol{x}_k) = j)}{|\mathcal{V}|}, \tag{4.9}$$

where $g(\cdot)$ is the current classification model learned with $\boldsymbol{u}_i$ and $\boldsymbol{x}_k$ and $\mathcal{V}$ is a set of feature embeddings, $g(\boldsymbol{x}_k)$, in validation set. Thus $\mathbb{1}(g(\boldsymbol{z}_k) = j)$ equals to the number of feature embeddings whose obtained label by the current model is j. $|\cdot|$ denotes cardinality of a set. The ideal PMF is a delta function when $c = j$; $\delta(c = j)$. Note that the measure depends on the sample distribution under the current model. Thus, the entropy of an entity can be written as:

$$H(\boldsymbol{u}_i) = -\sum_{j \in \mathcal{C}} P_{\boldsymbol{u}_i}(j) \log P_{\boldsymbol{u}_i}(j), \tag{4.10}$$

where $\mathcal{C}$ is a set of all class labels. For the joint entropy, we need to derive a joint probability mass function of multiple label entities.

4.3.2.3 Joint Probability Mass Function of Multiple Entities

For computing a joint entropy, deriving a joint probability mass function (PMF) of multiple entities from Eq.(4.9) is straightforward. We start from the joint PMF of two entities, $P_{\boldsymbol{u}_i,\boldsymbol{u}_j}(c_1, c_2)$. Since the probability of $\boldsymbol{u}_i$ being label c_1 is dependent on the obtained labels of neighboring feature embeddings, $\boldsymbol{z}_1, \cdots, \boldsymbol{z}_N$, $P_{\boldsymbol{u}_i}(c_1)$ is actually a conditional probability as:

$$\begin{aligned} P_{\boldsymbol{u}_i}(c_1) &= P_{\boldsymbol{u}_i}(c_1 | \boldsymbol{z}_1, \ldots, \boldsymbol{z}_N) \\ &= P_{\boldsymbol{u}_i}(c_1 | \{\boldsymbol{z}_k | \boldsymbol{z}_k \in \mathcal{N}^i\}). \end{aligned} \tag{4.11}$$

We can write the joint PMF of $\boldsymbol{u}_i$ and $\boldsymbol{u}_j$ as:

$$\begin{aligned} P_{\boldsymbol{u}_i,\boldsymbol{u}_j}(c_1,c_2) &= P_{\boldsymbol{u}_i|\boldsymbol{u}_j}(c_1|c_2)P_{\boldsymbol{u}_j}(c_2) \\ &= P_{\boldsymbol{u}_i|\boldsymbol{u}_j}(c_1|\{\boldsymbol{z}_k|\boldsymbol{z}_k \in \{\mathcal{N}^i \cup \mathcal{N}^j\}\}, c_2)P_{\boldsymbol{u}_j}(c_2|\{\boldsymbol{z}_k|\boldsymbol{z}_k \in \{\mathcal{N}^i \cup \mathcal{N}^j\}\}) \\ &= P_{\boldsymbol{u}_i|\boldsymbol{u}_j}(c_1|\{\boldsymbol{z}_k|\boldsymbol{z}_k \in \{\mathcal{N}^i - \mathcal{N}^j\} \cup c_2\})P_{\boldsymbol{u}_j}(c_2|\{\boldsymbol{z}_k|\boldsymbol{z}_k \in \mathcal{N}^j\}) \\ &= P_{\boldsymbol{u}_i|\boldsymbol{u}_j}(c_1|\{\boldsymbol{z}_k|\boldsymbol{z}_k \in \{\mathcal{N}^i - \mathcal{N}^j\} \cup c_2\})P_{\boldsymbol{u}_j}(c_2), \end{aligned} \tag{4.12}$$

the second to third line is because if $\boldsymbol{u}_j$ is given (or known), $\mathcal{N}^j$ are not necessary as conditioned variables; $P_{\boldsymbol{u}_i|\boldsymbol{u}_j}(c_1|\{\boldsymbol{z}_k|\boldsymbol{z}_k \in \{\mathcal{N}^i \cup \mathcal{N}^j\}\}, c_2) = P_{\boldsymbol{u}_i|\boldsymbol{u}_j}(c_1|\{\boldsymbol{z}_k|\boldsymbol{z}_k \in \{\mathcal{N}^i - \mathcal{N}^j\} \cup c_2\})$. Then the conditional probability of $P_{\boldsymbol{u}_i|\boldsymbol{u}_j}(c_1|c_2)$ and the joint probability of $(\boldsymbol{u}_i, \boldsymbol{u}_j)$ can be written as:

$$P_{\boldsymbol{u}_i|\boldsymbol{u}_j}(c_1|c_2) = \frac{\left(\sum_{\boldsymbol{z}_i \in \mathcal{N}^i - \mathcal{N}^j} \mathbb{1}(g(\boldsymbol{z}_i) = c_1)\right) + \mathbb{1}(c_1 = c_2)}{|\mathcal{N}^i - \mathcal{N}^j| + 1} \tag{4.13}$$

$$P_{\boldsymbol{u}_i,\boldsymbol{u}_j}(c_1,c_2) = \frac{\left(\sum_{\boldsymbol{z}_i \in \mathcal{N}^i - \mathcal{N}^j} \mathbb{1}(g(\boldsymbol{z}_i) = c_1)\right) + \mathbb{1}(c_1 = c_2)}{|\mathcal{N}^i - \mathcal{N}^j| + 1} \cdot \frac{\sum_{\boldsymbol{z}_i \in \mathcal{N}^j} \mathbb{1}(g(\boldsymbol{z}_i) = c_2)}{|\mathcal{N}^j|}. \tag{4.14}$$

A joint PMF of more than three variables can be straightforwardly obtained by the chain rule.

4.3.2.4 Conditional Entropy

By the independence of variable for conditional entropy, we have the following equation:

$$\begin{aligned} & H(\boldsymbol{u}_{a_1}, \ldots, \boldsymbol{u}_{a_k} | \boldsymbol{u}_{t_1}, \ldots, \boldsymbol{u}_{t_g}) \\ & = H(\boldsymbol{u}_{a_1}, \ldots, \boldsymbol{u}_{a_k}, \boldsymbol{u}_{t_1}, \ldots, \boldsymbol{u}_{t_g}) - H(\boldsymbol{u}_{t_1}, \ldots, \boldsymbol{u}_{t_g}) = 0, \\ & H(\boldsymbol{u}_{a_1}, \ldots, \boldsymbol{u}_{a_k}, \boldsymbol{u}_{t_1}, \ldots, \boldsymbol{u}_{t_g}) = H(\boldsymbol{u}_{t_1}, \ldots, \boldsymbol{u}_{t_g}). \end{aligned} \tag{4.15}$$

Using Eq.(4.15) here, we can derive Eq.(6) in the main paper from Eq.(5) in the main paper as:

$$\begin{aligned} S(R, \boldsymbol{U}) &= H(\boldsymbol{u}_t, \boldsymbol{u}_{a_1}, \boldsymbol{u}_{a_2}) - H(\boldsymbol{u}_{a_1}, \boldsymbol{u}_{a_2}) \\ &= H(\boldsymbol{u}_t) - H(\boldsymbol{u}_{a_1}, \boldsymbol{u}_{a_2}). \end{aligned} \tag{4.16}$$

4.3.2.5 Scoring Metric to Prioritize the Queries

Given the pool of queries, we prioritize the queries to reduce the number of questions to be answered for efficiency. Note that in the interactive setting, in principle, it is optimal to ask one question at a time. However, this can be expensive as it requires frequent re-training of the model. An alternative is to ask mini-batch of questions at a time. In both cases the scoring scheme is crucial for picking one (or a few) most useful questions from the pool to improve the quality of the knowledge transfer. We tested a number of metrics for prioritizing queries.

Entropy Based Score. Entropy based score uses the scores computed in the pool generation process to prioritize the queries (Eq.(4.16)). Although this metric is good

for generating a potential set of queries that could improve accuracy the most, it cannot directly predict the accuracy improvement from enforcement of the corresponding relational semantics. For example, when *Deer* is the confused target class and *Elephant* and *Killer Whale* are confident anchor classes, the entropy is going to be high, but the actual accuracy improvement that may result by enforcing the relational semantics of *Deer* is closer to *Elephant* than *Killer Whale* may not be.

Classification Accuracy. To obtain a good scoring function of the relational semantics, we use classification accuracy of each candidate constraint computed using a validation set. Validation set accuracy is the most direct prediction of expected classification gain when a certain relation is used. Further, since we only order questions within a pool of small number of queries, this is still computationally viable (not so if we would have considered all possible semantic relationships as the pool).

Predicting the Classification Accuracy by a Regression Model. Computing the classification accuracy of each constraint even within the pool at every iteration is still computationally expensive, we can approximate it by regressing over multiple types of *features*, which is a proxy for estimating the classification improvement by a vector of various scores ($\boldsymbol{c}$) to the validation accuracy (s). Suppose the relationship consists of target class t and two anchor classes a_1 and a_2. We use a score vector to estimate the validation accuracy. The score vector $\boldsymbol{c}$ consists of confidence/confusion of t, a_1 and a_2 on both training set and validation set, geometric fitness $\left(\frac{\|\boldsymbol{u}_{a_2}-\boldsymbol{u}_t\|^2}{\|\boldsymbol{u}_{a_1}-\boldsymbol{u}_t\|^2}\right)$ and ball radius of sample distribution with respect to each class label prototype for t, a_1 and

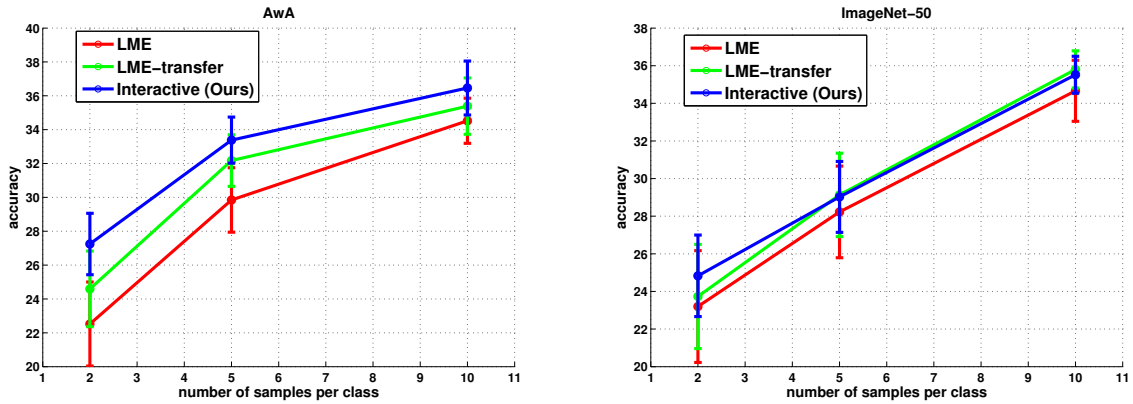


Figure 4.3: Classification performance on AWA and ImageNet-50 dataset. Results are average accuracies over five random splits with standard error shown at 95% confidence interval.

a_2. Using a set of features ($\boldsymbol{C}$) and corresponding validation accuracies($\boldsymbol{s} = \{s\}$), we obtain a linear regression model, with a bias, by solving $\hat{\boldsymbol{R}} = \arg\min_{\boldsymbol{R}} \|\boldsymbol{R}^T\boldsymbol{C} - \boldsymbol{s}\|_2^2$. Using $\hat{\boldsymbol{R}}$, we can approximate the validation accuracy by extracting the feature described at test time and regressing s.

4.3.3 Feedback

Feedback can be obtained from human expert(s). We simulate human feedback by an oracle that gives answers based on the distance of attribute description. Since the attribute description is an agglomerative score of different criteria from a number of human annotators, it is a reasonable measure for the semantic decision regarding validity of relational queries. Specifically, for each triplet-based relationships, we compute the distance of attribute description of $\boldsymbol{u}_t$ and $\boldsymbol{u}_{a_1}$ and $\boldsymbol{u}_t$ and $\boldsymbol{u}_{a_2}$. If

the distance between $\boldsymbol{u}_t$ and $\boldsymbol{u}_{a_1}$ is smaller than the distance between $\boldsymbol{u}_t$ and $\boldsymbol{u}_{a_2}$, the oracle gives an answer to the system of 'Yes', otherwise 'No'. We only use the relationships that are answered as 'Yes' as constraints.

4.3.4 Interactive Learning

The key to our approach is to adaptively update the query generation. We refer to this as 'interactive' model. So far, we describe the process of one iteration of human interaction. We iterate the process multiple times, updating the embedding manifold (model) and use the updated model to generate a new pool and prioritize the queries for the next iteration. The adaptive query generation and prioritization scheme achieve better classification accuracy with fewer number of relational constraints, as compared to a single iteration model, which we refer to as 'active' model. In other words, interactive model is more efficient in terms of utilizing human feedback.

4.3.5 Computational Complexity

The computational complexity of Algorithm 1 depends on the complexity of training the model on the anchor and target classes, and generating a query pool.

First, the complexity of training the model on the anchor classes in Eq.(4.1) for each $\boldsymbol{W}^{\boldsymbol{A}}$ and $\boldsymbol{U}^{\boldsymbol{A}}$ is $O(md(N^A+1))$ and $O(m(dN^A+C^A))$ respectively, and the complexity of training the model for target categories in Eq.(4.2) is $O(md(N^T+2))$ and $O(m(dN^T+C^T+|\mathcal{R}|))$ for $\boldsymbol{W}$ and $\boldsymbol{U}$. It is dominated by $O(mdN^T)$ as $dN^T \gg$

$C^T + |\mathcal{R}|$.

To generate a query pool (Sec. 4.3.2) takes total of $O(NC + N^A C^A + N^T C^T + k_p C_r^2)$. We first compute the probability mass function (PMF) for each label entity by $O(NC)$, where $N = N^A + N^T, C = C^A + C^T$ and a confusion matrix of label entities by its PMF with the complexity of $O(N^A C^A + N^T C^T)$. A naive way of enumerating all possible constraints takes $O(C^T {C^A}^2)$ but we generate a decent sized subset (k_p) to consider the most confusing entities' nearest neighboring label embeddings (C_r) by $O(k_p C_r^2)$. Thus, the complexity of generating the pool is $O(NC + N^A C^A + N^T C^T + k_p C_r^2)$. Re-scoring the pool using cross validation takes $O(k_p m d N^T)$. Finally, the outer loop of algorithm usually iterates few times and thus the total complexity of Algorithm 1 is $O(N^A(md + C^A) + N^T(k_p md + C^T) + NC + k_p C_r^2)$.

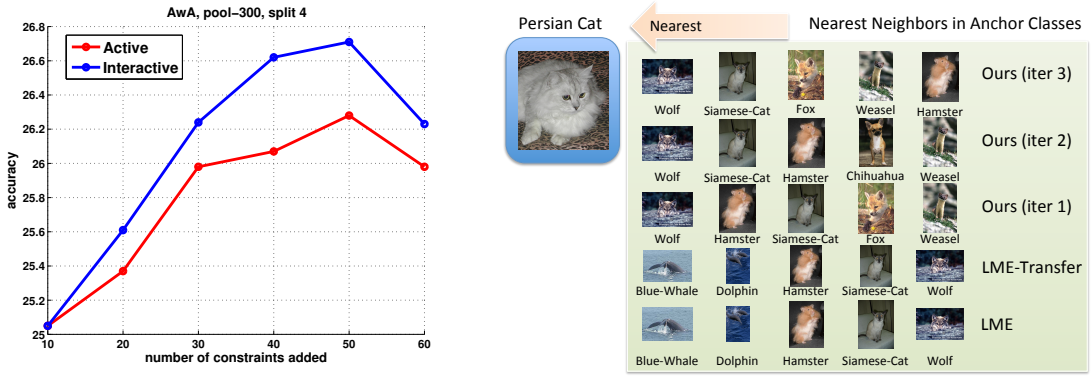
Test time complexity is $O(m(C^T + d))$, which is the same for all linear embedding methods.

4.4 Experiments

4.4.1 Datasets and Experimental Details

We validate our method on two object categorization datasets: 1) Animals with Attributes (AWA) [36], which consists of 50 animal classes and 30,475 images, 2) ImageNet-50 [25], which consists of 70,380 images of 50 categories.

We evaluate the performance of knowledge transfer by classification accuracy on target classes in a challenging set-up that has very small number of training



(a) (b)

Figure 4.4: Effect of Interaction. (a) Classification accuracy as a function of number of constraints added by active or interactive scoring. (b) Qualitative result of nearest neighbor of target class.

samples (2, 5 and 10 samples per class, few-shot learning) with a prior learned with anchor classes that have comparatively more numbers of training samples (30 samples per classes). For testing and validation set, we use a 50/50 split of remaining samples, excluding the training samples. In both datasets, we use 40 classes as anchor classes and 10 classes as target classes. We configure the anchor/target classes, following the configuration of training/test classes in zero-shot/few-shot learning set-up in [38].

Low-Level Features: The low-level features of both dataset is SIFT and other texture and color descriptors with PCA, provided by dataset authors [25, 38]. In AWA dataset, we do PCA to reduce the dimensions to 300. In ImageNet-50, we use 1000 dimensional feature of same type of low level description to AWA dataset. We center the features by the sample mean. For dimension of the embedding space, we choose

75, which is slightly bigger than the number classes (50) for encoding additional semantic information.

Animals with Attribute (AwA): There are 50 classes in total in AwA dataset [38]. Ten of them are target classes. The target classes of AwA dataset are 'Leopard', 'Pig', 'Hippopotamus', 'Seal', 'Persian Cat', 'Chimpanzee', 'Rat', 'Humpback Whale', 'Giant Panda' and 'Racoon'. The rest of the 40 classes of AwA serves as anchor classes.

ImageNet-50: There are 50 classes in total in the ImageNet-50 dataset [25]. The 50 classes are randomly chosen from the entire ImageNet dataset. The 50 classes are: 'Kitfox', 'australianterrier', 'lesserpanda', 'egyptiancat', 'persiancat', 'cougar', 'badger', 'greatdane', 'scottishdeerhound', 'jaguar', 'blackfootedferret', 'skunk', 'corgi', 'weasel', 'colobus', 'orangutan', 'chimpanzee', 'gorilla', 'greyhound', 'hare', 'patas', 'baboon', 'macaque', 'tabby', 'raccoon', 'polecat', 'lion', 'cheetah', 'otter', 'sunflower', 'bonsai', 'strawberry', 'lamp', 'pooltable', 'acorn', 'drum', 'marimba', 'daisy', 'comb', 'rule', 'ferriswheel', 'rollercoaster', 'buckle', 'button', 'barnspider', 'gardenspider', 'bridge', 'featherboa', 'bathtub', 'basketball.'

Among them, we randomly choose ten of them are target classes. The target classes of ImageNet-50 dataset are 'cougar', 'weasel', 'colobus', 'gorilla', 'tabby', 'raccoon', 'pool-table', 'comb', 'roller-coaster', 'feather-boa'. The rest of the 40 classes of ImageNet-50 serves as anchor classes.

4.4.2 Classification Accuracy

Fig. 4.3 shows the classification accuracy on target classes for the two datasets. Our interactive model (Interactive) with the scoring metric described in Sec. 4.4.2.1 outperforms the baseline transfer models (LME-transfer) without semantic constraints and the large margin model without knowledge transfer (LME). Specifically, 'LME' refers to the model learned using Eq.(4.2) with $\lambda_3 = 0, \gamma = 0$, and 'LME-Transfer' refers to the model learned using Eq.(4.2) without the semantic constraints ($\gamma = 0$). For the 'Interactive', we add 20 semantic constraints per iteration and run 5$\sim$6 iterations, so add 100$\sim$120 semantic constraints in total.

Effect of Interaction. Our interactive learning scheme continuously updates the model to select a better set of questions in terms of classification accuracy. We use a mini batch size of 10 for interactive setting. The interactively mined constraints provide better classification accuracy over an equivalent sized set of constraints produced in a batch. Fig. 4.4-(a) shows the classification accuracy as a function of number of constraints added by the iteratively updated model and by a batch model. In both cases same measure for selection and ordering was used. Interestingly, as iterations continue, the accuracy starts to drop. We believe it is because there is not much helpful semantics to be added for classification past that iteration. (similar argument is in the introduction)

As a qualitative result, we present the nearest neighbor of a target class in the anchor set in Fig. 4.4-(b). As baseline models (LME, LME-Transfer) do not explic-

# Iter	Positively answered query at its highest rank
1	|fox - persian cat| < |blue whale - persian cat|
2	|grizzly bear - persian cat| < |horse - persian cat|
3	|dalmatian - persian cat| < |beaver - persian cat|
4	|dalmatian - persian cat| < |german shepherd - persian cat|

Table 4.1: Top Ranked Query as Interaction (Iter) Proceeds. As interactions continue, top ranked query whose target class is 'Persian Cat' becomes semantically more meaningful.

itly enforce the semantic relationships of categories, the nearest neighbors obtained by the baseline models are not semantically meaningful. The nearest neighbors obtained using our model, however, are semantically meaningful from the first iteration onward. As iterations proceed, the nearest neighbor is further refined to be semantically more meaningful, *e.g.*, *Siamese-cat* appears as the second nearest neighbor in the iteration 2 and 3 where as it was a third-nearest neighbor at the first iteration.

As interaction proceeds, the embedding space becomes semantically more meaningful so do the generated queries. Table 4.1 shows top positive query related to *Persian-cat* as a function of iterations. In early iterations, the questions try to relate *Persian-cat* to *fox* and *blue whale*. But in the later iterations, the question becomes more semantically meaningful, comparing *Persian-cat* with *dalmatian* and *german shepherd*.

Dataset	Animals with Attribute			ImageNet-50		
# samples/class	2	5	10	2	5	10
LME	22.51±2.48	29.85±1.90	34.52±1.33	23.20±2.97	28.22±2.43	34.67±1.62
LME-Transfer	24.59±2.23	32.17±1.53	35.39±1.67	23.47±2.66	28.78±2.05	34.94±1.03
Random	24.75±2.11	31.32±1.31	35.96±1.66	24.23±1.92	28.72±2.26	34.74±2.26
Entropy	24.96±2.24	31.81±1.27	35.92±1.91	24.60±2.80	28.88±2.43	35.64±0.99
Active-Regression	25.43±1.90	32.49±1.58	36.18±0.88	23.34±2.76	28.99±2.34	35.49±0.89
Active	26.62±1.67	32.42±1.45	36.40±1.33	24.35±2.42	28.55±2.07	35.60±1.01
Interactive	**27.24±1.82**	**33.31±1.28**	**36.46±1.60**	**24.95±2.20**	**29.08±1.88**	**35.62±1.01**
Interactive-UB	28.57±1.85	33.61±2.15	36.86±1.83	25.15±2.13	29.23±1.85	35.95±1.53

Table 4.2: Classification Accuracy (%) for Comparing Quality of Scoring Function. For ease of comparison, we provide two baselines of the method (LME and LME-Transfer) and the upper-bound of our interactive model (Interactive-UB), which is obtained by adding constraints scored by the test set.

4.4.2.1 Comparison Among Query-Scoring Metrics

Scoring metric for query is one of the most important components in the interactive framework. In Table 4.2, we compare the accuracy obtained by adding the constraints by the various scoring schemes that we have presented in Sec. 4.3.2.5. Number of constraints added and other hyper-parameters are determined by cross validation. 'Random'–random ordering of query from the selected pool. 'Entropy'–Entropy-based scores. 'Active'–classification accuracy based score by a batch-mode model. 'Active-Regression'–regressed score of the classification accuracy obtained by a batch-mode model. 'Interactive'–classification accuracy based score by an adaptively updated model, which is our proposal. 'Interactive-UB' refers to a upper

bound that our framework can achieve; we score and add the queries based on classification accuracy with test set itself in our interactive model. Note that except 'Interactive', all other scoring metrics are in a batch-mode. The interactive model outperforms the batch mode model, which we denote as 'Active', and other scoring schemes, and is tight to the upper bound. We also present the baseline results of 'LME' and 'LME-Transfer' for reference.

Note that all methods use the same validation set to tune parameters. Our scoring metric in 'Active' and 'Interactive', in addition, uses it to prioritize queries to the user as this is the most direct way to measure the effect of adding a particular constraint on the recognition accuracy without using the testing set. While this perhaps makes direct comparison to the baselines slightly less transparent, the comparison of 'Active' and 'Interactive' variants, which both use this criterion, clearly points to the fact that 'Interactive' learning is much more effective in selecting and ordering of constraints.

4.5 Conclusion

We propose an interactive learning framework that takes human feedback to iteratively refine the learned model. Our method detects recurring relational patterns from a semantic manifold and translate them into semantic queries to be answered and retrain the model by imposing the constraints obtained by positively feed-backed semantic relationships. We validate our method against batch learning methods on classification accuracy of target classes with transferred knowledge from

anchor classes via relational semantics.

Chapter 5: Conclusion

We have explored several methods to improve the visual category recognition in various scenarios.

As there are large pools of unlabeled image sources readily available on the web, we propose a method to add quality samples from external sources to object categories by learned attributes with the minimal human supervision. Unlike conventional semi-supervised learning (SSL) methods that only use a single visual feature space, our method utilizes two different visual representations to discover quality samples. The added samples capture the commonality and diversity of the given labeled samples of each visual categories. The expanded set improves the classification accuracy over the baselines significantly.

When the unlabeled samples are not available, we need to build a better classification model to improve the classification accuracy by exploiting the given labeled samples. We propose to build an ensemble of classifiers that incorporate both diverse specificity and commonality of the subcategories. First, we discover the subcategories that are discriminative to the other categories. The set of classifiers captures the diversity of each discovered subcategory. Then we force the set of classifiers to share the common characteristic of the subcategories by minimizing

the rank of a matrix of classifiers. The new set of classifiers significantly outperform the baselines and the state-of-the-art generalizable classifiers by a noticeable margin, especially in difficult categories.

When semantic information of the category definitions is available, we could use an interactive semantic transfer learning formulation that exploits relational commonality and diversity of category definitions in an efficient manner. The discriminative classification model identifies the most helpful relational semantic queries and the semantic feedbacks refines the model in the form of regularization in a few iterations. The refined model improves the classification accuracy on the challenging categories that has only a few number of training samples.

However, the proposed methods have following limitations. The learned attribute based SSL method has a risk of semantic drift of adding unrelated samples to categories in a long run as all SSL methods have. The learning information shared classifier does not scale with the size of the labeled training set. The sparse semantic information in the third method does not improve the accuracy dramatically. Addressing these issues can be interesting research directions for building better visual recognition systems.

Bibliography

[1] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid. Label-Embedding for Attribute-Based Classification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[2] Y. Amit, M. Fink, N. Srebro, and S. Ullman. Uncovering Shared Structures in Multiclass Classification. In *International Conference on Machine Learning (ICML)*, 2007.

[3] S. Andrews, I. Tsochantaridis, and T. Hofmann. Support Vector Machines for Multiple-Instance Learning. In *NIPS*, 2003.

[4] A. Argyriou, T. Evgeniou, and M. Pontil. Convex Multi-Task Feature Learning. *Machine Learning*, (Special Issue on Inductive Transfer Learning):243–272, 2008.

[5] S. Bengio, J. Weston, and D. Grangier. Label Embedding Trees for Large Multi-Class Tasks. In *NIPS*, 2010.

[6] T. L. Berg, A. Sorokin, G. Wang, D. A. Forsyth, D. Hoiem, A. Farhadi, and I. Endres. It's All About the Data. In *Proceedings of the IEEE, Special Issue on Internet Vision*, 2010.

[7] A. Bergamo and L. Torresani. Meta-Class Features for Large-Scale Object Categorization on a Budget. In *CVPR*, 2012.

[8] M. Bilgic, L. Mihalkova, and L. Getoor. Active learning for networked data. In *International Conference on Machine Learning (ICML)*, 2010.

[9] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, 2(2):121–167, 1998.

[10] R. Caruana. Multitask Learning. In *Machine Learning*, volume 28, pages 41–75, 1997.

[11] J. Choi, M. Rastegari, A. Farhadi, and L. S. Davis. Adding unlabeled samples to a category by learned attributes. In *CVPR*, 2013.

[12] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009.

[13] E. Eaton, G. Holness, and D. McFarlane. Interactive learning using manifold geometry. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2010.

[14] E. Eaton and P. L. Ruvolo. ELLA: An efficient lifelong learning algorithm. In *International Conference on Machine Learning (ICML)*, pages 507–515, 2013.

[15] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes (VOC) Challenge. *IJCV*, 2010.

[16] R. E. Fan, K. W. Chang, C. J. Hsieh, X. R. Wang, and C. J. Lin. LIBLINEAR: A library for large linear classification. *JMLR*, 2008.

[17] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing Objects by their Attributes. In *CVPR*, 2009.

[18] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object Detection with Discriminatively Trained Part-Based Models. *PAMI*, 32(9):1627–1645, 2010.

[19] R. Fergus, Y. Weiss, and A. Torralba. Semi-supervised Learning in Gigantic Image Collections. In *NIPS*, 2009.

[20] H. Grabner, J. Gall, and L. Van Gool. What makes a Chair a Chair? In *CVPR*, 2011.

[21] G. Griffin, A. Holub, and P. Perona. Caltech-256 Object Category Dataset. Technical report, 2007.

[22] Z. Harchaoui, M. Douze, M. Paulin, M. Dud'ık, and J. Malick. Large-scale image classification with trace-norm regularization. In *CVPR*, 2012.

[23] T. Hastie, R. Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning*. Second edition, 2003.

[24] M. Hoai and A. Zisserman. Discriminative Sub-categorization. In *CVPR*, 2013.

[25] S. J. Hwang, K. Grauman, and F. Sha. Analogy-preserving semantic embedding for visual object categorization. In *International Conference on Machine Learning (ICML)*, pages 639–647, 2013.

[26] S. J. Hwang, F. Sha, and K. Grauman. Sharing features between objects and their attributes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1761–1768, 2011.

[27] P. Jain, S. Vijayanarasimhan, and K. Grauman. Hashing Hyperplane Queries to Near Points with Applications to Large-Scale Active Learning. In *NIPS*, 2010.

[28] Y. Jia. Caffe: An open source convolutional architecture for fast feature embedding.
urlhttp://caffe.berkeleyvision.org/, 2013.

[29] A. Karbasi, S. Ioannidis, and L. Massoulie. Comparison-based learning with rank nets. In *International Conference on Machine Learning (ICML)*, pages 855–862, 2012.

[30] M. Kendall and J. D. Gibbons. *Rank Correlation Methods*. 5 edition, 1990.

[31] J. Kim and K. Grauman. Shape Sharing for Object Segmentation. In *ECCV*, 2012.

[32] A. Kovashka, S. Vijayanarasimhan, and K. Grauman. Actively selecting annotations among objects and attributes. *IEEE International Conference on Computer Vision (ICCV)*, pages 1403–1410, 2011.

[33] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *NIPS*, 2012.

[34] M. P. Kumar, B. Packer, and D. Koller. Self-Paced Learning for Latent Variable Models. In *NIPS*, 2010.

[35] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar. Attribute and Simile Classifiers for Face Verification. In *ICCV*, 2009.

[36] C. Lampert, H. Nickisch, and S. Harmeling. Learning to Detect Unseen Object Classes by Between-Class Attribute Transfer. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

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

[38] C. H. Lampert, H. Nickisch, and S. Harmeling. Attribute-Based Classification for Zero-Shot Visual Object Categorization. *IEEE Trans. on PAMI*, 2014.

[39] M. T. Law, N. Thome, and M. Cord. Quadruplet-wise Image Similarity Learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[40] Y. J. Lee and K. Grauman. Learning the Easy Things First: Self-Paced Visual Category Discovery. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[41] J. J. Lim, R. Salakhutdinov, and A. Torralba. Transfer Learning by Borrowing Examples for Multiclass Object Detection. In *NIPS*, 2011.

[42] N. Loeff and A. Farhadi. Scene Discovery by Matrix Factorization. In *ECCV*, 2008.

[43] N. Loeff, A. Farhadi, I. Endres, and D. A. Forsyth. Unlabeled Data Improves Word Prediction. In *ICCV*, 2009.

[44] J. B. MacQueen. Some Methods for classification and Analysis of Multivariate Observations. In *Proc. of 5th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967.

[45] T. Malisiewicz, A. Gupta, and A. A. Efros. Ensemble of Exemplar-SVMs for Object Detection and Beyond. In *ICCV*, 2011.

[46] M. Marszalek and C. Schmid. Constructing category hierarchies for visual recognition. In *European Conference on Computer Vision (ECCV)*, 2008.

[47] P. Matikainen, R. Sukthankar, and M. Hebert. Classifier Ensemble Recommendation. In *ECCV Workshop on Web-scale Vision and Social Media*, 2012.

[48] T. Mikolov, W. Tau Yih, and G. Zweig. Linguistic regularities in continuous space word representations. In *HLT-NAACL*, pages 746–751, 2013.

[49] B. Mishra, G. Meyer, F. Bach, and R. Sepulchre. Low-rank optimization with trace norm penalty. In *ACM CoRR*, 2011.

[50] S. J. Pan and Q. Yang. A Survey on Transfer Learning. *IEEE Trans. on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

[51] D. Parikh and K. Grauman. Interactively building a discriminative vocabulary of nameable attributes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1681–1688, 2011.

[52] A. Parkash and D. Parikh. Attributes for Classifier Feedback. In *ECCV*, 2012.

[53] G.-J. Qi, C. Aggarwal, Y. Rui, Q. Tian, S. Chang, and T. Huang. Towards Cross-Category Knowledge Propagation for Learning Visual Concepts. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[54] M. Rastegari, A. Farhadi, and D. Forsyth. Attribute Discovery via Predictable Discriminative Binary Codes. In *ECCV*, 2012.

[55] L. Rokach. Ensemble-based classifiers. *Artificial Intelligence Review*, 33(1-2), 2010.

[56] P. Ruvolo and E. Eaton. Active task selection for lifelong machine learning. In *AAAI Conference on Artificial Intelligence (AAAI)*, July 2013.

[57] R. Salakhutdinov, A. Torralba, and J. Tenenbaum. Learning to Share Visual Appearance for Multiclass Object Detection. In *CVPR*, 2011.

[58] B. Settles. Active Learning Literature Survey. Technical report, 2009.

[59] A. Shrivastava, S. Singh, and A. Gupta. Constrained semi-supervised learning using attributes and comparative attributes. In *ECCV*, 2012.

[60] O. Tamuz, C. Liu, S. Belongie, O. Shamir, and A. T. Kalai. Adaptively Learning the Crowd Kernel. In *International Conference on Machine Learning (ICML)*, 2011.

[61] S. Thrun. A lifelong learning perspective for mobile robot control. In V. Graefe, editor, *Intelligent Robots and Systems*. Elsevier, 1995.

[62] T. Tommasi, F. Orabona, and B. Caputo. Safety in Numbers: Learning Categories from Few Examples with Multi Model Knowledge Transfer. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[63] L. Torresani, M. Szummer, and A. Fitzgibbon. Efficient Object Category Recognition Using Classemes. In *ECCV*, 2010.

[64] L. van der Maaten and K. Weinberger. Stochastic Triplet Embedding. In *IEEE Int'l Workshop on Machine Learning for Signal Processing*, 2012.

[65] A. Vedaldi and A. Zisserman. Efcient Additive Kernels via Explicit Feature Maps. *IEEE Trans. PAMI*, 2011.

[66] K. Weinberger and O. Chapelle. Large Margin Taxonomy Embedding with an Application to Document Categorization. In *NIPS*, 2008.

[67] J. Weston and J. Blitzer. Latent Structured Ranking. In *UAI*, 2012.

[68] J. Wright, A. Ganesh, S. Rao, and Y. Ma. Robust Principal Component Analysis: Exact Recovery of Corrupted Low-Rank Matrices. *CoRR*, abs/0905.0233, 2009.

[69] H. Yang, I. King, and M. R. Lyu. Multi-task Learning for one-class classification. In *Neural Networks (IJCNN), The 2010 International Joint Conference on*, pages 1–8, 2010.

[70] W. Zhang, S. X. Yu, and S.-H. Teng. PowerSVM: Generalization with Exemplar Classification Uncertainty. In *CVPR*, 2012.

[71] D. Zhou, L. Xiao, and M. Wu. Hierarchical Classification via Orthogonal Transfer. In *International Conference on Machine Learning (ICML)*, 2011.

[72] J. Zhou, J. Chen, and J. Ye. *MALSAR: Multi-tAsk Learning via StructurAl Regularization*. Arizona State University, 2011.

[73] X. Zhu. Semi-Supervised Learning Literature Survey. Technical report, 2008.

[74] X. Zhu, D. Anguelov, and D. Ramanan. Capturing long-tail distributions of object subcategories. In *CVPR*, 2014.