

HIERARCHICAL LEARNING OF DISCRIMINATIVE FEATURES AND  
CLASSIFIERS FOR LARGE-SCALE VISUAL RECOGNITION

by

Ning Zhou

A dissertation submitted to the faculty of  
The University of North Carolina at Charlotte  
in partial fulfillment of the requirements  
for the degree of Doctor of Philosophy in  
Computing and Information Systems

Charlotte

2014

Approved by:

---

Dr. Jianping Fan

---

Dr. Aidong Lu

---

Dr. Zbigniew W. Ras

---

Dr. Min C. Shin

---

Dr. Nigel Zheng

©2014  
Ning Zhou  
ALL RIGHTS RESERVED

## ABSTRACT

NING ZHOU. Hierarchical learning of discriminative features and classifiers for large-scale visual recognition. (Under the direction of DR. JIANPING FAN)

Enabling computers to recognize objects present in images has been a long standing but tremendously challenging problem in the field of computer vision for decades. Beyond the difficulties resulting from huge appearance variations, large-scale visual recognition poses unprecedented challenges when the number of visual categories being considered becomes thousands, and the amount of images increases to millions. This dissertation contributes to addressing a number of the challenging issues in large-scale visual recognition.

First, we develop an automatic image-text alignment method to collect massive amounts of labeled images from the Web for training visual concept classifiers. Specifically, we first crawl a large number of cross-media Web pages containing Web images and their auxiliary texts, and then segment them into a collection of image-text pairs. We then show that near-duplicate image clustering according to visual similarity can significantly reduce the uncertainty on the relatedness of Web images' semantics to their auxiliary text terms or phrases. Finally, we empirically demonstrate that random walk over a newly proposed phrase correlation network can help to achieve more precise image-text alignment by refining the relevance scores between Web images and their auxiliary text terms.

Second, we propose a visual tree model to reduce the computational complexity of a large-scale visual recognition system by hierarchically organizing and learning the

classifiers for a large number of visual categories in a tree structure. Compared to previous tree models, such as the label tree, our visual tree model does not require training a huge amount of classifiers in advance which is computationally expensive. However, we experimentally show that the proposed visual tree achieves results that are comparable or even better to other tree models in terms of recognition accuracy and efficiency.

Third, we present a joint dictionary learning (JDL) algorithm which exploits the inter-category visual correlations to learn more discriminative dictionaries for image content representation. Given a group of visually correlated categories, JDL simultaneously learns one common dictionary and multiple category-specific dictionaries to explicitly separate the shared visual atoms from the category-specific ones. We accordingly develop three classification schemes to make full use of the dictionaries learned by JDL for visual content representation in the task of image categorization. Experiments on two image data sets which respectively contain 17 and 1,000 categories demonstrate the effectiveness of the proposed algorithm.

In the last part of the dissertation, we develop a novel data-driven algorithm to quantitatively characterize the semantic gaps of different visual concepts for learning complexity estimation and inference model selection. The semantic gaps are estimated directly in the visual feature space since the visual feature space is the common space for concept classifier training and automatic concept detection. We show that the quantitative characterization of the semantic gaps helps to automatically select more effective inference models for classifier training, which further improves the recognition accuracy rates.



## ACKNOWLEDGMENTS

During the years of my Ph.D. research, I am indebted to many people who have helped me in different aspects.

First of all, I would like to thank my advisor, Prof. Jianping Fan, for his excellent guidance. Also, I would like to thank the other members of my dissertation committee, Prof. Aidong Lu, Prof. Zbigniew W. Ras, Prof. Min C. Shin and Prof. Nigel Zheng, for their helpful suggestions and comments on this work.

I would like to thank Prof. Xiangyang Xue for being an excellent advisor when I was a master student at Fudan University. My sincere thanks also goes to Prof. William K. Cheung at Hong Kong Baptist University and Prof. Guoping Qiu at the University of Nottingham. I have been very fortunate to be supervised by you two during the half year of being an exchange student at HKBU. William, your advice and attitude towards research and life have changed and shaped me in a very positive way.

I am indebted to my wonderful mentors and collaborators during my summer internships. They are Dr. Ser-Nam Lim, Dr. Ting Yu and Dr. Yi Xu at GE Global Research, Dr. Anelia Angelova at Google Research, Dr. Zhu Liu, Dr. Behzad Shahraray and Dr. Eric Zavesky at AT&T Labs Research. It has been a great honor for me to have the opportunities to work with you, and hope our cooperation continues in the future.

Also, I am very thankful to have many great friends, including Liqin Su, Yijing Zhou, Sam Wang, Peter Xu, Jessie Wang, Peixiang Dong, Hao Lei, Jinyue Xia,

Zhiqiang Ma, Chunlei Yang, Yi Shen, Junjie Shan and many many more, who made my time at UNC-Charlotte much more enjoyable.

Finally, I would like to thank my wife, Jiaowei Wu for her sacrifice, understanding, and encouragement throughout the years, my parents and sister for their non-stopping support, my uncles for their visions and helps, and our God for everything.

## TABLE OF CONTENTS

LIST OF FIGURES	xii
LIST OF TABLES	xvii
CHAPTER 1: INTRODUCTION	1
1.1. Contexts	1
1.2. Challenges	2
1.2.1. Collection of Labeled Image Data	4
1.2.2. Efficiency of Classifier Training and Testing	6
1.2.3. Effectiveness of Visual Content Representations	8
1.2.4. Learning Complexity of Visual Concept Classifiers	10
1.3. Contributions	11
CHAPTER 2: RELATED WORK	15
2.1. Image Data Set Construction	15
2.1.1. Manually Labeling	15
2.1.2. Automatic Approach	16
2.2. Multiclass Classifier Learning and Organizing	17
2.2.1. Flat Approach	17
2.2.2. Hierarchical Method	19
2.3. Image Content Representation	20
2.3.1. Global Descriptors	21
2.3.2. Local Descriptors	22
2.3.3. Bag-of-Visual-Words	24

2.3.4. Visual Dictionary Learning	27
2.4. Semantic Gap Modeling	28
CHAPTER 3: AUTOMATIC IMAGE-TEXT ALIGNMENT	33
3.1. Introduction	33
3.2. Automatic Generation of Large-Scale Image-Text Pairs	37
3.2.1. Web Page Crawling and Segmentation	37
3.2.2. Informative Image Extraction	39
3.2.3. Text Phrase Chunking and Ranking	42
3.3. Automatic Image-Text Alignment	46
3.3.1. Near-Duplicate Image Clustering	46
3.3.2. Phrase Aggregation	49
3.3.3. Relevance Re-ranking	51
3.4. Experiments for Algorithm Evaluation	56
3.4.1. Effectiveness of Image Clustering	58
3.4.2. Comparison of Phrase Aggregation Strategies	59
3.4.3. Effectiveness of Relevance Re-ranking	60
3.4.4. Performance on Web Image Indexing and Retrieval	61
3.4.5. Quality of Labeled Training Images	65
3.5. Summary	68
CHAPTER 4: VISUAL TREE FOR EFFICIENT CATEGORIZATION	70
4.1. Introduction	70
4.2. Visual Tree Model	74
4.2.1. Visual Representation of an Image Category	75

4.2.2.	Learning the Structure of a Visual Tree	77
4.2.3.	Learning Node Predictors of a Visual Tree	81
4.2.4.	Label Prediction with a Visual Tree	83
4.3.	Experimental Setup	85
4.4.	Experimental Results	86
4.4.1.	Comparison with other Tree Methods	87
4.4.2.	Evaluation of using Multiple Centroids per Class	92
4.4.3.	Evaluation on Soft Prediction	94
4.5.	Summary	95
CHAPTER 5: DISCRIMINATIVE DICTIONARY LEARNING		97
5.1.	Introduction	97
5.2.	Image Category Clustering for Joint Dictionary Learning	103
5.2.1.	Visual Category Representation	103
5.2.2.	Image Category Clustering	104
5.2.3.	Relation to Label Tree [9]	106
5.3.	Joint Dictionary Learning	107
5.3.1.	Formulation of JDL	107
5.3.2.	Discrimination Promotion Term	108
5.3.3.	Optimization of JDL	110
5.4.	Large-Scale Image Classification	112
5.4.1.	Local Classification Scheme	113
5.4.2.	Global Classification Scheme	114
5.4.3.	Hierarchical Classification Scheme	115

5.5. Experiments	120
5.5.1. Experimental Setup	121
5.5.2. Evaluation on Oxford Flower Image Set	121
5.5.3. Evaluation on ILSVRC2010 Image Set	124
5.5.4. Convergence and Discrimination	132
5.5.5. Dictionary Size	133
5.5.6. Computational Complexity of JDL	134
5.6. Summary	135
CHAPTER 6: SEMANTIC GAP MODELING	137
6.1. Introduction	137
6.2. Feature Extraction and Image Similarity Characterization	140
6.3. Inter-Concept Discrimination Complexity Score	143
6.4. Inner-Concept Visual Homogeneity Score	149
6.5. Quantitative Characterization of Semantic Gaps	150
6.6. Automatic Inference Model Selection for Concept Classifier Training	154
6.7. Algorithm Evaluation and Experimental Results	157
6.7.1. Experimental Results for NUS-WIDE Image Set	158
6.7.2. Experimental Results for ImageNet Image Set	162
6.7.3. Benefits from Semantic Gap Quantification	167
6.8. Summary	171
CHAPTER 7: CONCLUSION	173
7.1. Summary of Conclusions	173

## 7.2. Prospective Directions for Further Research

175

## REFERENCES

178

## LIST OF FIGURES

FIGURE 1: The state-of-the-art image categorization pipeline follows the paradigm of machine learning.	3
FIGURE 2: Schematic illustration of the standard pipeline of the Bag-of-Words model used in many vision tasks.	25
FIGURE 3: The schematic diagram of our proposed image-text alignment algorithm for web images and their associated text terms or phrases.	36
FIGURE 4: The web page segmentation process for image-text pairs generation: (a) visual layout of a web page rendered by IE; (b) the corresponding HTML document where the image node of interest is highlighted with red bounding box; (c) a part of the DOM-Tree where the image node of interest is in red bounding box.	40
FIGURE 5: An example of the generated image-text pairs.	40
FIGURE 6: Some web images and their associated phrases ranked by the pf-iif value.	45
FIGURE 7: Examples of near-duplicate image groups, associated text and top ranked extracted phrase.	48
FIGURE 8: Image clusters and associated phrase lists. The single ranking list is aggregated by the summing strategy.	52
FIGURE 9: Two alternative views of the hyperbolic visualization of the phrase correlation network. Only 120 terms are shown in this picture to avoid visualization clutter.	54
FIGURE 10: Example image clusters of original aggregated phrase lists and the re-ranked lists produced by our relevance re-ranking algorithm.	57
FIGURE 11: Performance comparison of different phrase aggregation strategies.	60
FIGURE 12: Performance comparison of image-text alignment with and without performing random walk (RW) for relevance re-ranking.	61



FIGURE 13: Performance comparison in the application of image indexing and retrieval. Three cases are assessed: (1) image clustering and relevance re-ranking are not adopted, denoted as “without clustering”; (2) image clustering is used but random walk is not adopted, denoted as “without re-ranking” and (3) both image clustering and random walk are used, denoted as “full approach”.	63
FIGURE 14: Performance comparison with Berg’s method and the relevance model in the application of image indexing and retrieval.	64
FIGURE 15: Sample training images collected by our algorithm for several object concepts.	67
FIGURE 16: Image classification accuracy comparison between the methods used to collect training images.	68
FIGURE 17: Schematic illustration of the mean feature representation of an image category. An image is represented as a bag of visual features (middle column); the mean feature vector w.r.t an category is then computed as the visual representation of that category (right column).	76
FIGURE 18: An example for illustrating the process of constructing a visual tree by using the hierarchical $k$ -means based on the mean feature representations. Both of the branch-factor $B$ and the maximum depth $H$ are set to be 3, and each node is labeled with a pair of numbers, $l$ - $j$ , indicating the $j$ th node at depth $l$ .	79
FIGURE 19: Visualization of the visual tree structures. The visual trees were built with branching factor $B = 6$ and maximum depth $H = 4$ . The leaf nodes are not shown since each of them contains a single class. (a) visual tree constructed using hierarchical $k$ -means (c.f. Algorithm 1) ; (b) visual tree constructed using hierarchical spectral clustering (c.f. Algorithm 2).	87
FIGURE 20: Visualization of the label tree structure. The label tree was built using spectral clustering based on the confusion matrix (see [9]) with the branching factor $B = 6$ and the maximum depth $H = 4$ . The leaf nodes are not shown since each of them contains a single category.	89
FIGURE 21: The classification performance of the visual tree model using different number of centroids per class. $B$ : branching factor; $H$ : the maximum depth of a tree.	93

FIGURE 22: The testing speedup performance of the visual tree model using different number of centroids per class. $K$ : the number of centroids per class; $B$ : branching factor; $H$ : the maximum depth of a tree.	94
FIGURE 23: Performance evaluation of the soft prediction scheme. The categorization accuracy is plotted over different average numbers of dot products.	95
FIGURE 24: Inter-related dictionaries for a group of visually correlated categories. A common dictionary $\mathbf{D}_0$ is used to characterize the commonly shared visual patterns, and five category-specific dictionaries $\{\hat{\mathbf{D}}_i\}_{i=1}^5$ are devised to depict the class-specific visual patterns.	99
FIGURE 25: Illustration of the local classification scheme when the labels of test images are defined as the visually similar classes within a single group.	114
FIGURE 26: Illustration of the global classification scheme when the labels of test images are defined as the classes from $T$ different groups.	115
FIGURE 27: Visualization of the visual and label tree structures. The visual and label trees were built by recursively performing spectral clustering on the visual affinity matrix and classification confusion matrix, respectively. The branching factor is 5 and the maximum depth is 4 in both trees. The leaf nodes are not shown since each of them contains only one image category.	117
FIGURE 28: Hyperbolic visualization of the visual tree of depth 2 for the ILSVRC2010 data set. AP clustering was used to partition the image categories (Section 5.2.2).	119
FIGURE 29: Illustration of the hierarchical classification scheme where the group and category classifiers are trained based on different dictionaries ( <i>i.e.</i> , feature spaces).	120
FIGURE 30: The number of groups (bar, units indicated in the left y-axis) and cumulative number of groups (line, units indicated in the right y-axis) of different image category clustering methods on ILSVRC2010 data set.	126
FIGURE 31: Example groups identified by different image category clustering methods on ILSVRC2010 data set. Each cell shows the sample pictures of the categories in the same group.	127

FIGURE 32: Classification accuracy rates using the dictionaries learned by JDL and UDL based on different image category clustering methods for ILSVRC2010 data set.	130
FIGURE 33: The values of the objective function (Eq. 33) in log scale of JDL on four image category groups of different sizes.	133
FIGURE 34: The Fisher scores ( $\text{tr}\left(\frac{S_W}{S_B}\right)$ ) of JDL on four image category groups of different sizes.	133
FIGURE 35: Comparison between JDL and IMDL using different dictionary sizes per category on the Oxford flower data set.	134
FIGURE 36: Dictionary training time of JDL on three image category groups of different sizes.	135
FIGURE 37: The visual concept network for the NUS-WIDE [25] data set.	144
FIGURE 38: The visual network of the 1000 image concepts on ImageNet [33] database.	145
FIGURE 39: Some visually-related image concepts in the NUS-WIDE [25] data set.	146
FIGURE 40: Some visually-related image concepts in ImageNet [33] database.	146
FIGURE 41: The consistency between the numerical values of semantic gaps $\Upsilon(\cdot)$ for learning complexity estimation and the accuracy rates for automatic concept detection in NUS-WIDE image set.	159
FIGURE 42: Comparison between two alternative approaches for supporting quantitative characterization of the semantic gaps ( <i>i.e.</i> , numerical values of the semantic gaps $\Upsilon(\cdot)$ ).	162
FIGURE 43: The consistency between the numerical values (scales) of semantic gaps $\Upsilon(\cdot)$ for learning complexity estimation and the accuracy rates for automatic concept detection in ImageNet data set.	164
FIGURE 44: The consistency on the trend of the semantic gaps under different similarity functions: our algorithm using kernel function <i>versus</i> cumulative codeword histograms.	166

- FIGURE 45: The consistency on the trend of the semantic gaps under different similarity functions: our algorithm using kernel function *versus* Mahalanobis distance. 166
- FIGURE 46: Performance comparison on the accuracy rates for automatic concept detection for NUS-WIDE image set: our structural learning algorithm, traditional structural SVM algorithm, traditional multi-task boosting algorithm. 168
- FIGURE 47: Performance comparison on the accuracy rates for automatic concept detection for ImageNet data set: our structural learning algorithm, traditional structural SVM algorithm, traditional multi-task boosting algorithm. 168
- FIGURE 48: Performance comparison on the accuracy rates for automatic concept detection for ImageNet image set: our structural learning algorithm by using both the scales of the semantic gaps and the visual concept network for inference model selection *versus* traditional structural SVM algorithm by performing structural output regression over the visual concept network. 170
- FIGURE 49: Our experimental results on the correlation among the accuracy rates for concept detection, the scales of semantic gaps ( $\Upsilon(\textit{castle}) > \Upsilon(\textit{flower}) > \Upsilon(\textit{beach})$ ), and the sizes of training image instances for concept classifier training. 171

## LIST OF TABLES

TABLE 1: Performance comparison of the proposed method with and without performing near-duplicate image clustering.	59
TABLE 2: The 80 concepts selected for classification evaluation.	66
TABLE 3: Classification accuracy (%) comparison between our visual trees and the label tree and its variants on the ILSVRC2010 image set. <i>Cues for node splitting</i> : information used to learn the tree structures in different methods. $T_{B,H}$ denotes a tree having at most $B$ children per node and $H$ depths (root node has depth 0).	91
TABLE 4: Computational efficiency comparison of different tree methods. $S_{train}$ : testing speedup compared to the flat model. $S_{test}$ : testing speedup compared to the flat model.	92
TABLE 5: A number of category groups identified by $k$ -means based on the visual representations of the categories in ILSVRC2010 data set.	104
TABLE 6: A number of category groups identified by AP clustering based on the visual similarities between the image categories in ILSVRC2010 data set.	105
TABLE 7: Recognition accuracy on the 17-category Oxford flower data set (continued in Table 8).	122
TABLE 8: Recognition accuracy on the 17-category Oxford flower data set (continued from Table 7).	122
TABLE 9: Performance comparison of the JDL algorithms with and without separating the common visual atoms ( $\mathbf{D}_0$ ) from the category-specific ones ( $\{\hat{\mathbf{D}}_i\}_{i=1}^{17}$ ) for the 17-category Oxford flower data set.	124
TABLE 10: The configurations of different dictionary learning and image category clustering methods. # of Groups: the number of groups; Total # of Words: the total number of visual words used in all dictionaries; Feat. Dim.: the dimension of the feature vector fed to SVM; UDL: unsupervised dictionary learning.	129

TABLE 11: Comparison between using the discriminative dictionaries (common and category-specific dictionaries) learned by JDL and the single dictionary trained by UDL for visual representation in hierarchical image classification. GC: group classifier; CC: category classifier; Feat. Dim.: feature dimension.	131
TABLE 12: Comparison between JDL and a few state-of-the-art methods on the ILSVRC2010 data set.	132
TABLE 13: The 81 image concepts in NUS-WIDE [25] for algorithm evaluation.	141
TABLE 14: A part of the 1000 image concepts in ImageNet [33] for algorithm evaluation.	142
TABLE 15: Image concepts with small semantic gaps in NUS-WIDE [25] data set.	158
TABLE 16: Image concepts with large semantic gaps in NUS-WIDE [25] data set.	158
TABLE 17: The consistency between the numerical values (scales) of semantic gaps $\Upsilon(\cdot)$ for learning complexity estimation and the accuracy rates for automatic concept detection.	160
TABLE 18: Comparison on image concepts with small semantic gaps.	161
TABLE 19: Image concepts with small semantic gaps for ImageNet [33] data set.	163
TABLE 20: Image concepts with large semantic gaps for ImageNet [33] data set.	163
TABLE 21: Comparison on the scales of semantic gaps for different image sets.	165

## CHAPTER 1: INTRODUCTION

### 1.1 Contexts

Our daily activities (*e.g.*, walking, driving, reading, social interaction, *etc.*) depend on our inherent and excellent visual recognition capabilities: we humans can effortlessly detect and categorize visual objects from among tens of thousands of classes [36] within a fraction of a second [143, 144], despite the tremendous variations in appearance of an object. Building computational systems to emulate our own visual recognition abilities is a long-standing goal in artificial intelligence which is dated back to the 1960s. Needless to say, solving this problem will have a huge impact on the human society as it can lead to many revolutionary applications such as intelligent robots, autonomous driving and image semantically understanding, to name a few. A few decades later, enabling computers to understand and interpret images as accurate as humans remains unreachable. Albeit, there are quite a few success and encouraging stories, such as optical character recognition (OCR), face detection in consumer cameras, and pose estimation in Microsoft Kinect.

The difficulty of visual recognition is partly explained by the fact that each object in the world can be captured into an infinite number of different 2-dimensional images since its position, pose, lighting, and background may vary relative to the viewer [120], and the number of different objects in the visual word is enormous. Furthermore, there

exist a huge intrinsic diversity in the appearances of instances within the same class. The visual appearances of individual instances belonging to an object category usually vary greatly due to the changes of viewpoint and lighting, the deformation of non-rigid objects, the variability of poses and *etc.* The visual ambiguity, two semantically different concepts can have a similar appearance, presents other difficulties as well.

In the field of computer vision, visual recognition is often casted as a learning problem which tries to relate the visual contents of images to the previously defined labels. In principle, the labels can correspond to any semantic pattern, *e.g.*, an object(flower), a part of an object(horse leg), a group of objects(person sitting on a bike), an action (person drinking), or even the whole scene (a basketball match). Also, the task of visual recognition varies in the level of detail, ranging from naming the objects present in an image (image categorization/classification), to localizing them with coarse spatial information (object detection), to outlining them out by estimating a pixel-level map of the named foreground objects and the background (object segmentation). In this dissertation, we study the problem of visual recognition in the context of large-scale image categorization. In particular, a number of unique and challenging issues in large-scale visual recognition have been investigated when the number of categories is in the order of thousands and the amount of instances is in the scale of millions.

## 1.2 Challenges

The key to address the problem of visual recognition is to unveil the mapping from image pixels to semantic patterns, *e.g.*, category name, object identity, *etc.* The



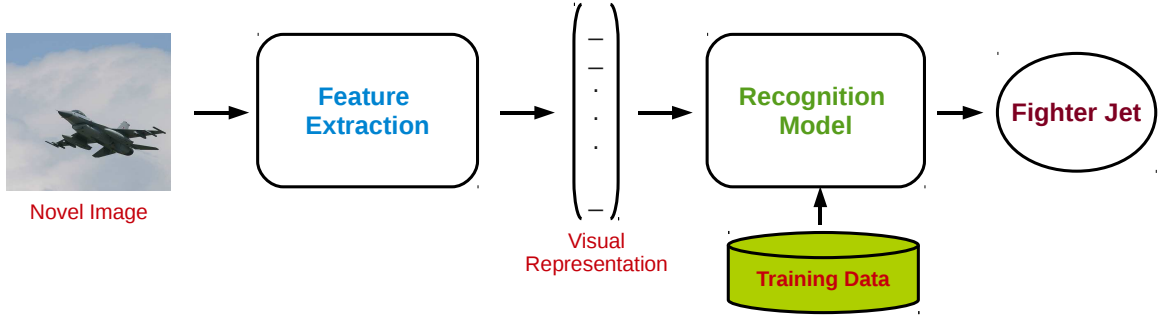


Figure 1: The state-of-the-art image categorization pipeline follows the paradigm of machine learning.

mapping itself is, however, infinite and extremely complex to model, which rules out the straightforward option of designing a set of rules. Instead, the machine learning paradigm currently prevails in state-of-the-art visual recognition methods since it provides a principle way to learn the mapping between low-level image pixels and high-level semantic abstractions from the seen data. Taking image classification as an example, the most advanced approaches which achieve state-of-the-art results are developed in the machine learning framework (See Figure 1). It mainly includes three components: feature extraction, model training and prediction. While effective feature extraction requires significant expertise from computer vision, the success of classifier training and prediction relies greatly on machine learning technical advances.

Visual recognition research along this line has made a great progress in the past decade, and accomplished promising results, especially on the data sets of moderate sizes. For example, to the best of our knowledge, the top results on the Caltech101 image database [51], a pioneer benchmark data set containing 101 categories, have reached up to 82.5% [15] with only 30 images per class being used as training data. The results on the Caltech256 data set[64] which is of similar nature as Caltech101

except that it is substantially larger, have evolved from around 30% [61, 165] to about 50% [15] if 30 images per category are used for training. Higher recognition rates are consistently reported in the literature if more training samples are available. However, these benchmarking data sets are still too small to be used to learn a real-world visual recognition system. The shortage is twofold: the number of categories and the amount of instances per class. Towards making a visual recognition system workable in practice, one has to recognize about 30K visual categories [12], and to cover the tremendous visual diversity of a particular visual category, thousands or even hundreds of thousands images are demanded.

Large-scale visual recognition which tries to recognize thousands of categories and process millions of images, has recently attracted vision researchers great attention [116, 32, 9] because it will pave the way for tons of potential applications. In the meanwhile, however, many unprecedented challenges emerge and need to be addressed as the data volume increases enormously. This dissertation focuses on addressing a number of the challenges, including the collection of training data, the efficiency of classifier learning and prediction, the effectiveness of visual features and the estimation of the learning complexities of visual concept classifiers.

### 1.2.1 Collection of Labeled Image Data

The performance of a visual recognition system hinges on the quality of the labeled training data. For example, the breakthrough in face detection [126, 151] was made after a large number of labeled face images were readily available. In the early vision research, training data was often manually labeled by users. While carefully labeled

data is usually reliable and of high quality, but the labeling process is known to be tedious and labor intensive. In large-scale visual recognition, a large volume of labeled data is desired because: (1) the number of visual categories is very large, from several thousands to tens of thousands; (2) the appearance diversity within a visual category is tremendous which requires a large variety of instances to express. Obviously, it is prohibitively expensive to manually label millions of images from among thousands of visual categories for large-scale visual recognition.

As the digital photo capturing devices and mass storage systems become ubiquitous, the proliferation of user contributed images on the Internet has provided potential ways to effectively collect a massive amount of training data for visual recognition. The Web images are often accompanied by a form of description, ranging from high-level annotations provided by users, to structured surrounding text contributed by webpage creators. The text descriptions associated with those weekly-labeled images contain rich information about the semantic meanings of the images, which can in turn be exploited as a reliable source of training data for many vision tasks, such as image annotation [179] and classification [163].

It is non-trivial to leverage weekly-labeled Web images for visual recognition since the accompanied text (*e.g.*, tags, captions, surrounding text, *etc.*) is only weekly related to the visual contents of the images. The key is to find an effective and affordable way to reliably establish the relations between the Web images and the text descriptions. One way to clean the weekly-labeled images is to resort to human supervision via crowdsourcing [140, 127, 33]. However, it is very difficult to scale crowdsourcing to web-scale image data, and recent interest on attribute-based methods for representa-

tion [77, 50] and fine-grained object recognition [158, 114] make the annotation tasks even more expensive, defeating many advantages of crowdsourcing. The challenge is to find an effective and efficient way to harvest large-scale weekly-labeled images on the Web as training data for vision tasks.

### 1.2.2 Efficiency of Classifier Training and Testing

Statistical learning models (*e.g.*, support vector machine (SVM), logistical regression, *etc.*) are often used in modern visual recognition systems to learn a mapping from image pixels to semantic patterns with the hope to generalize it to unseen data. In large-scale visual recognition, the massive amounts of categories and images poses tremendously computational challenges for traditional machine learning algorithms. First, it is prohibitively expensive, if not impossible, to load the training data all at once into the random access memory (RAM) to fit the designed learning models in the traditional batch-training manner. For example, the visual features extracted from the 1.2M images in the ILSVRC2010 data set typically amounts to 200GB or even more than 1,300GB [90] depending on the types of visual features. Second, the time for classifier training and inference significantly increases if a flat approach (*e.g.*, multi-class SVM with one-vs-rest schema) is simply adopted. To reduce the training time, it is very attractive to exploit the relatedness between different category classifiers to speed up the learning by training them jointly. The inter-category correlations which can be used as surrogates to measure the relatedness are of particular interest in this dissertation. The computational cost of inference in a flat approach grows linearly with the number of image categories, which prohibits its practical use

in many applications with real-time or near real-time requirements.

One way to reduce the computational cost of large-scale visual recognition systems is to hierarchically organize the categories in a tree structure by exploiting their inter-category relations. In [9], Bengio *et al.* proposed a label tree model for supporting efficient large-scale categorization, where each node is associated with a subset of the label set of its parent node and as well as a predictor used to determine the best-matching child node to follow at the next level. Each of the leaf nodes represents a single category. The classification (label prediction) for a test image is performed as traversing from the root to one certain leaf node. This often leads to sub-linear time complexity in terms of the number of categories since only a limited number of possible node predictors are needed to be evaluated. A classification confusion matrix was used in [9] to construct the label tree structure, motivated by the observation that putting the classes which are easily confused by classifiers into the same set (*i.e.*, the same tree node) makes the classifiers associated with that tree node to be easily learnable [9]. To obtain the confusion matrix for learning the label tree structure, one has to train many one-versus-rest (OVR) classifiers in advance. When the OVR classifiers and confusion matrix are reliable, the label tree method tends to assign visually correlated categories into the same node. However, training a large number of OVR classifiers is computationally expensive, and often suffers from the problem of huge sample imbalance. Therefore, the challenge here is to find an efficient and reliable method to hierarchically organize image categories in a tree structure for reducing the computational complexity of a large-scale visual recognition system.

### 1.2.3 Effectiveness of Visual Content Representations

Image content representation plays a critical role in visual recognition as the recognition accuracy depends largely on the discrimination power of visual content representations. It is generally agreed that distinguishing a large number of categories (*e.g.*, several thousands) is more difficult than distinguishing only a few (*e.g.*, less than a hundred). When the number of image categories increases to several thousands, an effective and yet efficient scheme for visual feature extraction is highly demanded to achieve better recognition results.

Among various approaches explored in the literature, the *bag-of-visual-words* (BoW) model has been widely used in many vision tasks, and achieved remarkable results, such as object recognition [62, 160], image classification [85, 79] and segmentation [176]. The idea of the BoW model was borrowed from the information retrieval community, where text documents contain some distribution of words, and thus are often represented by their word counts. This is known as a bag of words model [129] for text documents. An image can be analogically considered as a sort of document containing the local features as visual words. BoW model quantized the high-dimensional, continuous-valued local descriptors into a collection of visual words, called codebook or dictionary. The visual content of an image or object is then simply represented by a dictionary-based histogram. By taking advantage of the invariance properties of the local descriptors, BoW model provides great tolerance to viewpoint or pose variation, which makes it an effective visual representation scheme in many applications. Another particular convenience of the BoW representation is that it translates

a set of high-dimensional local descriptors (the number of the local features varies across different images) into a fixed-length vector representation for all images. This is desirable in many machine learning algorithms that by default assume the input feature space is vectorial.

Unfortunately, there is no off-the-shelf visual dictionary available for general vision tasks. As a result, a visual dictionary has to be learned from data. Learning codebooks with strong discrimination power is of particular interest since it is essentially related to the discrimination of the dictionary-based visual representation. However, the discrimination of a dictionary learned through unsupervised learning is intrinsically limited as it is optimized for reconstruction but not for classification.

Many supervised dictionary learning approaches have been recently proposed to learn more a discriminative *universal dictionary*. Specifically, in [103, 102] the dictionary learning and classifier training are combined in a single objective function. The discrimination of the dictionary to be learned is boosted by solving the unified optimization. However, the optimization is of high complexity, and often approximated by iteratively solving the constitutive sub-problems. In addition, to express the complex visual signal of a large number of image categories, the number of visual words of a universal dictionary should be large enough. However, learning a universal dictionary of a large size (*e.g.*, tens of thousands) is challenging in practice. Besides, many works have advocated learning *multiple category-specific dictionaries* [102, 122] to enhance the discrimination. However, learning category-specific dictionaries independently prevents them from sharing common visual atoms across different classes. What is desired is a new algorithm which can learn more discriminative dictionaries

in an effective but yet computationally affordable way.

#### 1.2.4 Learning Complexity of Visual Concept Classifiers

In large-scale visual recognition, statistical models are used to relate low-level image features to high-level semantic labels. The labels are semantically oriented as the ultimate goal of a visual recognition system is to help humans on managing, organizing, indexing and understanding visual data. It is generally agreed that one of the reasons for the limited success of current visual recognition systems is because there is a difference between low-level image similarity which is computed based on some distance measures of low-level visual features and high-level similarity that is the perceived image similarity by the human users. Two images that are measured as very similar based on some low-level similarity measures can in fact look very different. This inconsistency between low-level image similarity and high-level perceived subjective image similarity is often referred to as the *semantic gap* [138].

To bridge the semantic gap, a myriad of sophisticated machine learning models have been proposed to learn the visual concept classifiers from large amounts of labeled training images (*i.e.*, learning non-linear mapping functions between low-level visual features and the high-level visual concepts). However, it is not a trivial task because the learning complexities of concept classifier could significantly vary across different visual concepts. In other words, some visual concepts (*e.g.*, *grass*, *tree*) may have lower learning complexities since their semantic gaps are smaller; on the other hand, some other concepts (*e.g.*, *town*, *running* and *living room*, *etc.*) may have higher learning complexities with respect to concept classifier training because



their semantic gaps are larger.

Relating the semantic gap to the complexity of learning a visual concept classifier provides a potential way to estimate the learning complexity before the classifier being trained. It would greatly help us to more effectively learn the visual concept classifier. Thus, a quantitative analysis of the semantic gap is beneficial for estimating the complexities of learning different visual concept classifiers, and ultimately helps effectively training the classifiers. However, such an analysis is currently missing in vision research.

### 1.3 Contributions

In this dissertation, we contribute to large-scale visual recognition in many aspects by tackling the challenges elaborated in Section 1.2. The main contributions are structured in *four* chapters. In this section, we briefly describe the main contributions of each chapter.

- Chapter 3 In this chapter, we consider the task of collecting a large number of labeled images from the Web for classifier training. In particular, this is achieved by developing an automatic image-text alignment algorithm to align Web images with their most relevant auxiliary text terms or phrases. First, a large number of cross-media Web pages which contain Web images and their auxiliary texts are crawled and segmented into a set of image-text pairs (*i.e.*, informative Web images and their associated text terms or phrases). Second, near-duplicate image clustering is used to group large-scale Web images into a number of clusters according to their visual similarity, and each of them consists of near-duplicate images. Per-

forming near-duplicate image clustering can significantly reduce the uncertainty on the relatedness between the semantics of Web images and their auxiliary text terms or phrases. Finally, random walk is performed over a phrase correlation network to achieve more precise image-text alignment by refining the relevance scores between Web images and their auxiliary text terms or phrases. The work in this chapter was partly published in [181, 183].

- Chapter 4 In this chapter, we consider reducing the computational complexity of large-scale image categorization by hierarchically organizing the image categories with a tree structure. Specifically, a visual tree is constructed without using a cost-intensive classification confusion matrix which requires training a large number of classifiers in advance. The visual tree model is more computationally efficient to be constructed comparing to other tree models because it is built solely based on the visual correlations between image categories. We empirically show that the performance of the proposed visual tree model is comparable to that of other tree models in terms of recognition accuracy. The work in this chapter was submitted to IEEE Transactions on Image Processing.
- Chapter 5 In this chapter, we consider learning discriminative dictionaries for image content representation. Specifically, we present a joint dictionary learning (JDL) algorithm which exploits the inter-category visual correlations to learn more discriminative dictionaries. Given a group of visually correlated categories, JDL simultaneously learns one common dictionary and multiple category-specific dictionaries to explicitly separate the shared visual atoms from the category-specific ones. The problem of JDL is formulated as a joint optimization with a discrimination

promotion term according to the Fisher discrimination criterion. The visual tree method described in Chapter 4 is used to cluster a large number of categories into a set of disjoint groups, so that each of them contains a reasonable number of visually correlated categories. The process of image category clustering helps JDL to learn better dictionaries for classification by ensuring that the categories in the same group are of strong visual correlations. Also, it makes JDL to be computationally affordable in large-scale applications. We develop three classification schemes to make full use of the dictionaries learned by JDL for visual content representation in the task of image categorization. The work in this chapter was published in [182], and extended in [180].

- Chapter 6 In this chapter, we consider quantitatively characterizing the semantic gap for learning complexity estimation and inference model selection. In particular, a novel data-driven algorithm is developed to quantitatively compute the semantic gaps directly in the visual feature space since the visual feature space is the common space for concept classifier training and automatic concept detection. The main purpose of quantitatively characterizing the semantic gaps is to automatically select more effective inference models for concept classifier training. This is achieved by (1) identifying the image concepts with small semantic gaps (*i.e.*, the isolated image concepts with high inner-concept visual consistency), and training their one-against-rest SVM concept classifiers independently; (2) determining the image concepts with large semantic gaps (*i.e.*, the visually-related image concepts with low inner-concept visual consistency), and training their inter-related concept classifiers jointly; and (3) using more images to train the classifiers for the visual

concepts of large semantic gaps. This work was published in [46].

Before presenting our contributions in Chapter 3, 4, 5 and 6, we first review the mostly related works to ours in Chapter 2. Finally, we conclude this dissertation in Chapter 7.

## CHAPTER 2: RELATED WORK

As discussed in Chapter 1, the work of this dissertation is closely related to the efforts on labeled image data set construction, large-scale classifier training and organizing, image content representation and semantic gap modeling. In this chapter, we briefly review currently prevailing approaches to these problems.

### 2.1 Image Data Set Construction

Reliable and unbiased data sets are indispensable for computer vision research. In fact, data sets have been the main factor for the considerable progress in visual recognition, not just as sources of training data, but also as means of benchmarking different competing algorithms. However, constructing a data set of a large volume of labeled images requires significant efforts, either through human or automatic supervisions.

#### 2.1.1 Manually Labeling

A number of manually labeled data sets have been widely used for evaluating current vision algorithms. Some remarkable examples are Caltech101 [51], Caltech256 [64] and PASCAL VOC [41], among others. However, the number of categories of them is only ranging from dozens (20 in PASCAL VOC) to a few hundreds (256 in Caltech256), which is far away from that is demanded by a large-scale visual recognition system. Unfortunately, labeling millions of images from among thousands

of possibilities is inhabitant.

Recently, crowdsourcing have drawn significant attention from vision researcher with its increasing accessibility. Von Ahn *et al.* [152] proposed collecting labeled images through a computer game. In the game, two users are automatically matched as partners. The paired partners do not know each other’s identity, and they cannot communicate. Their task is to agree on a word that would be a possible label for the image shown to both of them in a limited time. Millions of image have been labeled through this game. However, the labels are usually of limited information with respect to the image contents since users are inclined to use general labels (*e.g.*, dog) other than specific ones (*e.g.*, poodle) to describe the images. Sorokin and Forsyth [140] made the first attempt at labeling image through crowdsourcing at low costs using platforms such as Amazons Mechanical Turk (AMT). Deng *et al.* have created the ImageNet [33] using AMT which have collected 14M labeled images for 22K categories. Although AMT have made the possibility to collect large amounts of annotations in a cost effective manner, the quality of the annotations is hard to verify. Furthermore, creating detailed instructions and user interfaces can take the same order of time as manually labeling the images, defeating the advantages of AMT [105].

### 2.1.2 Automatic Approach

The TinyImage database [146] is the most popular benchmarking data set created via an automatic way. A total of 53,464 nouns were submitted to Google’s Image Search and other engines to collect relevant images for each of them. There are

about 80M images in total with an average of 1K images per noun, among which 10-25% are estimated to be accurate images with respect to each noun. Many other interesting techniques have recently been developed to leverage Internet images or social images, such as user-tagged images from Flickr, for image understanding tasks. Li *et al.* [89] have developed an interesting technique for tag re-ranking by identifying the most relevant social tags for image semantics interpretation. Tang *et al.* [142] have developed a label propagation technique for image and tag cleansing, where both social images and web images are used for algorithm evaluation. Fan *et al.* [49] and Schroff *et al.* [136] have combined text, meta-data and visual information for image and tag cleansing with the aim to harvest large-scale image databases from the Internet. Many researchers have adopted relevance re-ranking tools to improve image/video search by fusing multiple modalities of web images/videos [94, 70, 69, 75, 95, 153, 96, 75], where the goal for relevance re-ranking is well-defined, *e.g.*, identifying the relatedness between the semantics of returned images/videos and the text terms or phrases given by a specific query.

## 2.2 Multiclass Classifier Learning and Organizing

Visual recognition via classification is a typical multiclass classification problem, and consequently demands a multiclass classifier. Regarding the multiclass classifier learning and organizing, there exists two main approaches: *flat* and *hierarchical*.

### 2.2.1 Flat Approach

The flat classification approach treats each category or class separately, thus in effect flattening the class structure. In particular, a myriad of efforts have been

made to enable *support vector machine* (SVM) on multiclass classification, including dividing the problem into a number of binary problems and directly casting it into a single multiclass learning problem.

Suppose there are  $L$  classes of interest, a commonly used division strategy is to learn  $L$  *one-versus-rest* (OVR) SVMs [150] which is also referred to as *one-versus-all* (OVA) SVMs. Another strategy is to build  $\frac{L(L-1)}{2}$  *one-versus-one* (OVO) SVMs [76]. In contrast, the *directed acyclic graph support vector machine* (DAGSVM) proposed in [121] trains  $\frac{L(L-1)}{2}$  binary SVMs as the OVO method dose, but uses a rooted binary DAG containing  $\frac{L(L-1)}{2}$  internal nodes and  $L$  leaves, to make inference. Weston *et al.* have proposed a multiclass SVM in [159] where  $L$  two-class decision functions are constructed, and the decision boundary determined by the  $i$ th function separates the samples of the  $i$ th class from the others. The hyper-parameters are learned by solving a single optimization. In [27], Crammer *et al.* casted the multiclass SVM as a constrained optimization problem with a quadratic objective function.

Training SVMs for large-scale visual recognition with flat approaches mentioned above is challenging because the number of categories could be thousands, and the training samples could be millions (*e.g.*, the ImageNet data set [33] contains 14M images for 22K categories). The computational complexity of the flat approaches grows at least linearly with the number of classes but super-linearly with the number of training samples, given as

$$Q_{train}^{OVR} = L \cdot O(N^c), c > 1, \quad (1)$$

where  $N$  is the number of training samples of all the classes, and  $L$  is the number



of categories. The term  $O(N^c)$  denotes the average training complexity of a binary SVM for a single class. For prediction, we need to feed the novel instances to all the  $L$  classifiers. Therefore, the complexity is

$$Q_{test}^{OVR} = L \cdot O(1), \quad (2)$$

where  $O(1)$  denotes the average complexity for a single SVM to predict a testing sample. Assuming the training samples are uniformly distributed over all the categories, the training and testing complexities of the OVO SVM are given as

$$Q_{train}^{OVO} = \frac{L(L-1)}{2} \cdot O\left(\left(\frac{N}{L}\right)^c\right), c > 1, \quad (3)$$

$$Q_{test}^{OVO} = \frac{L(L-1)}{2} \cdot O(1). \quad (4)$$

The computational complexity of the multiclass SVMs implemented by solving a single optimization is even higher which makes them inferior to the OVR and OVO strategies on large-scale classification in practice. The memory complexity is another critical issue when all the training samples can not be loaded into the memory at once. While bagging [19] is a trade-off between memory complexity and classification performance, more elegant techniques are recently proposed, *e.g.*, stochastic gradient descent [16] and block minimization [169], to seek for better performance.

### 2.2.2 Hierarchical Method

The flat classification methods mentioned above ignore the hierarchical structure of the visual categories which exists naturally in a large number of visual categories [125]. Furthermore, as the number of visual categories increases to thousands, the

computational complexity of the flat approaches is prohibitively high, which forms a tremendous obstacle to adopting it in practice. Recently, computer vision researchers have advocated using a taxonomy to organize the categories hierarchically in a tree structure, aiming to reduce the computational complexity of visual recognition systems. A number of semantic taxonomies (*e.g.*, WordNet [52]) have been used for image classification [45, 43, 32]. It is worth noting that it is more reasonable to use the visual information to learn a category hierarchy because the visual space is the common space for classifier training and image classification [48, 46]. Specifically, Sivic *et al.* [137] automatically discovered a hierarchical structure from a collection of unlabeled images by using a hierarchical latent Dirichlet allocation (hLDA) model. In [6], Bart *et al.* adopted a completely unsupervised Bayesian model to learn a tree structure for organizing large amounts of images. Griffin *et al.* [65] and Marszalek *et al.* [107] constructed visual hierarchies to improve the classification efficiency. Bengio *et al.* [9] proposed a label tree model for the same purpose, and Deng *et al.* [34] further extended it by simultaneously learning the tree structure and the classifiers associated with the tree nodes.

### 2.3 Image Content Representation

In this section, we first briefly review a myriad of visual features which have been widely used in the computer vision community including *global descriptors* and *local descriptors*. Also, we describe the bag-of-visual-words (BOW) model in details. Finally, we survey a number of prevailing dictionary learning methods for learning a visual dictionary which is a key ingredient of the BOW model.

### 2.3.1 Global Descriptors

To construct holistic features to capture the visual cues in an image or a region, the simplest way is to describe the pixel intensities or color values in either an ordered or an orderless way. Given an image or region, we can directly concatenate the pixel intensities into a single feature vector in a specific order, *e.g.*, from top to bottom and left to right. The feature vector is often optionally processed by subspace learning methods, such as Principle Component Analysis (PCA) [115] and Fisher Linear Discriminative Analysis (LDA) [38], to seek for a more compact representation. A limitation of the ordered intensity concatenation method is that it often assumes the images or regions of interest are well aligned, otherwise a trivial pixel position shift would result in very different representations.

We can alternatively construct a simple global description of the image pixels with the distribution of its color values or intensities. This is the so-called color histogram. We can use any color space for partition, for example, RGB, Lab, *etc.* Being orderless, a color histogram gives some tolerance for positional shift and partial occlusions. However, as the colors within a certain category usually greatly varies, the effectiveness of color histogram is intrinsically limited. The initial effort to adopting color histogram for object recognition was proposed by Swain and Ballard [141], and a recent extension along this line can be found at [149].

Aside from raw pixels, contrast-based descriptors are generally used to describe the visual content of images partly due to their invariance to illumination changes and color variations. The gradients of image pixel intensities which capture edges and

texture patterns are of particular interest. Two remarkable examples of making use of gradients to form a holistic image representation are the *Histogram of Oriented Gradient* (HOG) [29] and GIST [112]. Beyond the appearance-based features above, the shape feature is another widely used global descriptor which describes objects' outer boundaries and as well as interior contours. A formal discussion on the shape representations and shape matching problem is beyond the scope of this dissertation, and the interested readers can refer to [173] for details.

### 2.3.2 Local Descriptors

While global descriptors provide holistic representations of an image or object, various local descriptors have been invented to obtain repeatable and distinctive image patches for more effective visual representation. Local invariant descriptors are initially proposed for specific object recognition, and have been successfully extended to generic object detection and recognition with promising performance (*e.g.*, in [62, 79, 165, 166]).

To extract local descriptor, we typically go through two main steps: (1) key or interest points detection; and (2) descriptors extraction from the neighbor pixels of the detected key points. A variety of interest point detectors have been proposed in the past decades to seek for specific invariant properties, such as scale and affine invariance. Specifically, the *Hessian detector* [8] computes the matrix of second derivatives (the Hessian), and searches for locations that exhibit strong derivatives in two orthogonal directions determined by the determinant of the Hessian. Another popular corner-like feature detector is the *Harris detector* [58] which finds the locations whose

second-moment matrix has the two largest eigenvalues. Although Harris and Hessian detectors are particularly robust to image plane rotations, illumination changes and noise [133], they are vulnerable to large photometric and geometric variations. To detect scale invariant features, automatic scale selection has been proposed, and implemented using either the *Laplacian-of-Gaussian* (LoG) detector or the *Difference-of-Gaussian* (DoG) detector. LoG [92, 91] detect blob-like features, and automatically determine the scale by searching the extrema in the scale space produced by a series of scale-normalized Laplacian of Gaussians. While LoG achieves very robust detection result, the computational cost of the Laplacian is high. Lowe in [97] proposed using the DoG to approximate the LoG, and DoG is computed using the difference of two Gaussians of adjacent scales. In practice, DoG usually obtains very similar results as LoG dose. For more details of local invariant detectors, we refer to the comprehensive survey paper [148].

The scale-invariant detectors mentioned above are particularly effective for specific object recognition by detecting only a sparse set of key points due to their repeatability and distinctiveness. However, for generic visual recognition, such a sparse set of local features is often insufficient. Instead, many works [165, 166, 17] have empirically shown that a regularly *dense* sampling strategy results in better recognition performance since it ensures that the objects of interest have better coverage. In practice, dense sampling commonly utilizes a regular grid of multiple scales to produce a set of image patches.

Once a number of interest points have been detected from an image, a collection of corresponding descriptors are extracted to encode their content. The most popu-

lar local descriptors are the *Scale Invariant Feature Transform* (SIFT) and *Speed-up Robust Feature* (SURF) descriptors. The SIFT descriptor [97, 98] encodes the image gradient distribution in a localized set of gradient orientation histograms to achieve robustness to lighting changes and small positional shifts. The descriptor computation is performed on the Gaussian image with the closest scale to that of the detected key point. A regular grid of  $16 \times 16$  pixels centered on the key point is sampled as the interest region, and is further divided into sixteen  $4 \times 4$  grids. For each grid, a gradient orientation histogram of 8 orientation bins is computed, and weighted by the corresponding pixel’s gradient magnitude. The sixteen 8-bin gradient orientation histograms are concatenated to form a 128-dimension descriptor vector. The SURF descriptor [7] is an efficient alternative to SIFT. It adopts simple 2D box filters which can be efficiently evaluated using integral images [28, 151] instead of Gaussian derivatives for interest point detection. The SURF descriptor divides the feature region into  $4 \times 4$  grids which is similar to that of SIFT descriptor. However, instead of constructing a gradient orientation histogram for each cell, SURF only summarizes the statistics of the filter responses to form the feature vector.

### 2.3.3 Bag-of-Visual-Words

The Bag-of-Visual-Words (BoW) model, a.k.a. Bag-of-Features (BoF), is in some sense a hybrid (global pooling plus local patches) of the global and local representation styles [63]. It encodes the occurrence of the local feature descriptors within a region of interest. Considering an image or object as a document, BoW simply compute a dictionary-based histogram as the image’s signature. It consists of three major

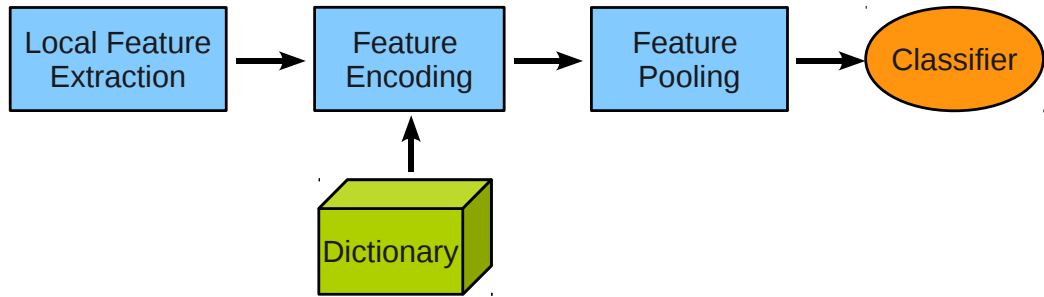


Figure 2: Schematic illustration of the standard pipeline of the Bag-of-Words model used in many vision tasks.

steps as illustrated in Fig. 2. The first step involves the local feature detection and extraction where any detector and extractor mentioned in Section 2.3.2 are applicable. Second, each local descriptor is encoded using one or multiple visual words from a visual dictionary. Third, the occurrence of the visual words are pooled to form a image-level signature of fixed dimensionality.

After a set of local features being extracted from an image or region, each of them is quantized to the space partitioned by the visual dictionary. This process is often named as *feature encoding* or *visual word assignment*. A myriad of encoding methods have been proposed in the literature which can be categorized into two types: hard assignment and soft assignment. Hard assignment encodes a local feature only using the closest visual word to the feature based on some distance metric. On the other hand, multiple visual atoms are used to represent the local feature in soft assignment. Encoding algorithms along this line include *kernel codebook encoding* [61, 119], *Fisher encoding* [117], *super vector encoding* [184], *locality-constrained linear encoding (LLC)* [155] and sparse coding [113] (a.k.a. Lasso [145] or basis pursuit [23]). For the details and comparison of feature encoding methods, we refer to an comprehensive comparison in [22].

The final step which aggregates the codes of local features to form the vectorial representation for an image or region is often referred to as *feature pooling*. Let the encoding response of local features  $\{\mathbf{x}_i\}_{i=1}^M$  be  $\{\mathbf{a}_i\}_{i=1}^M$ , where  $\mathbf{a}_i \in \mathbb{R}^K$ . Feature pooling essentially maps the feature codes to some statistic that summarizes the joint distribution of the codes over a region of interest. There are two main pooling types: average pooling [81, 82] and max pooling [123]. Let  $\mathbf{v}$  be the statistic representation, and  $f$  be the pooling operator. The average pooling is essentially equivalent to the histogram pooling, mathematically given as

$$\mathbf{v} = f(\{\mathbf{a}_i\}_{i=1}^M) = \frac{1}{M} \sum_{m=1}^M \mathbf{a}_m. \quad (5)$$

The max pooling inspired by the mechanism of the primary visual cortex area V1 [120] which computes  $\mathbf{v}$  using a max operator, given as

$$\mathbf{v}[j] = \max\{|\mathbf{a}_1[j]|, \dots, |\mathbf{a}_M[j]|\}, \quad (6)$$

where  $\mathbf{v}[j]$  is the  $j$ -th component of  $\mathbf{v}$ . A theoretical analysis of the two pooling methods in visual recognition can be found at [18], and a thorough comparison has been done in [17]. The max pooling generally results in better performance in visual recognition as it is more robust to noise than the average pooling. More recently, a so-called  $l_p$ -norm pooling is proposed in [53] which is tailored for the class-specific feature spatial distribution.

The BoW representation is completely orderless which means great flexibility allowed with respect to viewpoint and pose changes. However, it also loses the spatial layout of the visual words which is important for visual recognition. To incorporate



the positional information into the BoW model, Lazebnik *et al.* [79] proposed using a spatial pyramid to partition the image plane, and then computes and concatenates the BoW features in the bins of the pyramid. Spatial pyramid representation is sensitive to translations, making it most appropriate for scene-level recognition or images containing only primary objects with some regular backgrounds.

### 2.3.4 Visual Dictionary Learning

A key ingredient of the BoW model, the visual dictionary, has to be learned from data since there are no off-the-rack ones for general vision tasks. Current prevailing approaches to dictionary learning can be divided into two main groups, *unsupervised* and *supervised* dictionary learning.

Unsupervised dictionary learning algorithms usually train a single dictionary through minimizing the residual errors to reconstruct the original signals. In particular, Aharon *et al.* [1] have generalized the  $k$ -means clustering method and proposed the  $K$ -SVD algorithm to learn an over-complete dictionary from image patches. Lee *et al.* [83] treated the problem of dictionary learning as a least squares problem, and solved it efficiently by using its Lagrange dual. Wright *et al.* [161] used the entire set of training samples as the dictionary for face recognition and achieved very competitive results. In [165], Yang *et al.* proposed the ScSPM model which combined sparse coding and spatial pyramid matching [79] for image classification, and the dictionary was trained using the same method as in [83]. The dictionaries learned via unsupervised learning are often lack of discrimination because they are optimal for reconstruction but not for classification.

Most existing supervised dictionary learning methods can be roughly categorized into three main categories in terms of the structure of the dictionaries. In [103, 174, 167, 166], one *single dictionary* is learned for all the classes. To enhance the discrimination of the dictionary, the processes of dictionary learning and classifier training are unified in a single objective function. Many other works have advocated learning multiple *category-based dictionaries*, and tried to enhance their discrimination by either incorporating reconstruction errors with the soft-max cost function [102, 104] or promoting the incoherence of different class dictionaries [122]. Although the sparse coefficients embody richer discriminative information than the reconstruction errors, the classification decision in [102, 104, 122] still solely relies on the residual errors. More recently, a *structured dictionary*, whose visual atoms have explicit correspondence to the class labels, was proposed in [73, 168]. Specifically, Jiang *et al.* [73] integrated the label consistent constraint, the reconstruction error and the classification error into one single objective function to learn a structured dictionary. A  $K$ -SVD-like algorithm was used to solve the optimization. Yang *et al.* [168] also adopted the Fisher discrimination criterion and proposed the Fisher discrimination dictionary learning (FDDL) algorithm to train a structured dictionary.

## 2.4 Semantic Gap Modeling

The semantic gap between the low-level visual features and the high-level semantic concepts has become fundamental barrier in developing learning method for visual recognition. The semantic gaps are actually not uniformly distributed across different visual concepts. That is, the semantic gaps may significantly vary across the image

concepts. In the last decade, many machine learning approaches have been developed to bridge the semantic gap by training more reliable concept classifiers (*i.e.*, the mapping functions from low-level visual features to high-level semantic concepts) [37, 66, 101, 178, 44]. However, no previous works focus on quantitative characterization of the semantic gap directly in the visual feature space.

There are some existing researches on leveraging various information sources to bridge the semantic gap [37, 66, 101, 178, 44, 40, 154, 139, 131, 124, 109]. Specifically, Enser *et al.* [40] have provided a comprehensive survey of the semantic gap in image retrieval. Zhao *et al.* [178] have integrated *latent semantic indexing* (LSI) to negotiate the semantic gap in multimedia web document retrieval. Hare *et al.* [66] have developed both bottom-up and top-down approaches to bridge the semantic gap for the purpose of multimedia information retrieval. Ma *et al.* [101] have developed a two-level data fusion framework to bridge the semantic gap between the visual content of social images and their tags. Wang *et al.* [154] have developed an effective distance metric learning approach to reduce the semantic gap in web image retrieval and annotation. Fan *et al.* [44] have developed a hierarchical approach to bridge the semantic gap more effectively by partitioning the large semantic gaps into four small and reachable gaps. Snoek *et al.* [139] have presented a semantic pathfinder architecture to bridge the semantic gap for generic indexing of multimedia archives. Santini *et al.* [131] have integrated human-system interactions to bridge the semantic gaps, and deal with emergent semantics interactively. Rasiwasia *et al.* [124] have combined query-by-visual-example with semantic retrieval to bridge the semantic gap. Natsev *et al.* [109] have constructed a model vector to bridge the semantic gap by supporting

compact semantic representation of the visual content of the images.

Recently, Lu *et al.* [100, 99] have developed an interesting approach for determining the high-level image concepts with small semantic gaps. To the best of our knowledge, it is a pioneer attempt for determining the high-level visual concepts with small semantic gaps by assessing the consistency between the visual similarity and the semantic similarity. However, good consistency between the two may not always indicate that the corresponding concepts have small semantic gaps, and many auxiliary text terms for the images are weakly-related or even irrelevant with their semantics because of huge tag uncertainty (spam tags, ambiguous tags, loose tags, abstract tags, *etc.* [49]). Hauptmann *et al.* [68] also pointed out what kind of high-level video concepts are most important for supporting semantic video retrieval, and they have also examined how many high-level video concepts are needed and what kind of high-level video concepts should be selected for supporting semantic video retrieval [67]. Deselaers *et al.* [35] have done a pioneering work on evaluating the relationship between the semantic similarity among the labels and the visual similarity among the relevant images in ImageNet [33] image set.

As the inner-concept visual diversity may change with the depth in a concept hierarchy, concept ontology may provide a good environment for identifying the image concepts with smaller semantic gaps. For example, the image concepts at the leaf nodes may have smaller semantic gaps because they may have strong limitation on their semantic senses and their relevant images may have good inner-concept visual consistency. Some pioneering work have been done recently by incorporating the concept ontology for organizing large-scale image/video collections according to their

inter-concept semantic contexts [108, 33, 52]. Schreiber *et al.* [134] and Fan *et al.* [44, 43] have integrated the concept ontology for achieving hierarchical image annotation.

It is worth noting that having good inner-concept visual consistency is just one criterion for semantic gap modeling, and there is another important criterion for supporting quantitative characterization of the semantic gaps: the visually-related image concepts. Their relevant images often share some common or similar visual properties, may have large semantic gaps because they may not be visually separable, and their concept classifiers may have significant overlapping in the visual feature space. For examples, even the relevant images for the object classes “sea water” and “blue sky” may have good inner-concept visual consistency, the object class “sea water” may be detected as “blue sky” because they share some common or similar visual properties. Based on these observations, both the inner-concept visual consistency (*i.e.*, inner-concept visual homogeneity scores) and the inter-concept visual correlations (*i.e.*, inter-concept discrimination complexity scores) should simultaneously be considered for supporting quantitative characterization of the semantic gaps directly in the visual feature space.

Recently, some remarkable works have been done by using Flickr distance [162] and KL divergence [43] to measure the inter-concept visual correlations directly in the visual feature space. However, the image distributions are very sparse and heterogeneous in the high-dimensional visual feature space. Thus the KL divergence between the sparse image distributions cannot characterize their inter-concept visual similarity contexts accurately. To avoid this problem, visual clustering and latent semantic analysis have been used to generate visual ontology for automatic object

categorization [137, 107, 106, 65, 6, 2]. More recently, both the visual similarity contexts and the semantic similarity contexts are integrated for concept ontology construction in [87, 43].

Multi-task learning and structural learning are two potential solutions for addressing the issue of huge inter-concept visual similarity by modeling the inter-concept correlations explicitly and training multiple inter-related classifiers jointly [43, 147, 42, 14]. One open problem for multi-task learning and structural learning is that they have not provided good solutions for determining the inter-related learning tasks directly in the visual feature space [44]. Torralba *et al.* [147] proposed a multi-task boosting algorithm by leveraging the inter-task correlations for concept detection, where the inter-task correlations are simply characterized by various combinations of the image concepts. Simply using concept combinations for inter-task relatedness modeling may seriously suffer from the problem of huge computational complexity: there are  $2^n$  potential combinations for  $n$  image concepts. In addition, not all the image concepts are visually-related and combining the visually-irrelevant image concepts for joint classifier training may decrease the performance rather than improvement [43].

## CHAPTER 3: AUTOMATIC IMAGE-TEXT ALIGNMENT

In this chapter, we present our work on harvesting large-scale training images from the Web which aims to collect training data for large-scale visual recognition. Specifically, we developed an automatic image-text alignment algorithm to prepare large-scale labeled training images by aligning web images with their most relevant auxiliary text terms or phrases.

### 3.1 Introduction

As digital images are growing exponentially on the Internet, there is an urgent need to develop new algorithms for achieving more effective web image indexing and retrieval by automatically aligning web images with their most relevant auxiliary text terms or phrases, and such auxiliary text terms or phrases can be extracted from the associated web documents [138, 132, 10, 118]. Many potential applications could be benefited from achieving more accurate alignments between the semantics of web images and their auxiliary text terms or phrases:

(a) Web Image Indexing and Retrieval: Google image search engine has achieved big success on supporting keyword-based web image retrieval by simply using all the auxiliary text terms or phrases to index web images loosely. For each web image, many of its auxiliary text terms or phrases are weakly-related or even irrelevant with its semantics because each cross-media web page may consist of a rich vocabulary of text

terms or phrases for web content description and only a small portion of the auxiliary text terms or phrases are used to describe the semantics of web images. When all these auxiliary text terms or phrases are loosely used for web image indexing, Google Images search engine may seriously suffer from low precision rates and result in large amounts of junk images [56, 88, 20, 156, 59]. To achieve more effective web image indexing and retrieval, it is very attractive to develop new algorithms for supporting more precise alignments between the semantics of web images and their auxiliary text terms or phrases.

(b) Generating Large-Scale Labeled Images for Classifier Training: Automatic image annotation via classification plays an important role in supporting keyword-based image retrieval [138, 86, 5, 44, 175, 85, 21], where machine learning techniques are usually involved to learn the classifiers from labeled training images. The number of such labeled training images must be large due to: (1) the number of object classes and scenes of interest could be very large; (2) the learning complexity for some object classes and scenes could be very high because of huge inner-category visual diversity; and (3) a small number of labeled training images are insufficient to learn reliable classifiers with good generalization ability on unseen test images. However, hiring professionals to label large amounts of training images manually is cost-sensitive and poses a key limitation on practical use of some advanced machine learning techniques for image annotation applications. On the other hand, large-scale web images and their auxiliary text documents are available on the Internet. As web images and their auxiliary text documents co-occur naturally on the cross-media web pages, the auxiliary text documents may contain the most relevant text terms or phrases for



describing the rich semantics of web images effectively. Thus the cross-media web pages have provided a good source to generate large-scale labeled images for classifier training.

For each cross-media web page, it consists of two key components: web images and auxiliary texts. The auxiliary texts may contain a rich vocabulary of text terms or phrases: some of them are used for describing the semantics of web images but most of them are used for interpreting other web content. Thus we cannot simply use all the auxiliary text terms or phrases for web image indexing because most of them are weakly-related or even irrelevant with the semantics of web images. The highly uncertain relatedness between the semantics of web images and their auxiliary text terms or phrases prevent them from being directly used as a reliable source for web image indexing and classifier training. To collect more labeled training images from the Internet and achieve more accurate web image indexing, it is very attractive to develop new algorithms for achieving more accurate alignments between the semantics of web images and their auxiliary text terms or phrases.

Based on these observations, an automatic image-text alignment algorithm is developed in this paper for achieving more precise alignments between the semantics of web images and their auxiliary text terms or phrases. As illustrated in Fig. 3, our automatic image-text alignment algorithm contains the following components: 1) *web page crawling and segmentation*, which crawls large numbers of cross-media web pages and further partition them into large amounts of image-text pairs; 2) *phrase extraction*, which chunks the phrases from the associated texts and computes the pf-iif (phrase frequency-inverse image frequency) value for each phrase to initialize its

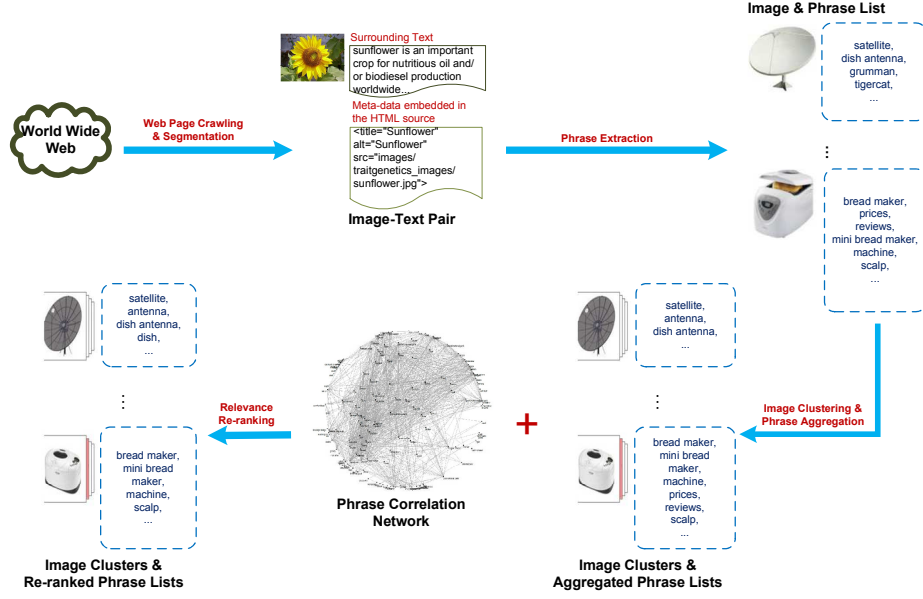


Figure 3: The schematic diagram of our proposed image-text alignment algorithm for web images and their associated text terms or phrases.

relevance score with a given web image; 3) *near-duplicate image clustering and phrase aggregation*, which groups near-duplicate web images together and aggregates multiple individual phrase lists to form a single phrase ranking list to achieve more precise interpretation of the semantics for the near-duplicate web images in the same cluster; 4) *relevance re-ranking*, which re-ranks the relevance scores between the semantics of web images and their auxiliary text terms or phrases by performing random walk over a phrase correlation network.

Our experiments are conducted on a large database of image-text pairs which are generated from large-scale cross-media web pages. Our experimental results have demonstrated that our proposed unsupervised algorithm is superior to some supervised methods on automatic image-text alignment. The rest of this paper is organized as follows. Section 3.2 describes how to generate a large database of image-text pairs; In Section 3.3, we presents our automatic image-text alignment algorithm by perform-

ing near-duplicate image clustering and relevance re-ranking via random walk over a phrase correlation network; Section 3.4 describes our experiments for algorithm evaluation; We conclude in Section 3.5.

### 3.2 Automatic Generation of Large-Scale Image-Text Pairs

In this section, we describe the details for generating a large database of image-text pairs by performing automatic web page partition. A simple strategy is used for informative image extraction, a practical technique is developed for text phrase extraction, and a pf-iif (phrase frequency-inverse image frequency) value is defined for initializing the relevance score between each web image and its auxiliary text term or phrase.

#### 3.2.1 Web Page Crawling and Segmentation

The most recent work on learning image semantics from the associated texts concentrate on either a specific domain, like BBC News [55] or a small data set [84]. In order to seek more insights of this problem, we attempt to collect large numbers of diverse web pages from unrestricted domains. To ensure that each web page (that is being crawled) hosts semantically meaningful images, the image search engines are used to collect the URLs (Uniform Resource Locator) of web pages that are being downloaded. Specifically, we submit 1,000 semantically meaningful query phrases (single or multiple terms) to Google image search engine and extract the hosted URLs of the top 500 returned results. In total, 500,000 URLs (some of them are duplicated) are collected and used to crawl the corresponding web pages.

Modern web pages contain rich cross-media, where the informative web content is

often surrounded by a bouquet of auxiliary web content, such as navigation menus, user comments, texts and images for advertisements, snippet previews of the related documents. Thus it is not feasible to learn the semantics of a web image by using the whole textual information on the associated web page because only a portion of its text terms or phrases is relevant to the semantics of web image. To narrow down the searching scope, one could identify the most relevant surrounding text terms or phrases by using web page segmentation techniques. The underlying reason is that most web page creators are likely to select the most relevant images to illustrate the topics which are discussed in the surrounding text paragraphs and they may place different pieces of web page (text blocks and their relevant images) to make a consistent appearance of these cross-media pieces according to their coherent correlations.

Most existing methods for web page partition [4, 170] can be roughly grouped into three categories: rule-based, DOM-based (Document Object Model), and visual-based approach. The rule-based methods focus on designing the wrappers which might have good performance on one particular type of web pages but fail when the web pages are constructed by using different templates. The visual-based approach [170] treats each web page as a content-rich image and borrows the idea of image segmentation, which partitions the web pages into a set of blocks according to their visual layouts rendered by IE (Internet Explorer). While the segmentation results are promising, applying such visual-based approach on larger-scale collections of web pages encounter many practical problems, such as computational cost and stability.

In this work, we adopt the DOM-based method to extract the most relevant text blocks for each web image because it can achieve a good trade-off between the com-

putational cost and the accuracy rate (Fig. 4). Given a web page, a DOM-tree is constructed for organizing its HTML document in a tree structure. The nodes on such DOM-tree contain element nodes (HTML elements), text nodes (the text in the HTML elements), attribute nodes (HTML attributes) and comment nodes (comments). For a particular image node on the DOM-tree (Fig. 4 (c)), the region growing algorithm is then employed to extract its most relevant text block(s), where the corresponding image node on the DOM-tree is set as the start point, and a upward growing search is performed until it reaches any text node. The inner texts embedded in the text node(s), which have been touched by the region growing search, are extracted as the text block(s) for this particular web image. In addition to the text blocks, we also extract meta-data embedded in the HTML tags as side information, which strongly reflects the semantics of a web image. Four types of meta-data, including alternate texts, image titles, image file names, and web page titles, are extracted for generating more meaningful image-text pairs.

### 3.2.2 Informative Image Extraction

In a typical cross-media web page, informative images often co-occur with uninformative web images, like navigation banners, image icons for advertisements, button icons, etc. Automatically isolating the informative blocks from the uninformative ones in the web pages has attracted many attentions from a lot of researchers. For instance, Debnath *et al.* [31] represented web page blocks by using some desired features and trained a classifier to distinguish the informative and uninformative blocks. A similar idea could be used to isolate the informative images and the uninformative images

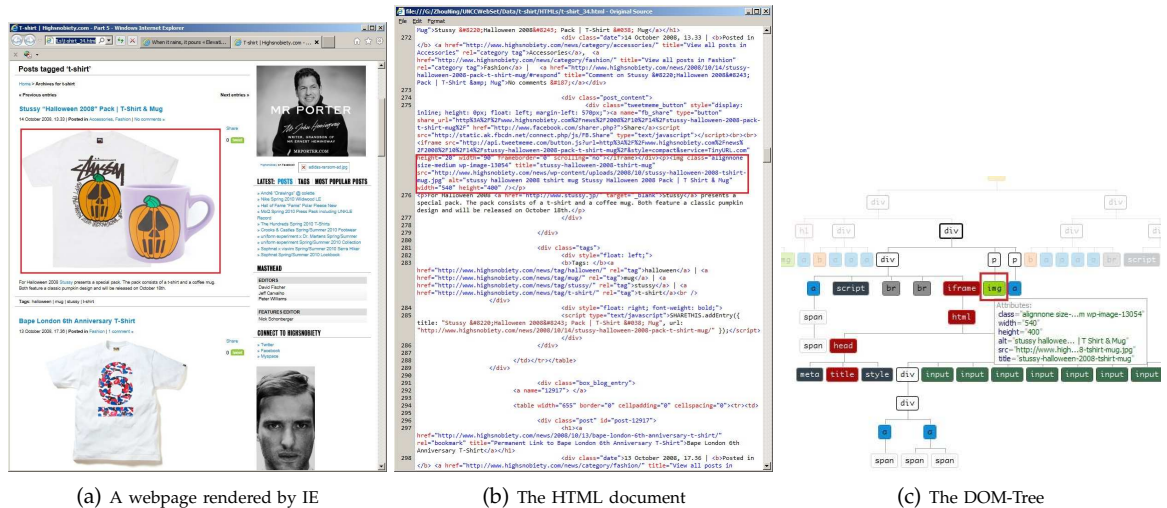


Figure 4: The web page segmentation process for image-text pairs generation: (a) visual layout of a web page rendered by IE; (b) the corresponding HTML document where the image node of interest is highlighted with red bounding box; (c) a part of the DOM-Tree where the image node of interest is in red bounding box.

```

<-Image Url="http://www.wayfaring.info/images/karnak-temple.jpg">
  <Local>./Taj+Mahal/Images/Taj+Mahal_21_6.jpg</Local>
  <Width>460</Width>
  <Height>320</Height>
  <Alt>The Astonishing Temple of Karnak in Luxor, spiritual center of the Ancient Egyptians</Alt>
  <Title>
  <Name>karnak-temple</Name>
  <Text>Temple of Karnak in Luxor What a breathtaking place it is... This vast temple complex is dedicated
    to god Amon and was spiritual center of the Ancient Egyptians. It now amaze us with it's really
    impressive architectural achievements and the atmosphere it stills holds.</Text>
</Image>
<-Image Url="http://www.wayfaring.info/images/Ngorongoro_map_crater.jpg">
  <Local>./Taj+Mahal/Images/Taj+Mahal_21_7.jpg</Local>
  <Width>546</Width>
  <Height>354</Height>
  <Alt>The natural amphitheatre at Ngorongoro Crater</Alt>
  <Title>
  <Name>Ngorongoro_map_crater</Name>
  <Text>The Ngorongoro Crater is a natural amphitheatre created about 2 million years ago when the cone
    of a volcano collapsed into itself, leaving a 100 square mile caldron-like cavity. This caldera,
    protected by a circular unbroken 2,000-foot high rim (610-metres), contains everything necessary
    for Africa's wildlife to exist and thrive. </Text>
</Image>
<-Image Url="http://www.wayfaring.info/images/bali_DewiSri.jpg">
  <Local>./Taj+Mahal/Images/Taj+Mahal_21_8.jpg</Local>
  <Width>200</Width>
  <Height>397</Height>
  <Alt>Explore Indonesia's island Bali travel wonders</Alt>
  <Title>
  <Name>bali_DewiSri</Name>
  <Text>The island of Bali in Indonesia is a traveler's dream-come-true – and offers visitors an exciting
    variety of things to do and see. Bali's natural attractions include miles of sandy beaches (many are
    well-known amongst surfers), picturesque rice terraces , towering active volcanoes over 3,000
    meters (10,000 ft.) high, fast flowing rivers, deep ravines, pristine crater lakes, sacred caves, and
    lush tropical forests full of exotic wildlife. </Text>
</Image>

```

Figure 5: An example of the generated image-text pairs.

from the web pages. However, the computational cost of this algorithm is high, and its robustness is unclear, especially when we deal with large-scale cross-media web pages.

In this work, we adopt a simple method to roughly filter out the uninformative web images. First, we extract web images by looking at the “IMG” tags in a web page’s html document. The size and aspect ratio are then calculated to discard the uninformative web images according to some pre-defined thresholds. Given a web image  $I$ , its aspect ratio  $\rho(I)$  and size  $\sigma(I)$  are defined as

$$\rho(I) = \frac{I_{height}}{I_{width}}, \quad \sigma(I) = \max(I_{height}, I_{width}) \quad (7)$$

where  $I_{height}$  and  $I_{width}$  are the height and width of web image  $I$ . In our current implementations, the web image  $I$ , which matches one of the following conditions, is discarded automatically: (a) its aspect ratio  $\rho(I)$  is lower than 0.2 ( $\rho(I) < 0.2$ ); or higher than 5 ( $\rho(I) > 5$ ); (b) its size  $\sigma(I)$  is smaller than 60 ( $\sigma(I) < 60$ ), as it is often a noise image, like advertisement images and navigation banners, according to our observation. By filtering out the uninformative web images, we have archived around 5,000,000 web images from 500,000 web pages.

For each web image, a piece of text blocks is produced by using the web page segmentation method as described in the previous section which we refer as a image-text pair. We present some example image-text pairs in Fig. 5. Our basic assumption for generating such image-text pairs, *i.e.*, automatically aligning web images with their surrounding text blocks, is that most cross-media web page creators are likely to select the most relevant images to illustrate the topics that are discussed in the

surrounding text paragraphs and they may carefully place different pieces of the cross-media web pages to make a coherent appearance of these cross-media pieces according to their correlations. Such basic assumption may fail in two extreme cases: (a) spam images; (b) advertisement images.

For spam images, the relatedness between their semantics and their surrounding text paragraphs (blocks) is very weak and such spam images can be filtered out effectively by performing near-duplicate image clustering according to their visual similarity (See Section 3.3.1). For advertisement images, even they may interleave with carefully-written web content (*i.e.*, there is a good relatedness between the semantics of advertisement images and their surrounding text documents), such advertisement images can be filtered out effectively by our size and aspect ratio filters. Thus it is reasonable for us to align each informative web image with its surrounding text blocks.

### 3.2.3 Text Phrase Chunking and Ranking

After generating a database with five millions of image-text pairs, we attempt to learn the semantics of web images from their associated texts. For a given web image, we tackle the problem of text phrase chunking as extracting the words from its associated texts and ranking them according to their relatedness with the given web image. Extracting the text terms or phrases from the surrounding texts for image annotation has recently been investigated by researchers from both natural language processing and computer vision [55, 84, 135]. Particularly, Schroff *et al.* [135] applied stemming on the text pieces from various sources, like HTML-tags and web page



texts, to form the textual feature vector for image ranking. Feng *et al.* [55] extracted keywords from the captions of news images as the labels for image annotation. While stemming is used in these works, we believe that more rich semantics are conveyed in the forms of words and collocations, which are named as phrases in the remaining of this paper. In order to extract the phrases from the texts, a chunker provided by Antelope<sup>1</sup>, which is a well-known tool for natural language processing, is used to extract the phrases from the associated texts. The phrases are restricted to be  $n$ -grams (  $n \leq 4$  ). That is, any parsed phrase, whose constitutive items are more than four, is discarded. A standard list of English stop words is also used to remove high-frequency words, such as “the”, “to” and “also”. In addition, we only consider the first 300 tokens if the surrounding texts (the texts embedded in the tag  $\langle text \rangle$  and  $\langle /text \rangle$  in Fig. 5) have too many tokens.

For each web image, the rank scores for its chunked phrases are initialized according to their relatedness with the web image according to their statistical occurrence. Given a phrase  $w_i$ , its frequency  $\text{pf}(w_i)$ , is calculated as a weighted sum of the number of its occurrences in the text and meta-data sources, and it is defined as

$$\text{pf}(w_i) = \text{pf}_{\text{alt}}(w_i) \times 0.5 + \text{pf}_{\text{title}}(w_i) \times 0.3 + \text{pf}_{\text{name}}(w_i) \times 0.15 + \text{pf}_{\text{text}}(w_i) \times 0.05, \quad (8)$$

where  $\text{pf}_{\text{alt}}(w_i)$ ,  $\text{pf}_{\text{title}}(w_i)$ ,  $\text{pf}_{\text{name}}(w_i)$ ,  $\text{pf}_{\text{text}}(w_i)$  are the occurrences of  $w_i$  in alternate text, title, name and surrounding text, respectively. The weights are determined according to the observation that the phrases which are parsed from the alternate texts are more related to the semantics of image content than those phrases extracted

---

<sup>1</sup><http://www.proxem.com/Default.aspx?tabid=119>

from its title, name and surrounding texts. Finally, for each web image, the extracted phrases are ranked by using their pf-iif values which are calculated as

$$\text{pf-iif}(w_i) = \text{pf}(w_i) \times \text{iif}(w_i), \quad (9)$$

where the inverse image frequency  $\text{iif}(w_i)$  is a measure of the importance of the phrase, and it is estimated as

$$\text{iif}(w_i) = \log \left( \frac{|\mathcal{I}|}{|\{I : w_i \in L_I\}|} \right), \quad (10)$$

where  $|\mathcal{I}|$  is the total number of web images in the database,  $|\{I : w_i \in L_I\}|$  is the number of web images whose associated phrase lists contain the phrase  $w_i$ . The idea of using the inverse image frequency to reduce the importance weights of the commonly occurring phrases is very similar with the idea which is very popular in the information retrieval society. That is, the commonly occurring text terms are less discriminative for text document representation and the inverse document frequency is used to reduce their importance weights. Similarly, the commonly occurring phrases are less important on interpreting the semantics of web image as compared with the rarely occurring phrases. By computing the pf-iif value for each phrase  $w_i$ , the rank scores for all the phrases can be initialized and they can be used to identify more relevant phrases for interpreting the semantics of the given web image.

Ideally, we attempt to extract the phrases from the associated texts which are more relevant with the semantics of web images. The intuitive observation is that the phrases in the surrounding texts and meta-data are often semantically related to the semantics of web images. In addition, we quantitatively assessed the relatedness



Figure 6: Some web images and their associated phrases ranked by the pf-iif value.

between the extracted phrases and the semantics of web images. We randomly pick up 1508 web images and their associated phrase lists. For each web image, every phrase in the associated list is manually judged whether it is relevant with its semantics. We then calculated the accuracy as the number of the relevant phrases divided by the number of all the phrases in the list. The average accuracy over the 1508 web images is 0.3045 which indicates that one third of the extracted phrases are semantically relevant to the semantics of web images. Noting that the accuracy is measured without considering the phrases' positions in the list. It is safe to conclude that the extracted phrases are related to the semantics of web images. In Fig. 6, we presented some examples of web images and their associated phrase lists in which the phrases are ranked by their pf-iif values.

### 3.3 Automatic Image-Text Alignment

Ranking the phrases according to their pf-iif values can capture the intuition that the most co-occurring phrases are often related to the semantics of web images. The appearance of noise phrases may make ranking the phrases by their pf-iif values to be far from satisfaction, thus it is very important to develop a re-ranking method for achieving more accurate image-text alignment. In this section, an automatic image-text alignment approach is developed by performing near-duplicate image clustering and relevance re-ranking.

#### 3.3.1 Near-Duplicate Image Clustering

Our image-text alignment algorithm essentially relies on the observation that the most relevant phrases for a web image are very likely to re-appear on the phrase lists for its near-duplicates. By grouping near-duplicate web images into the same cluster, their most significant phrases which appear in the phrase lists of all these near-duplicate web images in the same cluster can be aggregated to achieve more accurate interpretation of their semantics. It is well known that the visually similar images are not necessary to be semantically similar due to the problem of semantic gap [46]. On the other hand, near-duplicate web images are strongly relevant on their semantics because they are usually generated from the same image via manual editing. When a web image or its near-duplicates appear in multiple web pages, its semantics may be described from different perspectives, *e.g.*, various web page content creators may use their own vocabularies. As shown in the first group of near-duplicate web images of Fig. 7 (Group 1), even the auxiliary text phrases for each near-duplicate web

image are different, some common phrases repeatedly occur on their auxiliary text documents. Thus we can achieve more accurate image-text alignment by aggregating such frequently co-occurring phrases.

For each cluster of near-duplicate web images, we aggregate all these phrases from multiple individual phrase lists by using three different strategies (See Section 3.3.2) for interpreting the semantics of near-duplicate web images in the same cluster. In practice, this method may fail in two extreme cases: 1) some unpopular web images might not have any duplicate in the database or even on the web; and 2) the associated text phrases for each near-duplicate web image are almost the same (see Group 2 in Fig. 7) even these phrases are derived from different web pages. For example, the web content creators sometimes simply copy the image and its associated texts from other reference web pages without changing anything.

To evaluate how much percentage of web images in our database have near duplicates, we randomly selected 80,000 web images from our database and found that about 21% of them have more than 3 near duplicates and around 3% have more than 10 near duplicates. This statistics indicates that more accurate image-text alignment can be envisioned by performing near-duplicate image clustering because more than one fifth web images in our database are duplicated.

Near-duplicate image detection has been intensively investigated recently. One of the key components for such a system is to use locality sensitive hashing (LSH) [30] or its variants [26] to approximate nearest-neighbor matching in high dimensional space. LSH devises a family of hash functions to ensure that the collision probability of two data points is inversely proportional to their distance in the original feature space.



	Associated Text	Top Ranked Phrase
Group 1  52 duplicates	<b>Alt:</b> Title: Name: air-jordan-xi-cool-grey-confirmed-1 <b>Text:</b> Air Jordan XI (11) Retro – Cool Grey – Holiday 2010 Release Confirmed	jordan, air, summer sample, nike air maxim torch, air jordan, ...
	<b>Alt:</b> Retro Air Jordan 11 Retro <b>Title:</b> Retro Air Jordans <b>Name:</b> jordan_retro_11 <b>Text:</b> Air Jordan 11 Retro White/Black-Dark Concord White/Columbia Blue-Black Black/Varsity Red-White Black/Varsity Red-Dark Charcoal (low) White/Cobalt-Zen Grey (low)	air jordan, retro, white, ...
	<b>Alt:</b> Title: Name: air-jordan-xi-cool-grey-confirmed-1 <b>Text:</b> Remember the Defining Moments Package from 2006? Well, 2010 appears to be the Defining Moments Year. The Air Jordan VI Varsity ...	jordan, air, air jordan, package, ...
Group 2  4 duplicates	<b>Alt:</b> KHUFU PYRAMID T-SHIRT <b>Title:</b> KHUFU PYRAMID T-SHIRT by Dianekmt <b>Name:</b> khufu_pyramid_t-shirt-p2351751824823459172r1oa_125 <b>Text:</b> KHUFU PYRAMID T-SHIRT	pyramid shirt, khufu, pyramid, khufu pyramid, shirt,
	<b>Alt:</b> KHUFU PYRAMID T-SHIRT <b>Title:</b> KHUFU PYRAMID T-SHIRT by Dianekmt <b>Name:</b> khufu_pyramid_t-shirt-p2351751824823459172r1oa_125 <b>Text:</b> KHUFU PYRAMID T-SHIRT	pyramid shirt, khufu, pyramid, khufu pyramid, shirt,
	<b>Alt:</b> KHUFU PYRAMID T-SHIRT <b>Title:</b> KHUFU PYRAMID T-SHIRT by Dianekmt <b>Name:</b> khufu_pyramid_t-shirt-p2351751824823459172r1oa_125 <b>Text:</b> KHUFU PYRAMID T-SHIRT	pyramid shirt, khufu, pyramid, shirt, khufu pyramid,

Figure 7: Examples of near-duplicate image groups, associated text and top ranked extracted phrase.

Given a feature vector  $\mathbf{x} = (x_1, \dots, x_d)^T$  of an image, each of the hash functions  $h_i(\cdot), i = 1, 2, \dots, k$  maps it onto the set of integers, mathematically expressed as

$$h_i(\mathbf{x}) = \left\lfloor \frac{\mathbf{a} \cdot \mathbf{x} + b}{c} \right\rfloor, \quad (11)$$

where  $c$  is a preset bucket size,  $\mathbf{a}$  is a  $d$  dimension vector whose entries independently chosen from a  $p$ -stable distribution and  $b$  is a scalar uniformly sampled from the range  $[0, c]$ . In our implementation, we choose using the 1-stable Cauchy distribution and  $c$  is empirically set as 1. The Cauchy distribution-based hash function exhibits the property that the collision probability monotonically decreases with the  $L_1$  distance between two feature vectors. That is, the closer of two vectors  $\mathbf{x}_1$  and  $\mathbf{x}_2$  in the original space (  $\|\mathbf{x}_1 - \mathbf{x}_2\|_1$  is smaller), the more likely their hashing signatures are

the same. A single hash is usually not sufficiently discriminative to support matching and thus we group  $k$  hashing values to form a  $k$ -tuples which is mapped into a hash table by using the second level of standard hashing. Web images in the same bucket, *i.e.*, web images having the same  $k$ -tuples, are deemed potential near-duplicates which are further verified by using the Euclidean distance among them.

Technically, any visual features can be adopted to represent the visual content of an image. In this work, we use the bag-of-feature model to capture both color and local gradient information of an image. Specifically, a color-based and a SIFT (Scale Invariant Feature Transform [98]) based codebook is trained, respectively. To learn the color dictionary, we randomly select 20,000 images from our database and then partition each image into a set of  $4 \times 4$  image patches with step size of 4. For each image patch, its average RGB color is computed. We then pool all the average color vectors as the training samples to learn a 768-dimensional color dictionary by using  $k$ -means. Following a similar process, we trained a SIFT-based codebook which contains 2048 visual words. Finally, the visual feature vector of an image is produced by concatenating the two codebook-based histograms with  $l_2$  normalization.

### 3.3.2 Phrase Aggregation

While the lists of candidate phrases are available for each web image within a particular cluster of near-duplicate web images, a phrase aggregation step is needed to merge all the phrase lists into a single ranking list, and the top ranked ones are then used to interpret the semantics of near-duplicate web images in the same group. In this section, we define three aggregation methods, namely naive voting, borda voting

[128] and summing, for this purpose.

Let  $\{L_i\}_{i=1}^n$  be the phrase lists associated with  $n$  near-duplicate web images in the same cluster,  $U = \bigcup_{i=1}^n L_i$  the union set of all unique candidate phrases. The three aggregation methods are described as following.

*Naive Voting:* For each candidate phrase  $w \in U$ , the naive voting strategy assigns 1 to its vote whenever  $w \in L_i$ , and 0, otherwise.

$$vote(w, i) = \begin{cases} 1 & \text{if } w \in L_i, \\ 0 & \text{otherwise.} \end{cases} \quad (12)$$

A single list of phrases  $S$  is obtained by sorting the candidate phrases in terms of the number of votes, namely the score, given as

$$score(w) := \sum_{i=1}^n vote(w, i). \quad (13)$$

*Borda Voting:* Unlike the naive voting which ignores the ranking positions of the phrases in the candidate lists, the borda voting strategy ranks the candidate phrases by taking into account the ranking information. In each phrase list  $L_i$ , the phrases are assigned a decreasing number of points. That is, the top-ranked phrase is given  $k - 1$  points, the second-ranked one receives  $k - 2$  points and so on until the last one is given no points. In many cases, phrase lists have various number of phrases, and we set  $k$  to be the size of the longest list among all the  $n$  lists. Finally, the candidate phrases are sorted in a decreasing order in terms of points that they received, given as

$$score(w) := \sum_{i=1}^n point(w, i), \quad (14)$$



where  $point(w, i)$  is the point that  $w$  received in list  $L_i$ .

*Summing:* The summing strategy also considers the ranking information of each phrase. For the phrases in the candidate union, the summing strategy sums over their pf-iff values (See Section 3.2.3). Thus the score of a phrase  $w \in U$  is calculated as

$$score(w) := \sum_{i=1}^n pf\text{-}iif(w, i), \quad (15)$$

where  $pf\text{-}iif(w, i)$  is the pf-iff value of the phrase  $w$  in list  $L_i$ .

By aggregating the phrases from multiple phrase lists which are associated with the near-duplicate web images in the same cluster, a single phrase ranking list is produced where the top ranking ones are then used to index all the near-duplicate web images in the same group. It turns out that borda voting and summing strategies outperform the naive voting strategy on this task (See Section 3.4.2). In Fig. 8, we present some groups of near-duplicate web images, their associated phrase lists and the aggregated single list produced by the summing strategy.

### 3.3.3 Relevance Re-ranking

When people compose their cross-media web pages, they often interchangeably use different text words or phrases with similar meanings to interpret the semantics of a web image. On the other hand, some phrases have multiple different but related senses under various contexts. Thus text phrases are strongly inter-related in terms of different levels and ways, which could be exploited to further enhance phrase ranking. In this section, a phrase correlation network is generated to characterize such inter-phrase similarities intuitively, and a random walk process is performed over the phrase

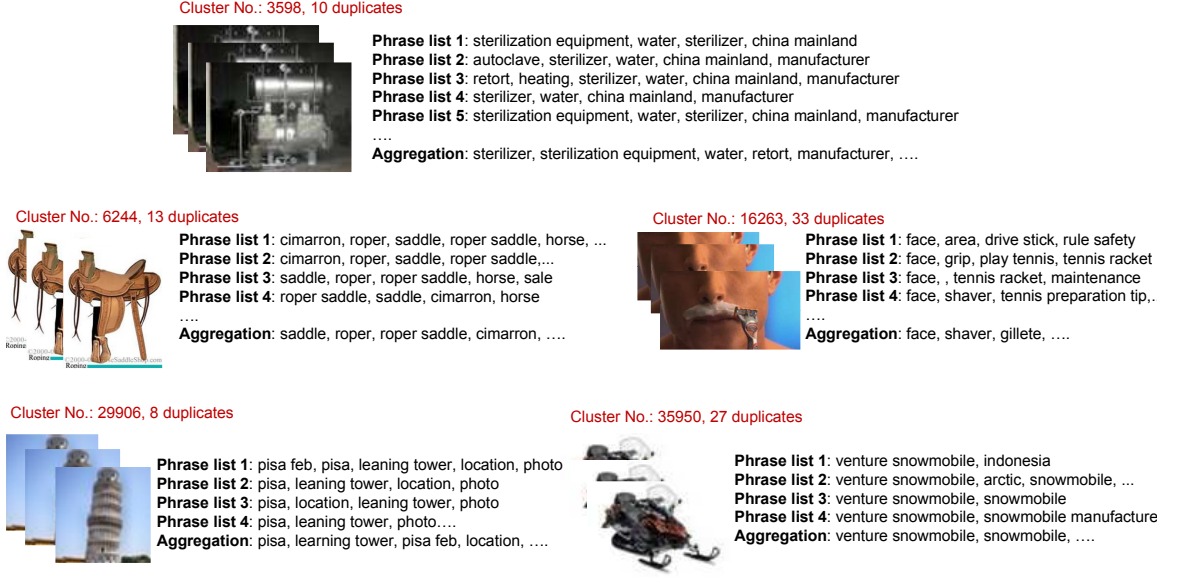


Figure 8: Image clusters and associated phrase lists. The single ranking list is aggregated by the summing strategy.

correlation network to further re-rank the phrases.

### 3.3.3.1 Phrase Correlation Network

To estimate the correlation between words or phrases for enhancing web image indexing, a myriad of related efforts have been reported in the literature, including the use of the ontology WordNet [74] and the normalized Google distance(NGD) derived from the Word Wide Web [157]. In this work, we consider two types of information sources to characterize the correlation between a pair of phrases, including inter-phrase co-occurrence and inter-phrase semantic similarity derived from WordNet. Specifically, given two phrases  $w_i$  and  $w_j$ , their co-occurrence correlation  $\beta(w_i, w_j)$  is defined as

$$\beta(w_i, w_j) = -P(w_i, w_j) \log \frac{P(w_i, w_j)}{P(w_i) + P(w_j)} \quad (16)$$

where  $P(w_i, w_j)$  is the co-occurrence probability for two text terms or phrases  $w_i$  and  $w_j$ ,  $P(w_i)$  and  $P(w_j)$  are the occurrence probabilities for the phrases  $w_i$  and  $w_j$  [47]. The semantic similarity  $\gamma(w_i, w_j)$  is defined as the Leacock and Chodorows similarity [80],

$$\gamma(w_i, w_j) = -\log \frac{L(w_i, w_j)}{2D} \quad (17)$$

where  $L(w_i, w_j)$  is the number of nodes along the shortest path between  $w_i$  and  $w_j$  on the WordNet hierarchy and  $D$  is the maximum depth of the WordNet taxonomy. Some specific phrase which are not in the WordNet taxonomy are omitted. Finally, the cross-modal inter-term correlation between the phrases  $w_i$  and  $w_j$  is defined as

$$\phi(w_i, w_j) = \alpha \cdot \gamma(w_i, w_j) + (1 - \alpha) \cdot \beta(w_i, w_j) \quad (18)$$

where  $\alpha = 0.5$  is the weighting parameter. The text phrases, which have large values of the cross-modal inter-phrase correlations, are connected to form a phrase correlation network. A part of the phrase correlation network for our database of large-scale web images is shown in Fig. 9, where each text term or phrase is linked with multiple most relevant text terms or phrases with larger values of  $\phi(\cdot, \cdot)$ . Such phrase correlation network could provide a good environment for addressing the issues of polysemy and synonyms more effectively and disambiguating the image senses precisely. This in turn allows us to identify more relevant text terms or phrases to interpret the semantics of web images.

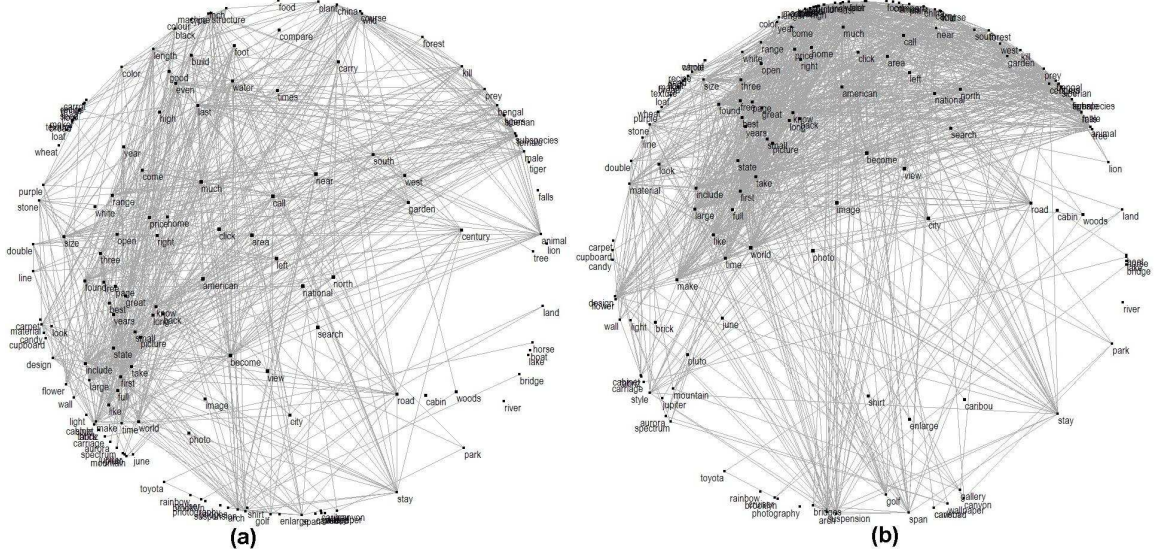


Figure 9: Two alternative views of the hyperbolic visualization of the phrase correlation network. Only 120 terms are shown in this picture to avoid visualization clutter.

### 3.3.3.2 Random Walk for Relevance Re-Ranking

In order to leverage the advantage of our phrase correlation network for learning the rich semantics of web images more precisely, *i.e.*, achieving more precise alignments between the semantics of web images and their auxiliary text terms or phrases, a random walk process is further performed over the phrase correlation network for refining the relevance scores. Given the phrase correlation network for  $n$  most significant text terms or phrases, we use  $\rho_k(w)$  to denote the relevance score for the text term or phrase  $w$  at the  $k$ th iteration. The relevance scores for all these phrases on our phrase correlation network at the  $k$ th iteration will form a column vector  $\overrightarrow{\rho(w)} \equiv [\rho_k(w)]_{n \times 1}$ . We further define  $\Phi$  as an  $n \times n$  transition matrix, where each element  $\phi_{ij}$  is used to define the probability of the transition from the  $i$ th text term or phrase

to its inter-related  $j$ th text term or phrase and  $\phi_{ij}$  is defined as:

$$\phi_{ij} = \frac{\phi(i, j)}{\sum_k \phi(i, k)} \quad (19)$$

where  $\phi(i, j)$  is the pairwise inter-phrase cross-modal similarities between the  $i$ th and  $j$ th phrase. We then formulate the random walk process as

$$\rho_k(w) = \theta \sum_{j \in \Omega_j} \rho_{k-1}(j) \phi_{tj} + (1 - \theta) \rho(C, w) \quad (20)$$

where  $\Omega_j$  is the first-order nearest neighbors of the  $j$ th text term or phrase on the phrase correlation network,  $\rho(C, w)$  is the initial relevance score for the text term or phrase  $w$  and  $\theta$  is a weight parameter. The initial relevance score  $\rho(C, w)$  is estimated by normalizing the scores of all the phrases in the unique phrase union set  $U_C$  for the cluster  $C$ ,

$$\rho(C, w) = \frac{\text{score}(w)}{\sum_{t \in U_C} \text{score}(t)}, \quad (21)$$

where the  $\text{score}(\cdot)$  is the phrase scoring function by using summing strategy as (15).

The random walk process will promote the inter-related text terms or phrases that have many nearest neighbors on the phrase correlation network, *e.g.*, the inter-related phrases that have strong semantic correlations and higher co-occurrence probabilities. On the other hand, this random walk process will also weaken the isolated text terms or phrases on the phrase correlation network, *e.g.*, the isolated text terms or phrases that have weak semantic correlations and low co-occurrence probabilities with other text terms or phrases. This random walk process is terminated when the relevance scores converge or after several iterations.

For a cluster  $C$  of near-duplicate web images, we re-rank all its auxiliary text

terms or phrases according to their relevance scores after the random walk process terminates. By performing random walk over the phrase correlation network, our relevance score refinement algorithm can leverage both the inter-phrase co-occurrence similarity and the inter-phrase semantic similarity for re-ranking the auxiliary text phrases more precisely. Finally, the top- $k$  auxiliary text terms or phrases, which have higher relevance scores with the semantics of near-duplicate web images, are selected for interpreting the semantics of near-duplicate web images in the given cluster  $C$ . An alternative way is to select the top phrases whose scores are larger than a pre-defined threshold. Such automatic image-text alignment process can support better understanding of cross-media web pages (web images and their associated text documents) because it can couple different information sources together to resolve the ambiguities that may arise from performing single media analysis.

Some experimental results on relevance re-ranking for automatic image-text alignment are given in Fig. 10. From these experimental results, one can observe that our automatic image-text alignment algorithm can effectively identify the most relevant auxiliary text terms or phrases to interpret the rich semantics of web images sufficiently.

### 3.4 Experiments for Algorithm Evaluation

Following the method described in Section 3.2, we have created a database with 5,000,000 image-text pairs. We have also excluded some web images because their visual features cannot be extracted successfully. Our database for large-scale image-text pairs finally contains 4,651,972 web images and 986,735 unique phrases in total.

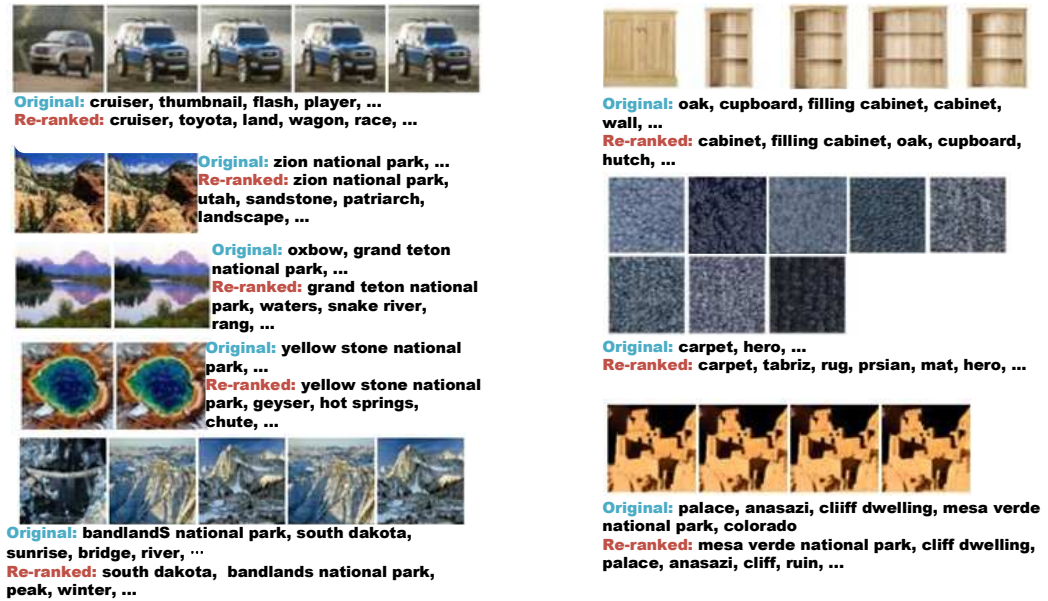


Figure 10: Example image clusters of original aggregated phrase lists and the re-ranked lists produced by our relevance re-ranking algorithm.

After removing those phrases occurring less than 10 times, 182,528 phrases were finally kept in our system. Unlike the vocabularies used in the traditional image annotation benchmarking databases where an annotation is often a single word or tag, such as “sky”, “sea”, and “car”, the phrases in our produced vocabulary are more specific and embodied richer semantics. A few sample phrases can be found in Fig. 6 and Fig. 8.

To evaluate our proposed methods, the recall and precision rates which are widely used in the image annotation and retrieval literature might be the desired performance metrics. Unfortunately, manually providing 182,528 phrases from such size of vocabulary to describe each web image in our large-scale database is tedious and time consuming. In this work, we used three alternative metrics, namely mean reciprocal rank (MMR), success at rank  $k$  ( $s@k$ ) and precision at rank  $k$  ( $p@k$ ), that capture the performance of the algorithm at different aspects. The MRR is defined as the

reciprocal of the rank of the first relevant phrase, averaged over all the testing photos, which indicates the ability of the proposed method to provide a semantically relevant phrase at the top of the ranking list. Success at rank  $k$  is 1 if a relevant tag is ranked in the top  $k$  results and 0 otherwise, averaged over all the test images. Precision at rank  $k$  is the number of relevant phrases ranked in the top  $k$  results, averaged over all test images as well. The relatedness between a phrase and an image is manually determined by users.

#### 3.4.1 Effectiveness of Image Clustering

In our work, near-duplicate image clustering is used to reduce the uncertainty between the semantics of web images and their auxiliary text terms or phrases. Thus it is very interesting to demonstrate the effectiveness of our method for near-duplicates image clustering on promoting the accuracy for automatic image-text alignment.

The 4,651,972 web images have been clustered into 216,896 groups of near-duplicate web images. The number of near-duplicate web images in each group ranges from 1 (no near-duplicate web images) to 383 (383 near-duplicate web images) and the average size is about 21 near-duplicate web images. Among all the clusters of near-duplicate web images, we randomly pick 1105 clusters whose numbers of near-duplicate web images are more than 4 (5,678 images in total) for algorithm evaluation. Some users without any background of this project were invited to manually evaluate the relatedness between the semantics of near-duplicate web images in the same cluster and each phrase on the ranking list which is produced by using the summing aggregation strategy. Specifically, if a phrase on the list is suitable to describe the semantics



Table 1: Performance comparison of the proposed method with and without performing near-duplicate image clustering.

	MMR	s@1	s@5	p@5
Without Image Clustering	0.7323	0.6233	0.8511	0.2935
With Image Clustering	0.8005	0.6857	0.9523	0.3581

of near-duplicate web images in the same group, score 1 would be assigned to that phrase, otherwise 0. In the case of without performing near-duplicate image clustering, each of these 5,678 web images in 1105 clusters is directly annotated with the phrases ranked by the pf-iif values (See Section 3.2.3). We also manually provides 1 and 0 to each phrase on the ranking lists for indicating whether it is applicable to describe the visual content of the associated web image. The comparison is tabulated in Table 1. One can clearly observe that performing near-duplicate image clustering has boosted the MMR, success and precision at rank  $k$  ( $k=1, 5$ ) significantly, which indicates that near-duplicate image clustering is particularly effective to find more relevant phrases and rank them at the top of the lists for image semantics interpretation.

#### 3.4.2 Comparison of Phrase Aggregation Strategies

As described in Section 3.3.2, three different strategies are used to aggregate the phrases from individual lists of phrases to form a single ranking list. We compared three strategies, *i.e.* naive voting, borda voting and summing and reported the results in Fig. 11. One can observe that the borda voting and summing methods are superior to the naive voting strategy in terms of all the three performance metrics. Therefore, we only reported results for the summing strategy when aggregation was applied.

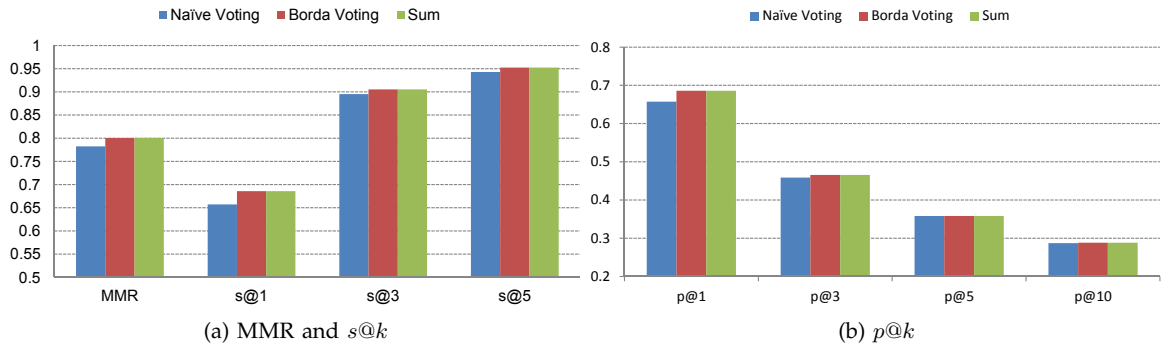


Figure 11: Performance comparison of different phrase aggregation strategies.

### 3.4.3 Effectiveness of Relevance Re-ranking

As discussed in Section 3.3.3.2, we have constructed a phrase correlation network which captures the inter-phrase co-occurrence and semantic similarity, so that we can re-rank all the phrases for each cluster of near-duplicate web images. The re-ranking is achieved by performing random walk over the phrase correlation network for all the clusters of near-duplicate web images. To assess the effectiveness of our relevance re-ranking algorithm via random walk, we randomly select 1,000 clusters of near-duplicate web images which have more than four duplicates, and compute three metrics, *i.e.* MMR,  $s@k$ , and  $p@k$  for two cases: with and without performing relevance re-ranking. The comparison is reported in Fig. 12. One can observe that the process of relevance re-ranking via random walk can further improve the results of automatic image-text alignment because our phrase correlation network can make use of multi-modal inter-phrase correlations to address the issues of polysemy and synonyms effectively.

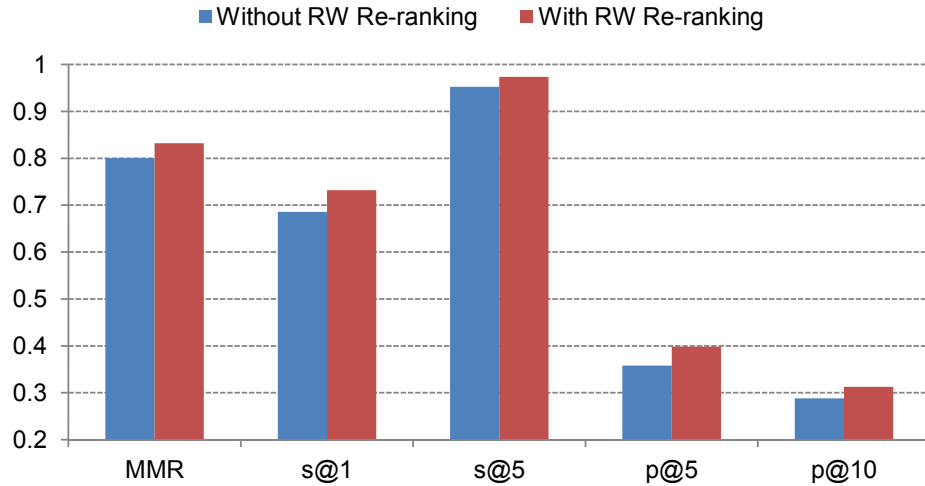


Figure 12: Performance comparison of image-text alignment with and without performing random walk (RW) for relevance re-ranking.

#### 3.4.4 Performance on Web Image Indexing and Retrieval

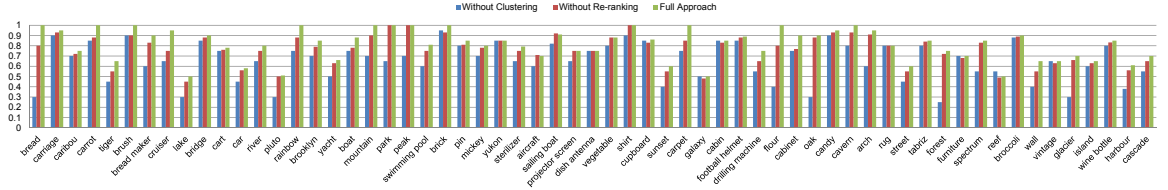
We also evaluate the effectiveness of our automatic image-text alignment algorithm for web image indexing and retrieval. We index each web image by using its ranking list of phrases and support web image retrieval by using their ranking list of phrases. Given a query term or phrase, the system returns all the web images that are associated with the text term or phrase and ranks them according to their relevance scores. The precision for top  $k$  returned images, denoted as  $\text{Pre}@k$ , is computed and the average precision over all the query phrases is reported. A user-friendly interface is designed for users to determine whether a returned image is relevant to the query phrase or not. Mathematically, the  $\text{Pre}@k$  is defined as

$$\text{Pre}@k = \frac{\sum_{i=1}^k \delta(i)}{k}, \quad (22)$$

where  $\delta(\cdot)$  is an indicator function which equals to 1 if the  $i$ th web image is relevant to the query and 0 otherwise.

To assess the effectiveness of image clustering and relevance re-ranking on automatic image-text alignment, we have compared the performance of our automatic image-text alignment algorithm under three different scenarios: (1) near-duplication image clustering is not performed for reducing the uncertainty between the relatedness between the semantics of web images and their auxiliary text terms or phrases; (2) random walk is not performed for relevance re-ranking; and (3) both near-duplicate image clustering and random walk are performed for learning the rich semantics of web images more precisely by achieving more accurate image-text alignments. We randomly select 61 phrases as queries from the vocabulary. As shown in Fig. 13, where the precision for top 20, 40 and 60 returned images are evaluated respectively. One can observe that incorporating near-duplicate image clustering for uncertainty reduction and performing random walk for relevance re-ranking can significantly improve the accuracy rates for automatic image-text alignment.

For the same task on identifying the most relevant text terms or phrases for interpreting the semantics of web images, we have compared the performance between three approaches for image-text alignment: our unsupervised image-text alignment approach *versus* Berg’s approach [10] and cross-media relevance model proposed by Feng *et al.* and Lavrenko *et al.* [54, 78]. Both Berg’s approach and cross-media relevance model are supervised and require large amounts of labeled training images to learn joint image-word relevance models. In our experiments, we use large amounts of web images and their auxiliary text terms or phrases as the labeled images to train the joint image-word relevance models for both Berg’s approach and cross-media relevance model. We plot the performance comparison in Fig. 14 with respect to the



(a) Pre@20, top 20 images are evaluated. Average precision rates: without clustering, 0.6497; without re-ranking, 0.7659; full approach, 0.8141.



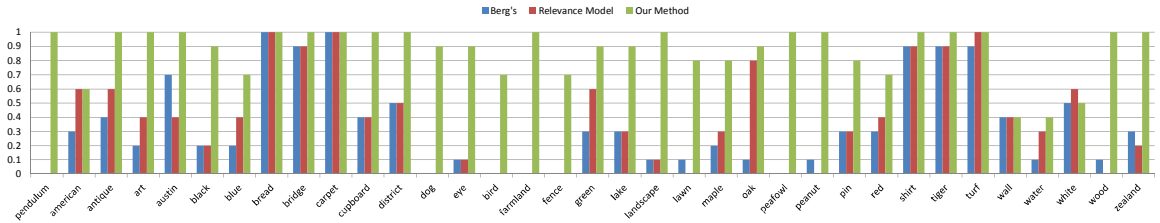
(b) Pre@40, top 40 images are evaluated. Average precision rates: without clustering, 0.5828; without re-ranking, 0.6853; full approach, 0.7279.



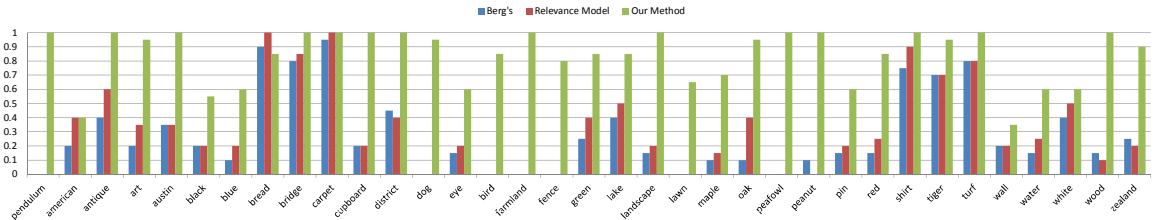
(c) Pre@60, top 60 images are evaluated. Average precision rates: without clustering, 0.5054; without re-ranking, 0.6332; full approach, 0.6819.

Figure 13: Performance comparison in the application of image indexing and retrieval. Three cases are assessed: (1) image clustering and relevance re-ranking are not adopted, denoted as “without clustering”; (2) image clustering is used but random walk is not adopted, denoted as “without re-ranking” and (3) both image clustering and random walk are used, denoted as “full approach”.

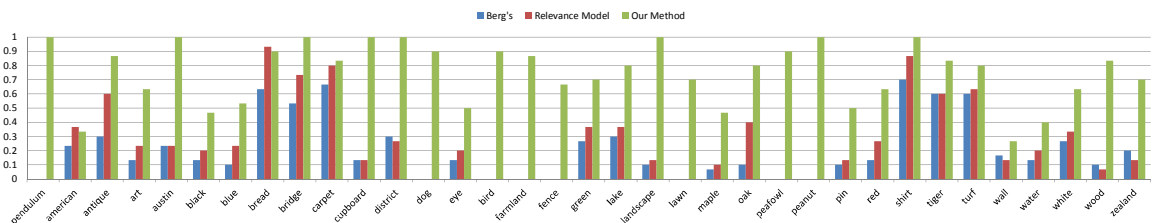
precision rates in the scenario of image indexing where 35 random query words are evaluated. One can observe that our unsupervised image-text alignment approach can significantly improve the precision rates by identifying the most relevant text terms or phrases for image semantics description and web image indexing. On the other hand, the other two supervised approaches, the Berg’s approach and the cross-media relevance modeling approach, are not be able to identify the most relevant text terms or phrases for web image indexing. One of the reasons is that loose image-text pairs, such as web images and their auxiliary text terms or phrases, are too noise to be



(a) Pre@10, top 10 images are evaluated. Average precision rates: Berg's method, 0.3371; Relevance model, 0.3886; Our method, 0.8714.



(b) Pre@20, top 20 images are evaluated. Average precision rates: Berg's method, 0.2771; Relevance model, 0.3285; Our method, 0.8319.



(c) Pre@30, top 30 images are evaluated. Average precision rates: Berg's method, 0.2105; Relevance model, 0.2762; Our method, 0.7533.

Figure 14: Performance comparison with Berg's method and the relevance model in the application of image indexing and retrieval.

treated as the reliable training source to accurately learn the joint image-word relevance models. Without depending on the labeled training images (training images with good relatedness between the annotation text terms or phrases and their semantics), our unsupervised image-text alignment approach can achieve higher precision rates.

Achieving more accurate indexing of large-scale web images, *i.e.*, identifying more relevant auxiliary text terms or phrases for interpreting the semantics of web images more accurately, will result in high precision rates for keyword-based web image re-

trieval, thus we can use the precision rates for keyword-based web image retrieval to evaluate the performance of various image-text alignment algorithms for web image indexing. For some queries as shown in Fig. 14, both the Berg’s approach and the cross-media relevance modeling approach may result in zero precision rates, *e.g.*, both the Berg’s approach and the cross-media relevance modeling approach cannot identify the most relevant auxiliary text terms or phrases for web image indexing because they cannot learn the joint image-word models accurately by using large amounts of loose image-text pairs.

Our unsupervised image-text alignment approach can achieve significant improvement on the precision rates and it benefits from three components: (1) Near-duplicate image clustering is performed to reduce the visual ambiguity between the web images and provide a good environment to effectively reduce the uncertainty on the relatedness between the semantics of web images and their auxiliary text terms or phrases; (2) A phrase correlation network is constructed to tackle the issues of polysemy and synonyms more effectively; (3) A random walk process is performed over the phrase correlation network for relevance re-ranking and achieving more precise alignments between the semantics of web images and their auxiliary text terms or phrases.

### 3.4.5 Quality of Labeled Training Images

As discussed in Section 3.1, one of our motivations is to automatically collect large-scale labeled image for classifiers training. In this section, we quantitatively show that our proposed approach is able to collect large-scale training images with reliable labels. To assess the quality of labeled training images which are collected

Table 2: The 80 concepts selected for classification evaluation.

acropolis	airplane	bait	bear	bread
bread machine	brick	bridge	broccoli	brush
candle	carpet	caribou	carriage	carrot
cormorant	cruiser	cupboard	dinner napkin	dish aerial
elephant	elk	eyeglasses	flower	football helmet
fountain	giant panda	goblet	grape	horse
hummingbird	hydrometer	kangaroo	knife	komodo dragon
laminated	lichen	lizard	locomotive	mandolin
meerkat	mini van	monitor	motorbike	oyster
pacifier	pajama	pan cake	peach	peafowl
pencil sharpener	pendulum clock	persimmon	pineapple	pingpong paddle
pingpong table	puffin	pumpkin	razor	riverbank
saddlebag	sailing boat	scooter	screwdriver	spatula
speedometer	sterilizer	sushi	table	tambourine
tennis ball	tent flap	tiger	toilet paper	t-shirt
volcano	volleyball	whale	wrench	zipper

by our proposed approach, we conducted the following experiments. We choose 80 concepts from the vocabulary of our database and each of them is a synset in the ImageNet [33], so that we can have manually labeled images for each object concept for algorithm evaluation. The 80 concepts are tabulated in Table 2. Given a concept, we treat it as a query to find top ranked images from our database and collect them as training instances for that concept. We organize our labeled training images in three different stages, so that we can clearly see the effectiveness of each component. The three stages are 1) images are labeled by using the phrases which are ranked only by pf-iif values, *i.e.* without image clustering; 2) images are labeled by the ranked phrases which are determined by only performing near-duplicate image clustering but without performing relevance re-ranking; and 3) image are labeled by using the phrases which are determined by performing both near-duplicate image clustering and relevance re-ranking and we refer to it as full approach. Fig. 15 presents some labeled training images which are collected by our full approach for some object concepts.



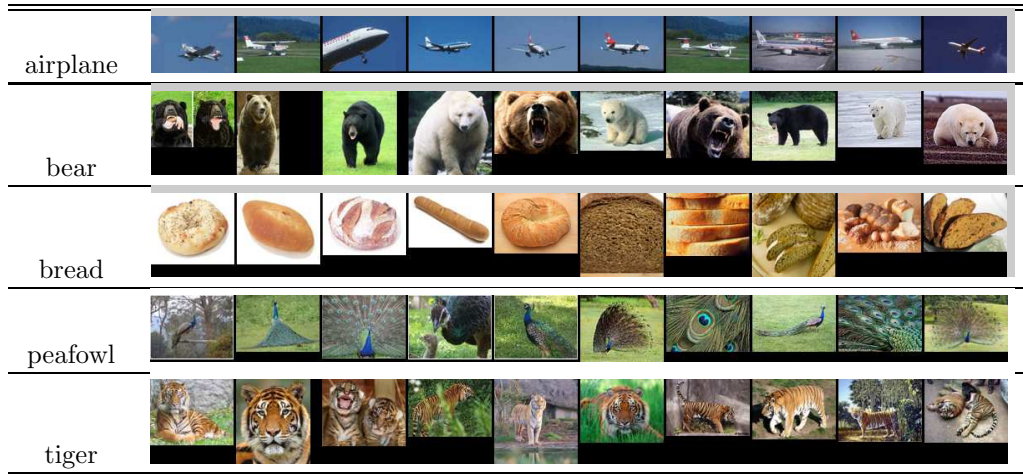


Figure 15: Sample training images collected by our algorithm for several object concepts.

For each concept, we use 200 images from the ImageNet data set as testing images.

We follow the standard bag-of-features pipeline to extract local SIFT features, encode the features using a codebook of 2048 visual words and pool the codes with a two-level spatial pyramid configuration to form the visual representation for each image. We then train one-against-all linear SVMs on the pooled features for achieving multi-class image classification due to its efficiency. In Fig. 16, we plot the average classification accuracy rates, *i.e.* the mean of the diagonal of confusion matrix, across different number of labeled training images. One can observe that our algorithm can obtain reliable labeled training images for classifier training, thus our algorithm can automatically harvest large-scale labeled training images from the Internet. Each component of our full approach, *i.e.* performing both near-duplicate image clustering and relevance re-ranking with phrase correlation network, can boost the performance by some margins and near-duplicate image clustering can improve the accuracy rates dramatically as comparing with the baseline method (“without clustering” in Fig.

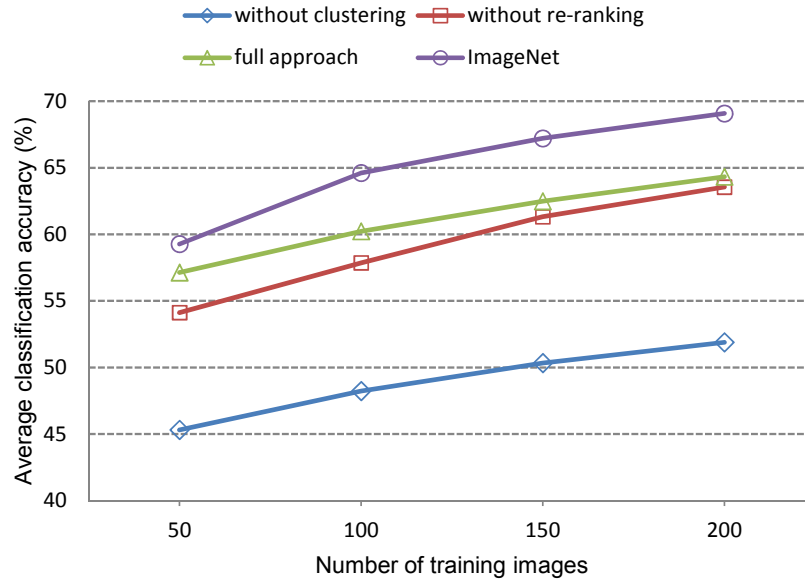


Figure 16: Image classification accuracy comparison between the methods used to collect training images.

16). However, the performance gap between two approaches for classifier training is still considerable: training images are manually labeled *versus* training images are automatically labeled by the auxiliary text terms or phrases which are determined by our algorithm. Thus leveraging large-scale web images and their auxiliary text terms or phrases to construct a unbiased and reliable image database is not a trivial task.

### 3.5 Summary

In this chapter, we have presented an unsupervised algorithm for automatic image-text alignment and relevance re-ranking to learn rich semantics of large-scale web images from their auxiliary text terms or phrases. It can lead to many potential applications: (1) supporting more effective web image retrieval with higher precision rates by identifying more relevant text terms or phrases for web image indexing; (2) by harvesting from the Internet, our research can create large-scale labeled image sets for achieving more accurate training of large numbers of classifiers, which is a

long-term goal of the multimedia research community; and (3) achieving cross-media alignment automatically and generating a parallel cross-media corpus (image-text pairs), which may provide many opportunities for future researches such as word sense disambiguation and image sense disambiguation. We will release our parallel cross-media corpus (image-text pairs) and source codes in the future.

## CHAPTER 4: VISUAL TREE FOR EFFICIENT CATEGORIZATION

### 4.1 Introduction

Image categorization is one of the fundamental problems in the field of computer vision. A great progress has been made in the past decades, especially on image sets of moderate sizes, such as Caltech101 [51] and Caltech256 [64]. Recent proliferation of user-contributed digital images on the Internet has created the need as well as new chances for tools to recognize a large number of visual categories. For example, ImageNet ([www.image-net.com](http://www.image-net.com)) has collected more than 14M images for 22K categories (classes/concepts). However, large-scale image classification which categorizes massive amounts of images into a large number of image categories poses tremendous computational challenges if a flat approach is simply employed (*e.g.*, image categorization is achieved by using many one-vs-rest (OVR) SVMs). For a testing image, the computational cost of the flat categorization approach grows linearly with the number of categories, which prohibits the widespread use of it in many applications with real-time or near real-time requirements.

One way to reduce the computational cost of a large-scale image categorization system is to organize the categories hierarchically in a tree structure by exploiting their inter-category correlations. In [9], Bengio *et al.* proposed a label tree model for this purpose. In a label tree, each tree node is associated with a number of classes

and as well as a predictor. For a given tree node, the set of the associated categories is required to be a subset of the classes which are associated with its parent node, and its node predictor is used to determine the best-matching child node to visit at the next level. Each leaf node corresponds to a particular image category. Given a testing sample, the label prediction process starts from the root node, and traverses down to a leaf node in the tree. The label of the leaf node is then taken as the prediction to it. This hierarchical label prediction process often leads to sub-linear time complexity in terms of the number of categories because a testing sample is only needed to be evaluated using a limited number of node predictors along the traversal path in the tree.

To construct a label tree for  $N$  image classes, one has to train  $N$  OVR support vector machines (SVMs) in advance, and then obtain a confusion matrix by evaluating all the  $N$  OVR classifiers based on a validation set. The confusion matrix is then utilized to build the label tree as a surrogate to approximate the inter-category correlations. However, learning the  $N$  OVR SVMs is very computationally expensive, especially when the number of categories is large. In addition, it often suffers from the class imbalance problem. That is, for a given image class (positive class), the negative instances from the other  $N - 1$  categories can heavily outnumber the positive instances. The negative instances may easily control and mislead the process of classifier training since the negative instances are numerous and the visual diversity of them is huge. In a word, the issue of class imbalance often results in unreliable OVR classifiers which further produces a misleading confusion matrix for learning the structure of a label tree. In [34], Deng *et al.* proposed a more efficient algorithm

for learning a label tree where the partition of the label set and the predictor of a tree node are learned at the same time. Given a tree node, the partition of its label set and the training of its node classifier are performed iteratively starting from a random initialization. The confusions of the node classifiers are essentially used to determine the tree structure.

In this paper, we propose a visual tree model which resorts to the visual similarities among image categories instead of the confusions yielded by pre-trained classifiers as the surrogate to approximate the correlations of image classes for tree structure learning. Specifically, we first estimate the visual representation of an image category by computing the mean feature of a number of images within that category. Second, we calculate the inter-category visual similarities between all the categories by taking their visual representations as input. Finally, we adopt clustering tools to partition the categories based on the inter-category visual similarities, and build the visual tree structure level by level. As the visual correlations between categories are used as the cue to construct the tree structure, we thus refer to this method as *visual tree* in the rest of the paper.

Noting that the mean feature which uses only a single feature prototype to visually characterize an image category might be of limited power in expressing the visual diversity of an image class, we introduce an extension in Section 4.2.1, which uses multiple feature prototypes per category for characterizing the category-specific visual properties of an image class. Also, we empirically investigate how the number of prototypes affect the performance of the visual tree model.

The underlying conjecture of using the inter-category visual correlations for visual

tree construction is that the visual similarities can be used as a proxy to approximate the learnability (*i.e.*, the possibility of them for being effectively distinguished by a classifier) of a label set (*i.e.*, a set of image categories). Intuitively, if two image categories are more visually similar, they are more difficult to be discriminated by a classifier [46]. Therefore, it is better to put the visually similar categories into the same set instead of partitioning them into different sets at the high-level nodes of a tree. Postponing the decision for label set partition (node splitting or image category partition) can avoid wrong partition of the label set at a high-level tree node and increase the learnability of the label set, which in turn helps us to train the node classifiers more accurately.

While hierarchical classification using a tree structure has a genuine advantage on computational efficiency, it usually degrades the categorization accuracy performance. If a mistake is made at a high-level node, it will be propagated to a leaf node and has no chance to be recovered. To alleviate this issue, a simple soft prediction scheme is utilized by exploring more branches if necessary. Specifically, we tend to choose a single node to visit if the current prediction is sufficiently confident, otherwise, both of the top-two scoring nodes (*i.e.*, nodes of the highest and the second-highest prediction scores) will be selected to visit at the next level. In summary, we make the following contributions in this chapter.

- A visual tree model is developed for organizing a large number of image categories hierarchically according to the inter-category visual correlations between them. Our method for the visual tree construction is very computationally efficient as compared to the traditional label tree method which requires obtaining

a cost-intensive confusion matrix in advance.

- The class mean feature and its extension are investigated for characterizing the visual representation of an image category. The impacts of them on the performance of the proposed visual tree method are empirically evaluated.
- A soft prediction scheme is designed for the tree-based hierarchical image classification to boost the categorization performance without incurring too much additional computational cost.

Experiments have been conducted on the ILSVRC2010<sup>2</sup> data set. Our experimental results have demonstrated that our visual tree can achieve very competitive or better results as compared to other tree models which demand training many OVR classifiers in advance or iteratively. Also, increasing the number of feature prototypes per class for characterizing the visual representation of an image category can improve the performance of our visual tree model if the number of feature prototypes is in a certain range. Third, the soft prediction scheme is effective for boosting the classification performance of tree models without re-training the node predictors.

## 4.2 Visual Tree Model

Given  $N$  image categories, we are interested in learning a tree structure  $T = (\mathbb{V}, \mathbb{E})$  which comprises a set of nodes  $\mathbb{V}$  and a set of edges  $\mathbb{E}$ , and each node  $v \in \mathbb{V}$  is associated with a set of category labels  $\mathbb{L}(v) \subseteq \{1, \dots, N\}$  as well as a predictor  $\mathbf{w}_v$  for selecting the best-matching child node to follow at the next level. In addition, the label set of a given node  $v$  is constrained to only contain a subset of the label set of

---

<sup>2</sup><http://www.image-net.org/challenges/LSVRC/2010>



its parent node.

To reduce the computational cost for visual tree construction, we develop a new algorithm on learning the tree structure by purely using the visual cues, *e.g.*, the inter-category visual similarities (correlations). In this section, we describe the details of our visual tree model, including characterizing the visual representations of image categories, estimating the inter-category visual similarities, learning the tree structure via hierarchical clustering and training the predictors associated with the tree nodes.

#### 4.2.1 Visual Representation of an Image Category

The problem of characterizing the visual appearance of an image category remains open partly due to the fact that finding an effective and yet efficient way to represent the visual content of an image has not been completely addressed. One way to estimate the visual representation of an image category is to resort to the overall visual properties of its relevant images. Let  $\mathcal{I}_i$  be a collection of relevant images of the  $i$ th image category. We can compute the mean visual feature by averaging the visual properties of all the images in  $\mathcal{I}_i$ , and take the mean feature as the overall characterization of the visual content of this image class. As illustrated in Fig. 17, we first represent the visual content of each image  $I_j \in \mathcal{I}_i$  using the *bag of visual feature* (BoF) model. Specifically, we encode the dense *histogram of gradient* (HoG) features using the locality linear coding (LLC) [155] over a dictionary, and then pool the codes to form an image-level signature  $\mathbf{h}_j$  for image  $I_j$  with max pooling. In our implementation, a dictionary of size 8,192 was used, and a two-level spatial pyramid partition, namely  $1 \times 1$  and  $2 \times 2$ , was employed to incorporate weak spatial

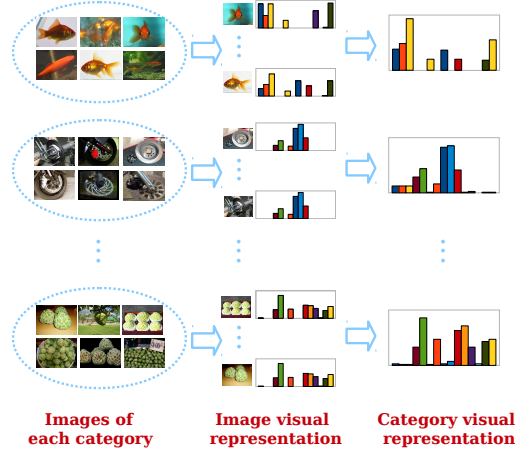


Figure 17: Schematic illustration of the mean feature representation of an image category. An image is represented as a bag of visual features (middle column); the mean feature vector w.r.t an category is then computed as the visual representation of that category (right column).

information. Second, the visual representation of the  $i$ th image category,  $\mathbf{H}_i$ , is defined as the class mean feature by averaging the image-level signatures of all the relevant images, given as

$$\mathbf{H}_i := \frac{1}{|\mathcal{I}_i|} \sum_{j \in \mathcal{I}_i} \mathbf{h}_j. \quad (23)$$

We then normalize the class mean feature vector  $\mathbf{H}_i$  to have unit  $l_2$  norm.

Considering the potential visual diversity of an image class, a single mean feature vector might be of intrinsically limited power to sufficiently express the visual appearance of an image category. We therefore extend the mean feature model to allow for more flexible category visual representations, which result in a model of stronger expression capability. The idea is to represent each category by a set of feature centroids, instead of only a single one class mean vector. Assume that we use  $K$  centroids for each class, the visual representation of the  $i$ th category can be defined as a set of

feature centroids, given as:

$$\mathbb{M}_i := \{\mathbf{m}_i^k\}_{k=1}^K, \quad (24)$$

where  $\mathbf{m}_i^k$  is the vector of a centroid. In practice, we can adopt  $k$ -means to group the image-level signatures  $\{\mathbf{h}_j\}_{j=1}^{|\mathcal{I}_i|}$  of the  $i$ th category into  $K$  clusters, and then take the cluster centroids to form the multiple feature centroids representation  $\mathbb{M}_i$ .

#### 4.2.2 Learning the Structure of a Visual Tree

The underlying conjecture of the visual tree model is that the visual correlations between the image categories (*i.e.*, inter-category visual similarities) can be used as a proxy to determine the learnability of a label set (*i.e.*, a set of image categories) [46]. The visually similar image categories are more likely to be confused by the node predictors. In other words, they are more difficult to be distinguished by the node classifiers. Assigning such visually similar image classes into the same label set would help to increase the learnability of the label set, which in turn allows us to learn the node predictors with higher accuracy rates. Here, we employ two hierarchical clustering techniques, namely hierarchical  $k$ -means and hierarchical spectral clustering, to learn the structure of a visual tree.

##### 4.2.2.1 Learning by using hierarchical $k$ -means

Given the mean feature representations  $\{\mathbf{H}_i\}_{i=1}^N$  of  $N$  image categories, we can use  $k$ -means clustering algorithm to partition the  $N$  image categories based on the class mean features, and build up a visual tree level by level. Specifically, for a particular node  $v$ , we can partition its label set  $\mathbb{L}(v) \subseteq \{1, \dots, N\}$  into  $B$  subsets according to the group assignments produced by the  $k$ -means algorithm. We then create  $B$  child

nodes, and each of the nodes takes one of the  $B$  label subsets as its own label set. For each of the child nodes, we go through the same procedure to generate its own child nodes until it is a leaf node which contains a single class label. Note that if the label set  $\mathbb{L}(v)$  of the particular node  $v$  has less than  $B$  labels, we only split it into  $|\mathbb{L}(v)|$  child nodes, instead of  $B$  nodes. We also control the maximum depth  $H$  of the visual tree. The depth of a tree node is defined as the length of the path from the root node to it. The root node has depth *zero*.

In Fig. 18, we demonstrate how to build a visual tree using the hierarchical  $k$ -means algorithm based on the mean feature representations. Assuming that there exists 10 classes, the root node contains the entire label set  $\mathbb{L}_{0,1} = \{1, 2, \dots, 10\}$ . We then run  $k$ -means by taking the corresponding mean features  $\mathbb{H}_{0,1}$  as input, and partition them into  $B$  clusters. To illustrate the visual tree construction process in this particular example, we set  $B$  to be 3. According to the cluster memberships of these ten categories produced by  $k$ -means, we split the root node into three child nodes accordingly, and each of them inherits a subset of the corresponding categories (labels) and a subset of the corresponding mean features as well. If a child node has more than  $B$  classes, for example the 3rd node at depth 1 (labeled with  $\mathbb{L}_{1,3}$  in Fig. 18), we split it into three child nodes by repeating the procedure above. We apply this hierarchical category partition strategy to every tree node until we build up a tree of desired depth. Finally, we summarize the learning procedure of constructing a visual tree by using hierarchical  $k$ -means in Algorithm 1.

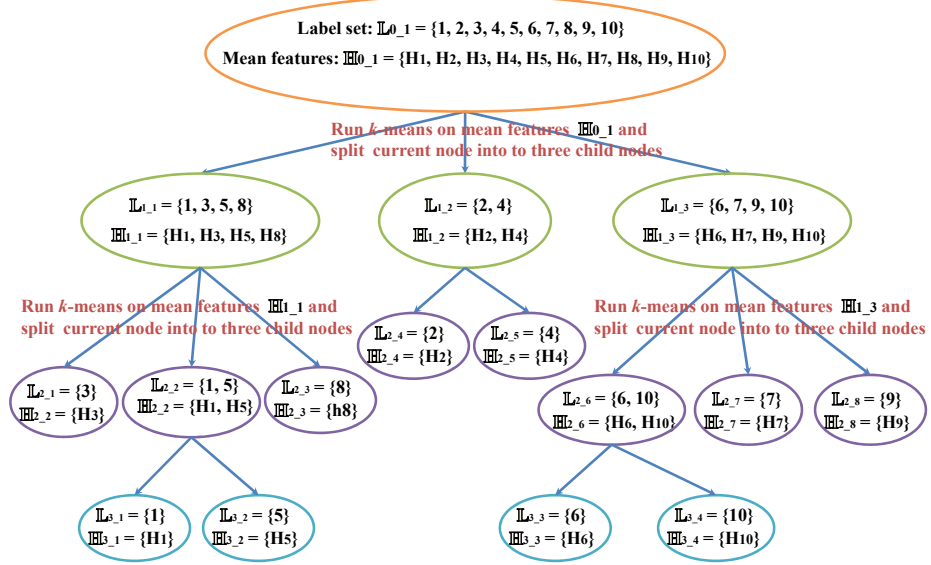


Figure 18: An example for illustrating the process of constructing a visual tree by using the hierarchical  $k$ -means based on the mean feature representations. Both of the branch-factor  $B$  and the maximum depth  $H$  are set to be 3, and each node is labeled with a pair of numbers,  $l_j$ , indicating the  $j$ th node at depth  $l$ .

---

Algorithm 1. Learning the structure of a visual tree structure by using hierarchical  $k$ -means

---

Input: Mean feature vectors  $\{\mathbf{H}_i\}_{i=1}^N$  of  $N$  image categories, the entire label set  $\mathbb{L} = \{1, \dots, N\}$  of the classes, branching factor  $B$  and maximum depth  $H$

- 1: Make a root node with depth 0 by taking  $\mathbb{L} = \{1, \dots, N\}$  and  $\{\mathbf{H}_i\}_{i=1}^N$  as its associated label set and mean feature set, respectively.
- 2: for  $h = 0, 1, \dots, H$  do
- 3:   for each node  $v$  on depth  $h$  do
- 4:     if  $|\mathbb{L}(v)| < B$  then
- 5:       Create  $|\mathbb{L}(v)|$  nodes  $\mathbb{C}(v)$  as the children of node  $v$  where  $\cup_{c \in \mathbb{C}(v)} \mathbb{L}_c = \mathbb{L}(v)$  and  $\mathbb{L}_i \cap \mathbb{L}_j = \emptyset, \forall i, j \in \mathbb{C}(v), i \neq j$
- 6:     else
- 7:       Partition the associated label set  $\mathbb{L}(v)$  into  $B$  disjoint subsets by using  $k$ -means based on the corresponding mean features  $\{\mathbf{H}_j\}_{j=1}^{|\mathbb{L}(v)|}$
- 8:       Create  $B$  nodes  $\mathbb{C}(v)$  as the children of node  $v$  where  $\cup_{c \in \mathbb{C}(v)} \mathbb{L}_c = \mathbb{L}(v)$  and  $\mathbb{L}_i \cap \mathbb{L}_j = \emptyset, \forall i, j \in \mathbb{C}(v), i \neq j$
- 9:     end if
- 10:   end for
- 11: end for

---

#### 4.2.2.2 Learning by using hierarchical spectral clustering

The visual tree, which is learned by performing hierarchical  $k$ -means algorithm over the mean features as described in Algorithm 1, is not guaranteed to be balanced.

However, a balanced tree is often desired if one wants to achieve the logarithmic complexity in tree-based image categorization. In [9], the spectral clustering algorithm [110] is used to build a label tree by recursively partitioning the confusion matrix because spectral clustering usually penalizes the unbalanced partitions. This inspires us to construct the proposed visual tree using the spectral clustering algorithm based on the inter-category visual similarities.

Given  $N$  image categories, we first compute the visual representations for them using the methods described in Section 4.2.1, including the class mean feature and its extension of using multiple centroids per class. We then compute the pair-wise visual similarity between two image classes based on their visual representations. The pair-wise similarity values among all the  $N$  image classes can be represented as an  $N$ -by- $N$  matrix  $\mathbf{S}$ , and an element  $S_{i,j}$  of the matrix represents the visual similarity value  $s(i, j)$  between the  $i$ th and the  $j$ th image categories, which can be defined as:

$$s(i, j) = \exp \left( -\frac{\text{dist}(C_i, C_j)}{\sigma} \right), \quad (25)$$

where  $\sigma$  is the bandwidth parameter which is chosen by using the self-tuning technique proposed in [172] and  $\text{dist}(C_i, C_j)$  is the distance between the  $i$ th and  $j$ th image classes. Note that  $\mathbf{S}$  is a symmetric matrix.

In case that the class mean feature vector is used as the visual representation of an image category, the distance  $\text{dist}(C_i, C_j)$  is defined as the Euclidean distance between the two mean feature vectors, given as

$$\text{dist}(C_i, C_j) = \sqrt{\sum_{d=1}^D (\mathbf{H}_i(d) - \mathbf{H}_j(d))^2}, \quad (26)$$

---

Algorithm 2. Learning the structure of a visual tree by using hierarchical spectral clustering

---

Input: Visual similarity matrix  $\mathbf{S} \in \mathbb{R}^{N \times N}$  of  $N$  image categories, the entire label set  $\mathbb{L} = \{1, \dots, N\}$  of the  $N$  classes, branching factor  $B$  and maximum depth  $H$

- 1: Make a root node with depth 0 by taking  $\mathbb{L} = \{1, \dots, N\}$  and  $\mathbf{S} \in \mathbb{R}^{N \times N}$  as its label set and visual similarity matrix, respectively
- 2: for  $h = 0, 1, \dots, H$  do
- 3:   for each node  $v$  on depth  $h$  do
- 4:     if  $|\mathbb{L}(v)| < B$  then
- 5:       Create  $|\mathbb{L}(v)|$  nodes as the children  $\mathbb{C}(v)$  of node  $v$  where  $\cup_{i \in \mathbb{C}(v)} \mathbb{L}_i = \mathbb{L}(v)$  and  $\mathbb{L}_i \cap \mathbb{L}_j = \emptyset, \forall i, j \in \mathbb{C}(v), i \neq j$
- 6:     else
- 7:       Partition the associated label set  $\mathbb{L}(v)$  into  $B$  disjoint subsets by using the spectral clustering algorithm [110] based on the corresponding affinity sub-matrix  $\mathbf{S}(v) \in \mathbb{R}^{|\mathbb{L}(v)| \times |\mathbb{L}(v)|}$
- 8:       Create  $B$  nodes as the children  $\mathbb{C}(v)$  of node  $v$  where  $\cup_{i \in \mathbb{C}(v)} \mathbb{L}_i = \mathbb{L}(v)$  and  $\mathbb{L}_i \cap \mathbb{L}_j = \emptyset, \forall i, j \in \mathbb{C}(v), i \neq j$
- 9:     end if
- 10:   end for
- 11: end for

---

where  $D$  is the dimension of the feature vector and  $\mathbf{H}_i(d)$  is the  $d$ th element of the vector  $\mathbf{H}_i$ . If an image class is represented using  $K$  feature centroids, the distance between the two categories is defined as

$$\text{dist}(C_i, C_j) = \frac{1}{K^2} \sum_{\mathbf{m}_i^p \in \mathbb{M}_i} \sum_{\mathbf{m}_j^q \in \mathbb{M}_j} \sqrt{(\mathbf{m}_i^p - \mathbf{m}_j^q)^T (\mathbf{m}_i^p - \mathbf{m}_j^q)}, \quad (27)$$

where  $\mathbf{m}_i^q$  is the  $p$ th centroid of the  $i$ th category and  $T$  denotes vector transpose. After the similarity matrix  $\mathbf{S}$  has been computed, we adopt the spectral clustering algorithm proposed in [110] to construct the visual tree by taking  $\mathbf{S}$  as input, and the details of the learning procedure is summarized in Algorithm 2.

#### 4.2.3 Learning Node Predictors of a Visual Tree

Given a visual tree with a fixed structure (*i.e.*, the tree structure has been learned for the  $N$  image categories using Algorithm 1 or 2), we learn the node predictors by

solving  $|\mathbb{V}|$  independent convex problems where  $|\mathbb{V}|$  is the total number of tree nodes in the tree. Specifically, let  $B$  be the number of the child nodes for the current node  $v$  and  $\mathbb{L}(v) = \{l_1, \dots, l_L\}$  be its label set where  $L = |\mathbb{L}(v)|$ . The associated node predictor to be learned is a linear model given as  $\mathbf{w} \in \mathbb{R}^{D \times B}$  where  $D$  is the feature dimension and each column of  $\mathbf{w}$  is the parameter vector of the classifier for a child node.

The assignments between the category labels and the child nodes can be represented as a  $L$ -by- $B$  label-node association matrix  $\mathbf{A}$ , where the entry in the  $l$ th row and the  $b$ th column  $\mathbf{A}[l, b] = 1$  if the  $l$ th category label is assigned to the child node  $b$ , otherwise,  $\mathbf{A}[l, b] = 0$ . Let  $\mathcal{T} = \{(\mathbf{x}_i, y_i)\}_{i=1}^M$  be the training set where  $\mathbf{x}_i \in \mathbb{R}^D$  and  $y_i \in \mathbb{L}(v)$  for learning the predictor for the current node  $v$ . Our goal is to minimize the empirical loss for the training images in the training set and we use 0/1 loss here, given as

$$L(\mathbf{x}_i, y_i; \mathbf{w}, \mathbf{A}) = 1 - \mathbf{A}[y_i, \hat{b}], \quad (28)$$

where  $\hat{b} = \arg \max_{b \in \mathbb{C}(v)} \mathbf{w}_b^T \mathbf{x}_i$  is the child node that the training image  $\mathbf{x}_i$  will select to visit at the next level. The loss incurs if the label set of the child node with the highest prediction score does not contain the true label  $y_i$  of training image  $\mathbf{x}_i$ . Considering that each class label is only assigned to a single child node in the visual tree, we can train  $B$  OVR classifiers independently, and then stack the parameter vectors of the  $B$  classifiers to form the predictor  $\mathbf{w}$  for the current node. Specifically, to optimize the predictor  $\mathbf{w}_b$  for the child node  $b$ , we first transform the label  $y_i$  of each image in the training set to be +1 if the label set of node  $b$  contains label  $y_i$ ,



and  $-1$  otherwise. In mathematics, it is given as:

$$\theta(y_i) = \begin{cases} +1 & \text{if } y_i \in \mathbb{L}(b), \\ -1 & \text{otherwise.} \end{cases} \quad (29)$$

Using the hinge loss, we are then left with the following convex problem

$$\begin{aligned} \underset{\mathbf{w}_b, \xi}{\text{minimize}} \quad & \frac{1}{2} \|\mathbf{w}_b\|^2 + \gamma \sum_{i=1}^M \xi_i \\ \text{subject to} \quad & \theta(y_i) \mathbf{w}_b^T \mathbf{x}_i \geq 1 - \xi_i, i = 1, \dots, M \\ & \xi_i \geq 0, i = 1, \dots, M \end{aligned} \quad (30)$$

where  $\xi_i$  are non-negative slack variables which measure the degree of misclassification of data point  $\mathbf{x}_i$ .

#### 4.2.4 Label Prediction with a Visual Tree

Given the visual tree with node predictors which have been learned from the training set, we follow the same prediction procedure as described in [9] to categorize testing images. For a given test image, the prediction begins at the root node, and stops at a leaf node by visiting the best-matching node which has the largest prediction score at each depth of the visual tree. The label of the leaf node is then taken as the predicted label to the test image. We summarize the prediction process in Algorithm 3. Since only the most confident (best-matching) child node is chosen to visit at each depth of the visual tree, we refer to the prediction algorithm as *hard prediction*.

One problem of the hard prediction scheme is that only one single node is selected to visit at each depth of the visual tree, even if the score of the highest scoring node

---

**Algorithm 3.** The hard prediction scheme with visual tree

---

Input: Test image  $\mathbf{x}$ , the visual tree  $T$

- 1: Let node  $r$  be the root node
- 2: **while**  $\mathbb{C}(r) \neq \emptyset$  **do**  $\triangleright \mathbb{C}(r)$  denotes children of node  $r$
- 3:      $r \leftarrow \arg \max_{c \in \mathbb{C}(r)} \mathbf{w}_c^T \mathbf{x}$
- 4: **end while**

Output: The label of  $r$  as the prediction for image  $\mathbf{x}$

---

dose not exceed that of the second-highest scoring node by a considerable margin. It is more likely that a wrong child node would be chosen to visit in this situation, and the mistake will be propagated all the way to a leaf node and cannot be recovered. This inevitably degrade the performance of categorization accuracy.

To alleviate the performance degradation which may induced by using the hard prediction scheme, we also adopt a soft prediction scheme in this work, where the top-two child nodes (tree nodes are ordered according to their prediction scores) are simultaneously chosen to visit at each depth of the visual tree if the difference between their prediction scores dose not exceed a considerable margin. Let  $p$  and  $q$  be the highest scoring and second-highest scoring child nodes, and  $score(p)$  and  $score(q)$  be their prediction scores, respectively. The difference between  $score(p)$  and  $score(q)$  is used as the evidence for determining whether we should select only node  $p$  or both node  $p$  and  $q$  to visit at the next level. Specifically, both  $p$  and  $q$  are selected if the difference between  $score(p)$  and  $score(q)$  is below a threshold  $\epsilon$ , otherwise only node  $p$  is selected.

In the soft prediction scheme, multiple leaf nodes are reached through different paths. However, in the context of image categorization, the label prediction is desired to assign a single label or multiple labels in sorted order to a unseen image. One way

---

Algorithm 4. The soft prediction scheme with visual tree

---

Input: Test image  $\mathbf{x}$ , the visual tree  $T$ 

```

1: Initialize an empty queue  $\mathcal{Q}$ 
2: Initialize an empty vector  $\mathcal{O}$ 
3:  $EnQueue(\mathcal{Q}, r)$  ▷ Insert the root node  $r$  to queue  $\mathcal{Q}$ 
4: while  $\mathcal{Q}$  is not empty do
5:    $s \leftarrow DeQueue(\mathcal{Q})$  ▷ Delete a node from  $\mathcal{Q}$  and assign it to  $s$ 
6:   if Node  $s$  is a leaf node then
7:     Put  $s$  together with its probabilistic score into  $\mathcal{O}$ 
8:   else
9:     if  $score(p) - score(q) < \epsilon$  then ▷  $p$  and  $q$  are the top-2 scoring nodes
10:       $EnQueue(\mathcal{Q}, p)$  and  $EnQueue(\mathcal{Q}, q)$  ▷ Follow both branches
11:    else
12:       $EnQueue(\mathcal{Q}, p)$  ▷ Follow only one branch
13:    end if
14:  end if
15: end while

```

Output: The label of the node with the largest score in  $\mathcal{O}$ 


---

to rank the leaf nodes that have been visited is to sort them with respect to their prediction scores. Unfortunately, the node predictors at different levels of the visual tree are trained independently, and their responses are not directly comparable. We thus adopt the Platt scaling [121] to transform the responses to probabilistic scores for all the node predictors. Note that the leaf nodes are not associated with predictors since a leaf node contains a single class. The label of the leaf node which has the highest score is taken as the final prediction to a novel image. Algorithm 4 describes the scheme of soft prediction.

### 4.3 Experimental Setup

In this section, we describe the experimental setup for evaluating the proposed visual tree algorithms, including image set, visual feature extraction and node predictor training. The ILSVRC2010 image set<sup>3</sup>, which originates from ImageNet [33]

---

<sup>3</sup><http://www.image-net.org/challenges/LSVRC/2010>

database, is used for assessing the performance of our algorithms. The ILSVRC2010 image set contains 1.4M images for 1,000 image classes (categories). The standard training/validation/testing split is used (respectively 1.2M, 50K and 150K images).

For visual content representation, the standard BoF pipeline is used to extract an image-level signature for each image, which typically consists of local feature extraction, encoding, and pooling. Specifically, we extract dense HoG features with four different patch sizes including  $16 \times 16$ ,  $25 \times 25$ ,  $31 \times 31$  and  $46 \times 46$ , and encode them using LLC [155] with a codebook of size 8192. After encoding the local features extracted from an image, a configuration of two-level spatial partitions (*i.e.*,  $1 \times 1$  and  $2 \times 2$ ) is used to pool the LLC codes into an image-level signature. Thus the dimension of the image-level signature is 40,960.

The *stochastic gradient descent* (SGD) [16] method is adopted here to train the node predictors because it is very efficient for large-scale classifier training. The parameters for SGD were optimized based on the validation set. For the purpose of supporting large-scale computation, a computer cluster with 492 computing cores is used for experiments.

#### 4.4 Experimental Results

We report our experimental results of our visual tree algorithm in this section, and compare it with many other tree methods based on the ILSVRC2010 benchmarking data set. The competing methods include the original label tree [9], the fast and balanced label tree in [34] and the recently proposed probabilistic label tree [93]. Also, we evaluate the effectiveness of the extension of using multiple centroids for

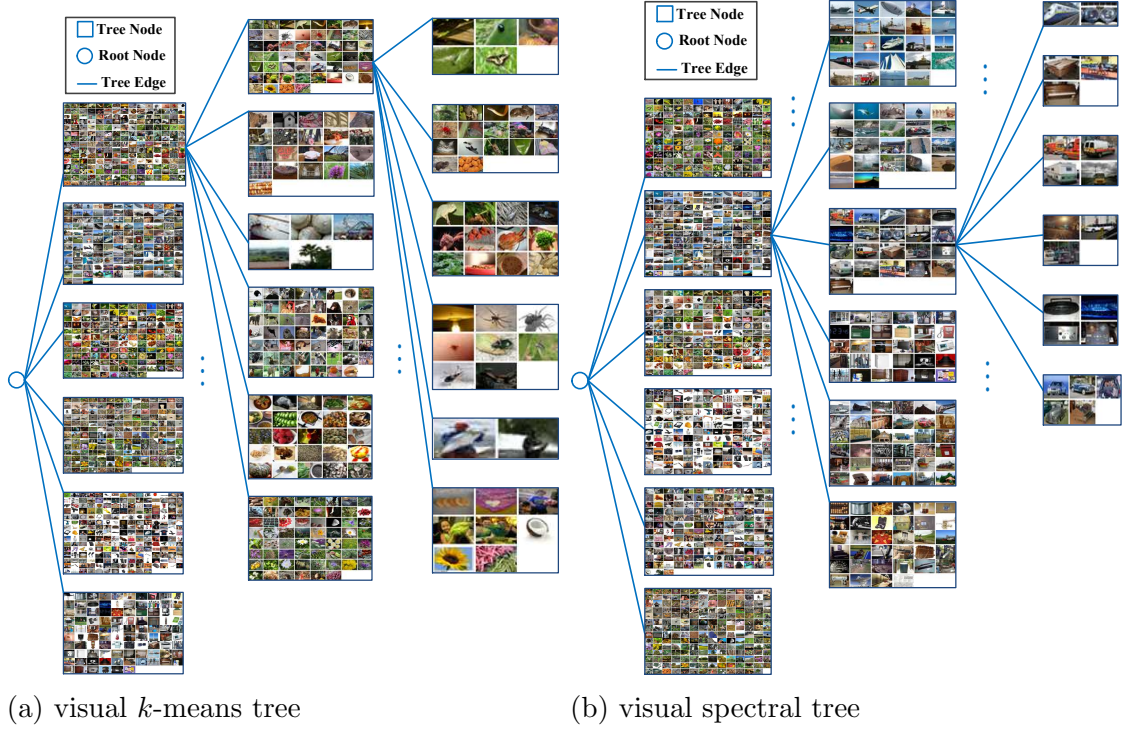


Figure 19: Visualization of the visual tree structures. The visual trees were built with branching factor  $B = 6$  and maximum depth  $H = 4$ . The leaf nodes are not shown since each of them contains a single class. (a) visual tree constructed using hierarchical  $k$ -means (c.f. Algorithm 1) ; (b) visual tree constructed using hierarchical spectral clustering (c.f. Algorithm 2).

image class visual representation and the soft prediction scheme.

#### 4.4.1 Comparison with other Tree Methods

We randomly selected 100 images per category from the ILSVRC2010 training set to compute the visual representation of each image class, and then use the proposed Algorithm 1 and 2 to learn the structures of visual trees. To learn the label tree in [9], the 100 images are further split into training and testing sets at the ratio of 3:2. The training set is used to train the OVR SVMs, and the testing set is used to obtain the confusion matrix. We repeated this process for three times, and computed an mean confusion matrix  $\mathbf{C}$  by averaging the three confusion matrices which are obtained

from the three different trials. Since the confusion matrix  $\mathbf{C}$  is not symmetric in general, we compute its symmetric version as  $\mathbf{B} = \mathbf{C} + \mathbf{C}^T$ , and learn the structure of the label tree based on  $\mathbf{B}$  using the algorithm in [9].

We control the structures of the visual tree and label tree by fixing the branching factor  $B$  and the maximum depth  $H$ . Each of the internal tree nodes is required to split into  $B$  children except if it is at depth  $H - 1$  or the cardinality  $|\mathbb{L}_v|$  of its label set is less than  $B$ . In either case, we create  $|\mathbb{L}_v|$  children for that tree node instead. We denote a tree of  $B$  branches and  $H$  maximum depth using  $T_{B,H}$ .

In Fig. 19, we visualize the tree structures of the *visual k-means tree* (c.f. Algorithm 1) and the *visual spectral tree* (c.f. Algorithm 2) with  $B = 6$  and  $H = 4$ . We use icon images to represent and illustrate image categories. An icon image is randomly selected from an image class for visually representing and illustrating it. The icon images of all the categories which are in the label set of the same tree node are used to visually illustrate that tree node. We only visualize a small part of the tree nodes due to the limitation of the screen space. One can observe that the spectral clustering algorithm can construct a more balanced tree than the  $k$ -means algorithm. The *label tree* with the same configuration is illustrated in Fig. 20.

First, we can observe that our visual tree algorithms produce similar tree structures as the label tree method, but our visual tree model directly uses the inter-category visual similarities for tree construction. It is worth noting that the inter-category visual similarities are much computationally cheaper than the confusion matrix used in the label tree method since a lot of OVR SVMs should be learned in advance to obtain the confusion matrix. Second, using the same clustering technique (*i.e.*,

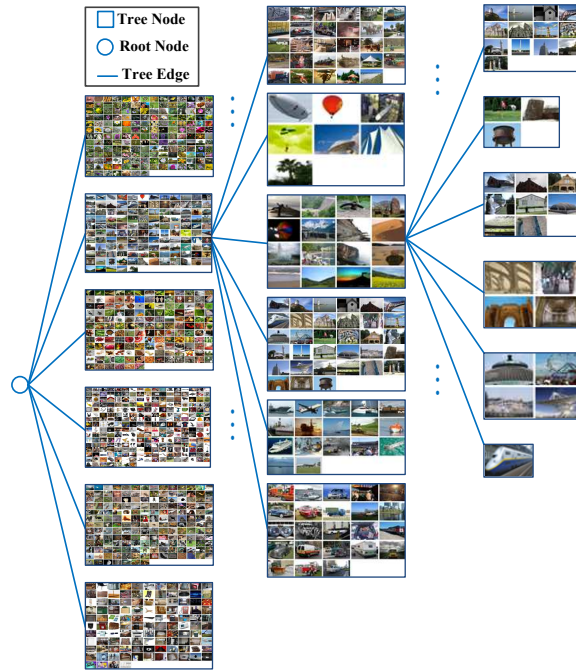


Figure 20: Visualization of the label tree structure. The label tree was built using spectral clustering based on the confusion matrix (see [9]) with the branching factor  $B = 6$  and the maximum depth  $H = 4$ . The leaf nodes are not shown since each of them contains a single category.

spectral clustering) for tree construction, our visual tree (Fig. 19(b)) exhibits an even more balanced tree structure than the label tree (Fig. 20) does. One of the reasons is that the confusion matrix computed by evaluating all the OVR SVMs based on a validation set is generally sparse. The missing values (zeros) in the confusion matrix usually cause unbalanced clustering results during the construction of a label tree. The appearances of the unknown (zeros) values in the confusion matrix not necessarily mean that the corresponding inter-category correlations do not exist. On the other hand, the visual similarity matrix  $\mathbf{S}$  computed by using the visual cues of image classes is guaranteed to be full in spite of that some of them could be very small or close to zero. The visual similarity matrix therefore often leads to more balanced clustering results for visual tree construction.

We tabulate the classification accuracy rates of our visual tree, the label tree and its two variants in Table 3 based on ILSVRC2010 image set. The results of the trees under different configurations are reported. First, the visual spectral tree learned using Algorithm 2 outperform the visual  $k$ -means tree which is constructed using Algorithm 1. Second, the performance of our proposed visual tree (visual spectral tree) is comparable to the label tree especially when the trees are of deeper depths (*e.g.* under configurations of  $T_{10,3}$  and  $T_{6,4}$ ). However, our visual tree method dose not require obtaining the cost-intensive confusion matrix, and hence avoid learning many OVR classifiers. It is worth noting that if multiple feature centroids are used for the visual representation of an image category, our visual tree is superior to the label tree in terms of the classification accuracy (c.f. Fig. 21). This has proved that the visual correlations between the image categories (*i.e.*, inter-category visual similarities) can be used as an effective surrogate to learn a tree structure for organizing a large number of image categories hierarchically. Third, our implementation of the label tree achieves state-of-the-art results on the ILSVRC2010 image set, *e.g.*, it even outperforms the probabilistic label tree in [93] under the  $T_{32,2}$  configuration.

One of the advantages for organizing a large number of image categories hierarchically in a tree structure is that it can significantly reduce the computational cost in large-scale image categorization. For large-scale image categorization, an ideally-balanced tree leads to logarithmic complexity in terms of the number of image categories in testing time, and it is achieved by testing against a limited number of possible node predictors along the traversal path. We use the average number of dot products [34] that is needed to make a label prediction for a unseen image as the test-



Table 3: Classification accuracy (%) comparison between our visual trees and the label tree and its variants on the ILSVRC2010 image set. *Cues for node splitting*: information used to learn the tree structures in different methods.  $T_{B,H}$  denotes a tree having at most  $B$  children per node and  $H$  depths (root node has depth 0).

Model	Cues for node splitting	$T_{32,2}$	$T_{10,3}$	$T_{6,4}$
Flat (1,000 OVR SVMs)		36.36		
Visual $k$ -means tree (Alg. 1)	visual feature	15.23	12.30	12.23
Visual spectral tree (Alg. 2)	visual feature	18.94	13.77	13.21
Label tree (our implementation)	output of OVR classifiers	<b>22.34</b>	14.81	13.11
Label tree implemented in [34]	output of OVR classifiers	8.33	5.99	5.88
Results in [34]	output of node classifiers	11.90	8.92	5.62
Results in [93]	output of node classifiers	21.38	<b>20.54</b>	<b>17.02</b>

ing efficiency metric. In the flat categorization approach, 1,000 OVR SVMs are used, and a test image has to be tested against all the 1,000 OVR classifiers for inference, which amounts to 1,000 dot products assuming that the SVMs are linear models. Let  $n$  and  $m$  be the average numbers of dot products used in the flat method and the tree models for image categorization, respectively. We define the testing speedup as  $S_{test} = \frac{n}{m}$ . In a similar spirit, we define the training speedup, and denote it as  $S_{train}$ .

In Table 4, we compare the training and testing efficiency of different tree methods. One can observe that the visual tree built by using hierarchical spectral clustering (Algorithm 2) is more efficient than the visual tree built by using hierarchical  $k$ -means (Algorithm 1) because spectral clustering usually produces a more balanced visual tree by penalizing unbalanced partitions during the node splitting. Also, our visual tree (*e.g.*, the visual spectral tree) outperform the original label tree and the other two variant label tree models in [34, 93] in terms of computation speedup, and achieves the most speedup in both training and testing. This verifies our previous observation that the visual tree has a more balanced tree structure than the label tree, which

Table 4: Computational efficiency comparison of different tree methods.  $S_{train}$ : testing speedup compared to the flat model.  $S_{test}$ : testing speedup compared to the flat model.

Model	$T_{32,2}$		$T_{10,3}$		$T_{6,4}$	
	$S_{train}$	$S_{test}$	$S_{train}$	$S_{test}$	$S_{train}$	$S_{test}$
Visual $k$ -means tree (Alg. 1)	13.83	14.76	28.38	26.86	36.56	32.87
Visual spectral tree (Alg. 2)	<b>15.17</b>	<b>15.00</b>	<b>32.32</b>	<b>32.62</b>	<b>42.86</b>	<b>42.44</b>
Label tree (our implementation)	12.72	12.19	28.28	27.21	36.02	35.53
Label tree implemented in [34]	-	10.30	-	15.20	-	9.32
Results in [34]	-	10.30	-	18.20	-	31.30
Results in [93]	-	10.42	-	17.85	-	31.25

leads to better training and testing efficiencies.

#### 4.4.2 Evaluation of using Multiple Centroids per Class

In this section, we empirically evaluate how the extension of using multiple centroids per class for visual representation (c.f. Section 4.2.1) affect the performance of the proposed visual tree model. Increasing the number of centroids ( $K$ ) per class allows for more flexible visual representation, and thus characterizes the visual appearance of an image class with more details. However, it might be less robust to noise. For example, if  $K$  equals to the number of relevant images in an image category (*i.e.*, the feature vector of each image is treated as a feature centroid), every image instance would then play a major role in estimating the inter-category visual similarity which is defined in Eq. (27).

We obtain the  $K$  centroids per class by using the  $k$ -means algorithm in the Euclidean space. Fig. 21 shows the classification results of different visual trees with different configurations, including various branching factor  $B$  and maximum depth  $H$  values. The classification accuracy rates are demonstrated with the number of cen-

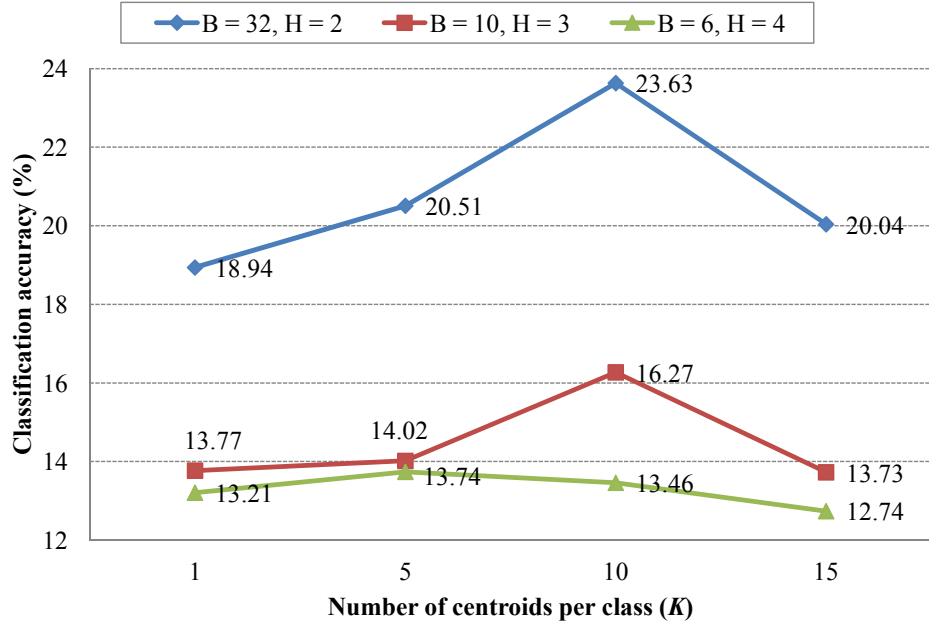


Figure 21: The classification performance of the visual tree model using different number of centroids per class.  $B$ : branching factor;  $H$ : the maximum depth of a tree.

troids ( $K$ ) ranging from 1 to 15. We observe that the accuracy rates increase first and then decrease. The reason is twofold: 1) using more feature prototypes initially increases the capability of the mean feature model for describing the visual properties of an image category which leads to better classification performance of the visual tree; 2) however, if the number of feature prototypes (centroids) is too many, the mean feature model is more sensitive to the noisy or insignificant image instances which results in classification performance sliding down. In Fig. 22, we show the testing speedup of different visual trees across different  $K$ 's. It is seen that the speedup of the proposed visual tree model is very robust to the number of centroids. For the training speedup of our visual tree model, we observe very little changes by adopting different numbers of feature centroids, and thus we do not report the detailed numbers here to save space.

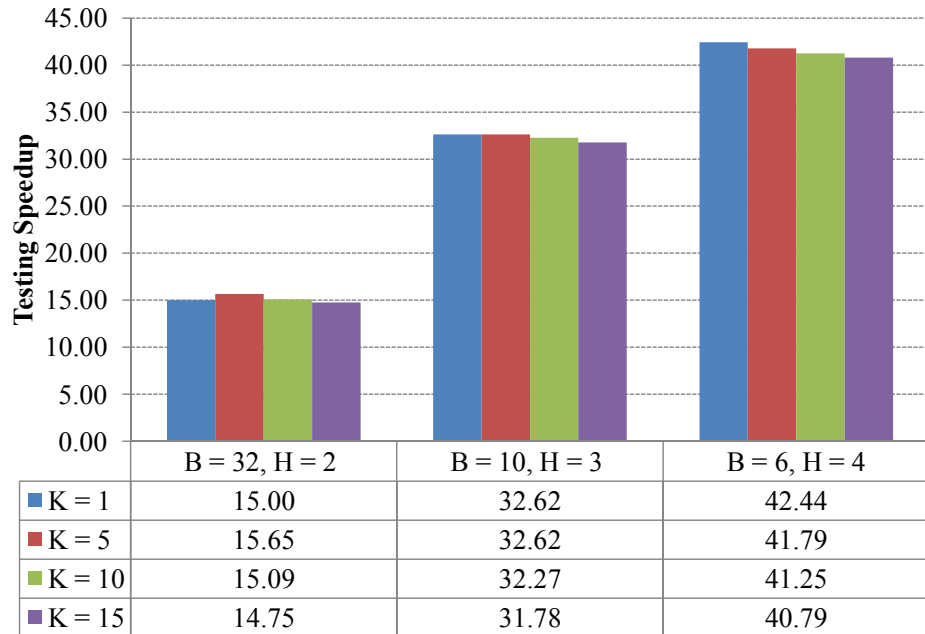


Figure 22: The testing speedup performance of the visual tree model using different number of centroids per class.  $K$ : the number of centroids per class;  $B$ : branching factor;  $H$ : the maximum depth of a tree.

#### 4.4.3 Evaluation on Soft Prediction

While soft prediction tends to boost the accuracy rates by exploring more branches in a tree model, it incurs more dot products to yield a prediction for a test image. As the threshold  $\epsilon$  in Algorithm 4 varies, the average number of dot products which are needed to categorize a test image also changes. We plot the classification accuracy rates against the average numbers of dot products of different tree models in Fig. 23. The scheme of soft prediction is more critical for the trees of deeper depths because a wrong child node is more likely to be chosen at a high-level node. We thus only evaluate the soft prediction using the trees with the configurations of  $T_{10,3}$  and  $T_{6,4}$ . One can observe that the soft prediction scheme can boost the categorization performance of both visual tree and label tree models by exploring more branches.

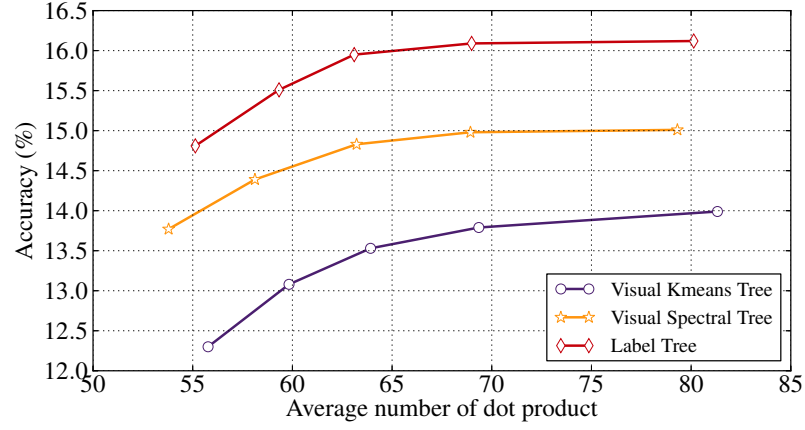
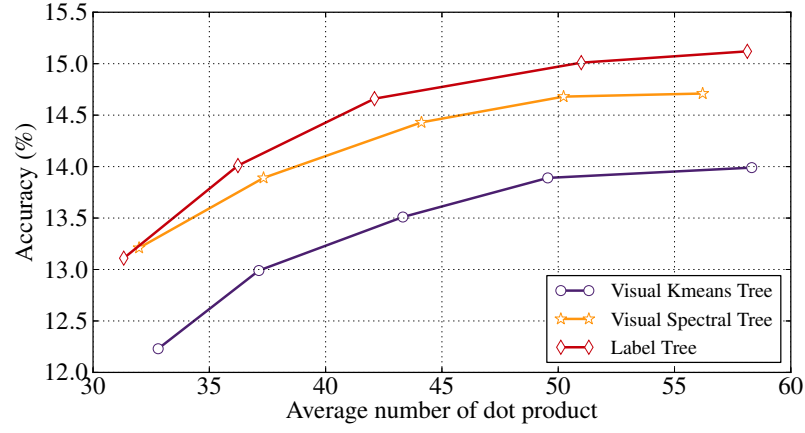
(a)  $T_{10,3}$ (b)  $T_{6,4}$ 

Figure 23: Performance evaluation of the soft prediction scheme. The categorization accuracy is plotted over different average numbers of dot products.

## 4.5 Summary

Multi-class image categorization becomes challenging when the number of image categories becomes very large. A flat approach where a testing sample has to test against every possible image category (*i.e.*, every possible classifier) is computationally infeasible. In this paper, to alleviate this issue, a visual tree has been constructed for organizing a large number of image categories hierarchically according to their inter-category visual correlations. Without resorting to a cost-intensive confusion

matrix, our visual tree method significantly reduces the computational cost for the tree construction compared to the traditional label tree approach. As well as presenting the visual tree model, we have developed an effective class mean feature method and its extension to characterize the visual appearance of an image category. Also, a soft prediction scheme has been designed to further boost the classification accuracy of tree methods by exploring more possible branches. Our experimental results have demonstrated that our visual tree can achieve very competitive categorization accuracy rates and better computational efficiency results than other tree methods.

## CHAPTER 5: DISCRIMINATIVE DICTIONARY LEARNING

### 5.1 Introduction

Large-scale visual recognition has been a tremendously challenging problem in the field of computer vision for decades. One of the paradigms is to automatically categorize objects or images into hundreds or even thousands of different classes. It has recently received significant attention [34, 32, 9, 90, 117, 116], partly due to the increasing availability of big image data. For example, ImageNet [33], a large-scale labeled image database, has collected 14M images for 22K visual categories. There are two important criteria for assessing the performance of a large-scale visual recognition system: (1) recognition accuracy; and (2) computational efficiency. The accuracy depends largely on the discrimination power of visual content representations as well as the effectiveness of classifier training techniques, and the efficiency relies greatly on the methods for category organization (*e.g.*, flat or hierarchical).

The *bag-of-visual-words* (BoW) model, one of the most successful models for visual content representation, has been widely adopted in many computer vision tasks, such as object recognition [62, 160], image classification [85, 79] and segmentation [176]. The standard recognition pipeline of BoW, which consists of local feature extraction, encoding, pooling and classifier training, harnesses both the discrimination power of some well-engineered local features (*e.g.*, SIFT [98], HoG [29], *etc.*) and the general-

ization ability of large margin classifiers (*e.g.*, SVM). It has accomplished top results in visual recognition, from medium-sized image sets [165, 184] to large-scale ones [116, 130]. Apart from many advanced methods for feature encoding, *e.g.*, sparse coding [165], local coordinate coding (LCC) [155], super-vector coding [184], Fisher vector [117], *etc.*, a dictionary (codebook) of strong discrimination power is usually demanded to achieve better classification results. However, the dictionaries learned through unsupervised learning [1, 39, 83] are usually lack of strong discrimination.

In [103, 174, 167, 102, 104, 73, 168], it has been shown that training more discriminative dictionaries via supervised learning usually leads to better recognition performance. A typical method is to integrate the processes of dictionary learning and classifier training into a single objective function by adding a discriminative term according to various criteria, such as the logistic loss function with residual errors [103], the soft-max cost function with classification loss [167], the linear classification error [174] and the Fisher discrimination criterion [71, 168]. More recently, researchers have proposed to learn individual dictionaries for different categories, and to enhance their discrimination by incorporating the reconstruction errors with the soft-max cost function [102], promoting the incoherence among multiple dictionaries [122] or exploiting the classification errors through a boosting procedure [177].

In large-scale visual recognition applications, the number of categories could be huge (*e.g.*, hundreds or even thousands). Therefore, it is computationally infeasible to integrate the dictionary learning and classifier training into one single optimization framework. More importantly, some image categories have stronger inter-category visual correlations than others. For example, the five image categories in Fig. 24, *whip-*



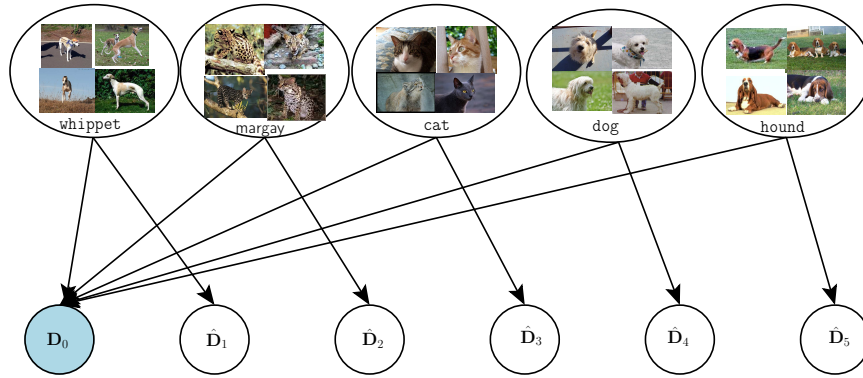


Figure 24: Inter-related dictionaries for a group of visually correlated categories. A common dictionary  $\mathbf{D}_0$  is used to characterize the commonly shared visual patterns, and five category-specific dictionaries  $\{\hat{\mathbf{D}}_i\}_{i=1}^5$  are devised to depict the class-specific visual patterns.

*pet*, *margay*, *cat*, *dog*, and *hound*, which are selected from the ImageNet [33] database, are of strong visual correlations since they are highly visually similar. A number of common visual features shared by the visually correlated categories contribute nothing to distinguishing them. However, in most existing dictionary learning methods, the commonly shared visual atoms are treated equally as the category-specific ones which are more useful for the recognition. For a group of visually similar categories, what is desired is a new dictionary learning algorithm which can explicitly separate the shared visual words from the class-specific ones, and jointly learn the inter-related dictionaries to enhance their discrimination.

Considering a large number of image categories, the visual features shared by all of them might be limited. However, visually similar categories usually share a considerable amount of features. It is natural to cluster these categories into a number of disjoint groups so that each group contains a reasonable number of visually correlated categories whose dictionaries indeed share some common visual atoms. In other words, image category clustering provides a way to support a newly proposed

joint dictionary learning (JDL) algorithm (Section 5.3) by guaranteeing that the categories in the same group have strong visual correlations. In addition, image category clustering makes the JDL algorithm to be computationally affordable in large-scale visual recognition applications by allowing one to perform JDL for different groups sequentially or in parallel. Finally, it is shown in Section 5.5.3.1 that the *unsupervised dictionary learning* (UDL) method [83] even learns better dictionaries for classification with the help of image category clustering.

The idea of clustering image categories into a set of disjoint groups is related to a myriad of works on learning image category hierarchies for reducing the computational complexity of image classification [6, 65, 107, 137, 9, 34]. A remarkable example is the label tree method [9] which learns a tree structure by recursively clustering the categories of interest into disjoint sets based on a confusion matrix. The adoption of the confusion matrix is based on the fact that putting the classes which are easily confused by classifiers into the same set (*i.e.*, the same tree node) makes the classifiers associated with the tree node to be easily learnable [9]. However, in this work, the main purpose of image category clustering is to determine which categories possess strong visual correlations, and their dictionaries should be learned together to enhance the discrimination.

In this chapter, we investigate the learning of discriminative dictionaries for large-scale visual recognition applications. First, for a group of visually correlated categories, a *joint dictionary learning* (JDL) algorithm is developed to make use of the inter-category visual correlations for learning more discriminative dictionaries. Specifically, JDL simultaneously learns one common dictionary and multiple category-

specific dictionaries to explicitly separate the commonly shared visual atoms from the category-specific ones. Considering again the example illustrated in Fig. 24, a common dictionary  $\mathbf{D}_0$  is devised to contain the visual atoms shared by all the five categories, and five dictionaries  $\{\hat{\mathbf{D}}_i\}_{i=1}^5$  are used to hold the category-specific visual atoms, respectively. To enhance the discrimination of the dictionaries, a discrimination promotion term is added according to the Fisher discrimination criterion [38] which directly operates over the sparse coefficients.

Second, we empirically study a number of different approaches to image category clustering and how they affect the performance of the proposed JDL algorithm. They include the label tree [9] method and a newly proposed *visual tree* approach. The purpose of image category clustering is twofold. (1) It determines the groups of visually correlated classes to ensure that the dictionaries for the categories in the same group share some common visual atoms. Thus, the proposed JDL algorithm can be used to learn more discriminative dictionaries by separating the common visual atoms from the category-specific ones. (2) It makes the JDL algorithm to be computationally tractable in large-scale visual recognition applications as JDL can be applied to different groups in sequence or parallel.

Third, three schemes are developed for image content representation, classifier training and image classification: (1) local classification scheme; (2) global classification scheme; and (3) hierarchical classification scheme. The local classification scheme is applicable when the labels of test images are defined as the classes of a single group. Particularly, after the dictionaries have been learned by JDL for the five classes in Fig. 24, the local classification scheme can be used if a test image is

only required to be classified into one of the five categories. Nevertheless, a large-scale visual recognition system should be capable of distinguishing hundreds or thousands of classes from different groups. The global classification scheme is thus designed to categorize a test image into any class from any group. Finally, by clustering the categories into a number of disjoint groups, a tree structure of depth two is actually constructed where the root, group and category nodes are of depth zero, one and two, respectively. We design a hierarchical classification scheme to make use of the tree structure for reducing the computational complexity of image classification, where the group classifiers at depth *one* are used to identify the most-likely group for a test image, and the category classifiers at depth *two* are used to predict the best-matching category in the chosen group. More importantly, the group and category classifiers are trained in two disjoint feature spaces by taking advantage of the structure of the dictionaries learned by JDL.

Experiments have been conducted on popular visual recognition benchmarks, including the 17-class Oxford flower image set and the ILSVRC2010 data set<sup>4</sup> which originates from the ImageNet [33] database. Our experimental results demonstrate that the proposed JDL algorithm is superior to many previous unsupervised and supervised methods on learning discriminative dictionaries for the task of image categorization.

The rest of this chapter is organized as follows. The visual tree method for image category clustering is described in Section 5.2. In Section 5.3, we present the joint dictionary learning algorithm, including formulation and optimization. Three classi-

---

<sup>4</sup><http://www.image-net.org/challenges/LSVRC/2010>

fication schemes are described in Section 5.4. The experimental setup and results are given in Section 5.5. We conclude in Section 5.6.

## 5.2 Image Category Clustering for Joint Dictionary Learning

When a group of categories have strong inter-category visual correlations, their dictionaries which share some common visual atoms should be trained jointly to enhance the discrimination. In this section, a *visual tree* method is proposed to generate such groups of visually correlated categories by clustering a large number of categories into disjoint groups according to their inter-category visual correlations.

### 5.2.1 Visual Category Representation

To characterize the inter-category visual correlations, we first estimate the visual representation of an image category based on its relevant images. Let  $\mathcal{I}_i$  be a collection of relevant images for the  $i$ th category. We compute the average visual feature as the overall visual representation for it. First, the content of image  $I_j \in \mathcal{I}_i$  is represented using the BoW model. Specifically, we encode the local SIFT features extracted from it over a dictionary using sparse coding, and then pool the sparse codes with max-pooling to form an image-level representation  $\mathbf{h}_j$  for it. In implementation, a dictionary of size 4,096 was used, and a two-level spatial pyramid partition ( $1 \times 1$ ,  $2 \times 2$ ) was employed to incorporate weak spatial information. Second, the visual representation  $\mathbf{H}_i$  of the  $i$ th category is defined as the average feature based on all the relevant images:

$$\mathbf{H}_i = \frac{1}{|\mathcal{I}_i|} \sum_{I_j \in \mathcal{I}_i} \mathbf{h}_j, \quad (31)$$

Table 5: A number of category groups identified by  $k$ -means based on the visual representations of the categories in ILSVRC2010 data set.

bullet train; CD player; grand piano; grille; odometer; subway train	ballpoint; bathtub; iPod; lamp; lampshade; paper towel; washbasin	airship; envelope; fountain pen; parachute; radio telescope; rule; stupa	breakwater; speedboat; stadium; dune; lakeside; promontory; sandbar; seashore
howler monkey; spider monkey; ilang-ilang; vanda; fig; paper mulberry; coral tree; Arabian coffee; holly	cassette player; hot plate; photocopier; Primus stove; printer; radio; scanner; shredder; swivel chair	monarch; nigella; cornflower; cosmos; dahlia; coneflower; gazania; African daisy; sunflower	hourglass; ladle; lighter; nail polish; needle; nipple; pencil; rubber eraser; saltshaker; toothbrush
banjo; bassoon; bow; Chinese lantern; cornet; dial telephone; drum; euphonium; flute; football helmet; sax; shovel; sword; trombone	sloth bear; polecat; orangutan; gorilla; chimpanzee; gibbon; siamang; guenon; langur; colobus; marmoset; titi; squirrel monkey; lesser panda	birdhouse; butcher shop; buttress; carousel; church; confectionery; dome; fountain; jinrikisha; padlock; picket fence; roller coaster; shoe shop; toyshop	isopod; honeycomb; cress; elderberry; lunar crater; juniper berry; ginkgo; wattle; mistflower; witch elm; silver maple; Oregon maple; sycamore; box elder
pheasant; spiny lobster; leopard; cheetah; wood rabbit; laurel; dusty miller; sorrel tree; alder; fringe tree; European ash; mountain ash; ailanthus; China tree; Japanese maple; pepper tree	tree frog; American chameleon; green lizard; African chameleon; green snake; green mamba; jellyfish; leaf beetle; weevil; fly; grasshopper; cricket; cicada; leafhopper; mayfly; lacewing	black grouse; snail; Chihuahua; English setter; Brittany spaniel; Saint Bernard; griffon; corgi; white wolf; Arctic fox; Egyptian cat; ice bear; weasel; mink; black-footed ferret; macaque; giant panda	can opener; circular saw; crash helmet; face powder; frying pan; hard disc; harmonica; iron; lens cap; Loafer; mouse; plane; pocketknife; projector; stethoscope; straight razor; trackball; waffle iron

where  $|\mathcal{I}_i|$  is the number of images in  $\mathcal{I}_i$ . Finally, we normalize the  $l_2$  norm of  $\mathbf{H}_i$  to be 1.

### 5.2.2 Image Category Clustering

After the visual representations  $\{\mathbf{H}_i\}_{i=1}^M$  for  $M$  categories are computed, theoretically any clustering methods can be used to cluster them into disjoint groups based on the representations. For instance, one could simply use the  $k$ -means algorithm to group the categories by taking  $\{\mathbf{H}_i\}_{i=1}^M$  as input. Table 5 presents a number of category groups which are identified by the method following this idea based on the ILSVRC2010 data set.

Another approach is to first compute the visual similarities between the categories, and then cluster them into groups according to the similarity values. Specifically, we

Table 6: A number of category groups identified by AP clustering based on the visual similarities between the image categories in ILSVRC2010 data set.

black and gold garden spider; garden spider; mantis; dragonfly; damselfly; lycaenid	ambulance; cab; limousine; police van; recreational vehicle; school bus; trolleybus	cocktail shaker; hair spray; lipstick; lotion; metronome; pendulum clock; shaver	espresso maker; flash; hand calculator; handheld computer; hipflask; loudspeaker; tumble-dryer
male orchid; marsh orchid; fragrant orchid; lizard orchid; gentian; kowhai; goat willow; coral fungus	barbell; binoculars; dumbbell; hand blower; joystick; knee pad; maraca; microphone; pencil sharpener	desktop computer; electric range; hot plate; ice maker; monitor; photocopier; printer; radio; scanner	bow tie; brassiere; eyepatch; gasmask; hand glass; hard hat; oxygen mask; seat belt; violin; wig
computer keyboard; desk; digital clock; dining table; dishwasher; grand piano; laptop; pool table; sewing machine; table-tennis table; trundle bed	abacus; balance beam; beaker; horizontal bar; ice skate; marimba; parallel bars; pew; rotisserie; subway train; volleyball	vine snake; magnolia; calla lily; butterfly orchid; aerides; cattleya; cymbid; dendrobium; odontoglossum; oncidium; phaius; moth orchid	bell cote; bridge; but-tress; castle; church; dome; fountain; mosque; picket fence; roller coaster; silo; skyscraper; triumphal arch
Rhodesian ridgeback; greyhound; Scottish deerhound; Australian terrier; vizsla; English setter; dalmatian; basenji; white wolf; dingo; Indian elephant; African elephant; birdhouse; rugby ball; soccer ball	baobab; kapok; red beech; New Zealand beech; live oak; cork oak; yellow birch; American white birch; downy birch; iron tree; mangrove; Brazilian rosewood; cork tree; weeping willow; teak	tiger; orangutan; gorilla; siamang; proboscis monkey; howler monkey; spider monkey; Madagascar cat; indri; lesser panda; chainlink fence; spider web; lunar crater; vanda; cacao; coral tree; holly	apiary; barrow; brass; croquet ball; doormat; greenhouse; jigsaw puzzle; maze; mountain bike; ox-cart; park bench; plow; rake; rug; shopping cart; sundial; window screen; bonsai

define the visual affinity value  $s(i, j)$  between the  $i$ th and  $j$ th classes as:

$$s(i, j) = \exp \left( -\frac{d(\mathbf{H}_i, \mathbf{H}_j)}{\sigma} \right), \quad (32)$$

where  $\sigma$  is the bandwidth which is automatically determined by the self-tuning technique proposed in [172], and  $d(\cdot, \cdot)$  is the Euclidean distance operator. The inter-category affinity values of  $M$  categories can be represented as an  $M$ -by- $M$  matrix  $\mathbf{S}$ , where  $S_{i,j} = s(i, j)$ . The affinity propagation (AP) [57] clustering algorithm is employed to partition the categories into a set of disjoint groups by taking  $\mathbf{S}$  as input because of the effectiveness of AP clustering in many applications. Assuming that all the categories have equal chances to be the exemplars, their preferences are set to be a common value which is chosen as the median of all the similarity values. In Table 6, we show several category groups which are generated by using AP clustering with affinity matrix  $\mathbf{S}$  as input for the ILSVRC2010 set.

It is seen in Table 5 and 6 that category clustering aims to assign a small number of visually similar categories into the same group, so that their inter-related dictionaries should share some common visual atoms. Given such a group of visually correlated categories, the proposed joint dictionary learning (JDL) algorithm (Section 5.3) can be utilized to learn more discriminative dictionaries by explicitly separating the common visual atoms from the category-specific ones.

### 5.2.3 Relation to Label Tree [9]

It is worth noting that the label tree method [9] can also be used as an effective approach to category clustering. Specifically, to obtain the confusion matrix  $\mathbf{C} \in \mathbf{R}^{M \times M}$  ( $M$  is the number of categories), which is needed in the label tree method, we train  $M$  one-vs-rest (OVR) binary SVMs, and then evaluate them on a validation set. The categories are partitioned into a number of disjoint groups using spectral clustering [110] with  $\mathbf{B}$  as the affinity matrix, where  $\mathbf{B} = \frac{\mathbf{C} + \mathbf{C}^T}{2}$ .

One can observe that the label tree method tends to assign the classes which are easily confused by the OVR SVMs into the same group. However, the visual tree method is designed to put the categories which have strong inter-category visual correlations into the same group, so that the JDL algorithm can be applied to learn more discriminative dictionaries. In Section 5.5.3.1, we not only compare the performance of the visual tree with the label tree in the task of category clustering, but also quantitatively evaluate how they contribute to the performance of the proposed JDL algorithm on image classification.



### 5.3 Joint Dictionary Learning

After a large number of visual categories are partitioned into a set of disjoint groups, we present a joint dictionary learning (JDL) algorithm in this section. It can simultaneously learn one common dictionary and multiple category-specific dictionaries for the visually correlated categories in the same group. Obviously, the discriminative dictionaries of different groups can be learned independently by performing the JDL algorithm sequentially or in parallel.

#### 5.3.1 Formulation of JDL

Given a group of  $C$  visually correlated categories, let  $\mathbf{X}_i \in \mathbb{R}^{d \times N_i}$ ,  $i = 1, \dots, C$ , be a collection of training points for the  $i$ th class, and  $\mathbf{D}_i \in \mathbb{R}^{d \times K_i}$  is its visual dictionary, where  $d$  is the dimension of a training sample,  $N_i$  is the number of training samples for the  $i$ th class, and  $K_i$  is the number of visual atoms in dictionary  $\mathbf{D}_i$ . The dictionaries  $\{\mathbf{D}_i\}_{i=1}^C$  for the visually correlated categories in the same group share some common visual words, so each of these dictionaries can be partitioned into two parts: (1) a collection of  $K_0$  visual words, denoted as  $\mathbf{D}_0 \in \mathbb{R}^{d \times K_0}$ , which are used to describe the common visual properties for all the visually similar classes in the same group; and (2) a set of  $K_i - K_0$  visual words, denoted as  $\hat{\mathbf{D}}_i \in \mathbb{R}^{d \times (K_i - K_0)}$ , which are responsible for describing the class-specific visual properties of the  $i$ th category. Following the notation of concatenating two column vectors as:  $[\mathbf{d}_1; \mathbf{d}_2] \triangleq \begin{bmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \end{bmatrix}$  and  $[\mathbf{d}_1, \mathbf{d}_2] \triangleq [\mathbf{d}_1 \ \mathbf{d}_2]$ , each dictionary  $\mathbf{D}_i$  can be mathematically denoted as  $\mathbf{D}_i = [\mathbf{D}_0, \hat{\mathbf{D}}_i]$ . We

formulate the joint dictionary learning problem for  $C$  visually similar classes as:

$$\min_{\{\mathbf{D}_0, \hat{\mathbf{D}}_i, \mathbf{A}_i\}_{i=1}^C} \sum_{i=1}^C \left\{ \|\mathbf{X}_i - [\mathbf{D}_0, \hat{\mathbf{D}}_i] \mathbf{A}_i\|_F^2 + \lambda \|\mathbf{A}_i\|_1 \right\} + \eta \Psi(\mathbf{A}_1, \dots, \mathbf{A}_C), \quad (33)$$

where  $\mathbf{A}_i = [\mathbf{a}_{i1}, \dots, \mathbf{a}_{iN_i}] \in \mathbb{R}^{K_i \times N_i}$  is the sparse coefficient matrix of  $\mathbf{X}_i$  over the  $i$ th visual dictionary  $\mathbf{D}_i$ ,  $\lambda$  is a scalar parameter which relates to the sparsity of the coefficients,  $\Psi(\mathbf{A}_1, \dots, \mathbf{A}_C)$  is a term acting on the sparse coefficient matrices to promote the discrimination of the dictionaries, and  $\eta \geq 0$  is a parameter which controls the trade-off between reconstruction and discrimination.

### 5.3.2 Discrimination Promotion Term

The discrimination promotion term  $\Psi(\mathbf{A}_1, \dots, \mathbf{A}_C)$  is designed to not only couple the processes of learning multiple inter-related dictionaries, but also promote the discrimination of the sparse coefficients as much as possible. According to Fisher linear discriminative analysis (LDA) [38], one can obtain more discriminative coefficients by maximizing the separation of the sparse coefficients of different categories in the same group. It is usually achieved by minimizing the within-class scatter matrix and maximizing the inter-class scatter matrix at the same time.

In our settings, the within-class scatter matrix is defined as:

$$S_W = \sum_{j=1}^C \sum_{\mathbf{a}_i \in \mathbf{A}_j} (\mathbf{a}_i - \mu_j)(\mathbf{a}_i - \mu_j)^T, \quad (34)$$

where  $\mu_j$  is the mean column vector of matrix  $\mathbf{A}_j$ ,  $\mathbf{a}_i$  is a column vector in  $\mathbf{A}_j$ , and  $T$  denotes the matrix transposition. Considering the structure of the dictionaries for

a group of visually correlated classes, the sparse coefficient matrix  $\mathbf{A}_j$  for the  $j$ th class is concatenated by two sub-matrices  $\mathbf{A}_j^0$  and  $\hat{\mathbf{A}}_j$  in the form of  $[\mathbf{A}_j^0; \hat{\mathbf{A}}_j]$ , where  $\mathbf{A}_j^0$  contains the sparse codes over the common dictionary  $\mathbf{D}_0$ , and  $\hat{\mathbf{A}}_j$  is the matrix holding the corresponding sparse coefficients over the class-specific visual dictionary  $\hat{\mathbf{D}}_j$ . The inter-class scatter matrix is defined as:

$$S_B = \sum_{j=1}^C N_j (\mu_j^0 - \mu^0)(\mu_j^0 - \mu^0)^T. \quad (35)$$

where  $\mu_j^0$  and  $\mu^0$  are the mean column vectors of  $\mathbf{A}_j^0$  and  $\mathbf{A}^0 = [\mathbf{A}_1^0, \dots, \mathbf{A}_C^0]$ , respectively. The discrimination promotion term is therefore defined as:

$$\Psi(\mathbf{A}_1, \dots, \mathbf{A}_C) = \text{tr}(S_W) - \text{tr}(S_B), \quad (36)$$

where  $\text{tr}(\cdot)$  is the matrix trace operator. Plugging (36) into (33), we have the optimization function for the JDL model, given as:

$$\begin{aligned} \min_{\{\mathbf{D}_0, \hat{\mathbf{D}}_i, \mathbf{A}_i\}} \sum_{i=1}^C \left\{ \|\mathbf{X}_i - [\mathbf{D}_0, \hat{\mathbf{D}}_i][\mathbf{A}_i^0; \hat{\mathbf{A}}_i]\|_F^2 + \lambda \|\mathbf{A}_i\|_1 \right\} \\ + \eta (\text{tr}(S_W) - \text{tr}(S_B)). \end{aligned} \quad (37)$$

The discrimination promotion term has several attractive properties. First, it directly operates on the sparse coefficients rather than on the classifiers [73, 102, 103, 174, 167], dictionaries [122], or both the reconstruction term and the sparse coefficients [168]. The discrimination promotion term can thus make the optimization of JDL to be more tractable. Second, the discrimination of sparse coefficients is closely related to the discrimination power of classifiers because sparse coefficients are usually used as the input features of the classifiers. By learning more discriminative coeffi-

cients, the discrimination of the learned dictionaries is essentially enhanced because the sparse codes and the visual atoms are updated in an iterative way. Finally, the discrimination promotion term  $\Psi(\cdot)$  is differentiable. An iterative scheme is used to solve the JDL problem (37) by optimizing with respect to  $\{\mathbf{D}_i\}_{i=1}^C$  and  $\{\mathbf{A}_i\}_{i=1}^C$  while holding the others fixed.

### 5.3.3 Optimization of JDL

The optimization procedure of the JDL problem (37) iteratively goes through two sub-procedures: (1) computing the sparse coefficients by fixing the dictionaries, and (2) updating the dictionaries by fixing the sparse coefficients.

Considering that the dictionaries  $\{\mathbf{D}_i\}_{i=1}^C$  are fixed, (37) essentially reduces to a sparse coding problem. However, the traditional sparse coding (*e.g.*,  $l_1$  norm optimization) only involves one single sample each time. The coefficient vector  $\mathbf{a}_i$  of a sample  $\mathbf{x}_i$  is computed without considering other samples' sparse coefficients. In JDL, when we compute the sparse codes of  $\mathbf{x}_i$ , the coefficients of other samples from the categories in the same group must be considered simultaneously. Therefore, we compute the sparse coefficients class by class. That is, the sparse codes of the samples from the  $i$ th class are simultaneously updated by fixing the coefficients of those from the other classes in the same group. Mathematically, we update  $\mathbf{A}_i$  by fixing  $\mathbf{A}_j$ ,  $j \neq i$ , and the objective function is given as:

$$F(\mathbf{A}_i) = \|\mathbf{X}_i - [\mathbf{D}_0, \hat{\mathbf{D}}_i]\mathbf{A}_i\|_F^2 + \lambda\|\mathbf{A}_i\|_1 + \eta\psi(\mathbf{A}_i), \quad (38)$$

where  $\psi(\mathbf{A}_i)$  is the discrimination promotion term derived from  $\Psi(\mathbf{A}_1, \dots, \mathbf{A}_C)$  when

all the other coefficient matrices are fixed, given as:

$$\psi(\mathbf{A}_i) = \|\mathbf{A}_i - \mathbf{M}_i\|_F^2 - \sum_{j=1}^C \|\mathbf{M}_j^0 - \mathbf{M}_{(j)}^0\|_F^2. \quad (39)$$

The matrix  $\mathbf{M}_i \in \mathbb{R}^{K_i \times N_i}$  consists of  $N_i$  copies of the mean vector  $\mu_i$  as its columns. And the matrices  $\mathbf{M}_j^0 \in \mathbb{R}^{K_0 \times N_j}$  and  $\mathbf{M}_{(j)}^0 \in \mathbb{R}^{K_0 \times N_j}$  are produced by stacking  $N_j$  copies of  $\mu_j^0$  and  $\mu^0$  as their column vectors, respectively. We drop the subscript  $j$  of  $\mathbf{M}_{(j)}^0$  in the rest of the chapter to limit the notation clutter as its dimension can be determined in the context. It is worth noting that except the  $l_1$  penalty term, the other two terms in (38) are differentiable everywhere. Thus, most existing  $l_1$ -minimization algorithms [164] can be modified to solve the problem effectively. In this work, we adopt one of the iterative shrinkage/thresholding approaches, named two-step iterative shrinkage/thresholding (TwIST) [13], to solve it.

Considering the sparse coefficients are fixed, we first update the class-specific dictionaries  $\{\hat{\mathbf{D}}_i\}_{i=1}^C$  class by class and then update the common dictionary  $\mathbf{D}_0$ . Specifically, when  $\mathbf{A}_i$  and  $\mathbf{D}_0$  are fixed, the optimization of  $\hat{\mathbf{D}}_i$  is reduced to the following problem:

$$\begin{aligned} \min_{\hat{\mathbf{D}}_i} & \|\mathbf{X}_i - \mathbf{D}_0 \mathbf{A}_i^0 - \hat{\mathbf{D}}_i \hat{\mathbf{A}}_i\|_F^2 \\ \text{s.t.} \quad & \|\hat{\mathbf{d}}_j\|_2^2 \leq 1, \forall j = 1, \dots, K_i. \end{aligned} \quad (40)$$

After the class-specific dictionaries  $\{\hat{\mathbf{D}}_i\}_{i=1}^C$  are updated, we can further update the basis in the common dictionary  $\mathbf{D}_0$  by solving the following optimization:

$$\begin{aligned} \min_{\mathbf{D}_0} & \|\mathbf{X}^0 - \mathbf{D}_0 \mathbf{A}^0\|_F^2 \\ \text{s.t.} \quad & \|\mathbf{d}_j\|_2^2 \leq 1, \forall j = 1, \dots, K_0, \end{aligned} \quad (41)$$

---

**Algorithm 5.** Joint Dictionary Learning
 

---

Input: Data  $\{\mathbf{X}_i\}_{i=1}^C$ , sizes of dictionaries  $K_i$ ,  $i = 1, \dots, C$ , sparsity parameter  $\lambda$ , discrimination parameter  $\eta$ , and similarity threshold  $\xi$ .

- 1: repeat {Initialize  $\{\mathbf{D}_i\}_{i=1}^C$  and  $\{\mathbf{A}_i\}_{i=1}^C$  independently.}
- 2:   For each class  $i$  in the group with  $C$  classes, update  $\mathbf{A}_i$  by solving  $\min_{\mathbf{A}_i} \|\mathbf{X}_i - \mathbf{D}_i \mathbf{A}_i\|_F^2 + \lambda \|\mathbf{A}_i\|_1$ ;
- 3:   For each class  $i$  in the group with  $C$  classes, update  $\mathbf{D}_i$  by solving  $\min_{\mathbf{D}_i} \|\mathbf{X}_i - \mathbf{D}_i \mathbf{A}_i\|_F^2$  using its Lagrange dual.
- 4: until convergence or certain rounds.
- 5: Select the basis in  $\{\mathbf{D}_i\}_{i=1}^C$  whose pairwise similarities (inner-product) are bigger than  $\xi$  and stack them column by column to form the initial  $\mathbf{D}_0$ .
- 6: Compute the initial  $\{\hat{\mathbf{D}}_i\}_{i=1}^C$  such that  $\mathbf{D}_i = [\mathbf{D}_0, \hat{\mathbf{D}}_i]$ .
- 7: repeat {Jointly updating  $\{\hat{\mathbf{D}}_i\}_{i=1}^C$  and  $\mathbf{D}_0$ .}
- 8:   For each class  $i$  in the group with  $C$  classes, update  $\mathbf{A}_i$  by optimizing (38) using TwIST [13].
- 9:   For each class  $i$  in the group with  $C$  classes, update  $\hat{\mathbf{D}}_i$  by solving the dual of (40).
- 10:   Update  $\mathbf{D}_0$  by solving the dual of (41).
- 11: until convergence or after certain rounds.

Output: The learned category-specific dictionaries  $\{\hat{\mathbf{D}}_i\}_{i=1}^C$  and the shared common dictionary  $\mathbf{D}_0$ .

---

where

$$\mathbf{A}^0 \triangleq [\mathbf{A}_1^0, \dots, \mathbf{A}_C^0], \quad (42)$$

$$\mathbf{X}^0 \triangleq [\mathbf{X}_1 - \hat{\mathbf{D}}_1 \hat{\mathbf{A}}_1, \dots, \mathbf{X}_C - \hat{\mathbf{D}}_C \hat{\mathbf{A}}_C]. \quad (43)$$

Both (40) and (41) are the least squares problems with quadratic constraints, which can be solved efficiently by using their Lagrange dual forms[83]. The overall optimization procedure of our JDL model is summarized in Algorithm 5.

#### 5.4 Large-Scale Image Classification

Learning discriminative dictionaries aims to extract more discriminative visual features for image content representation. To make better use of the discriminative dictionaries learned by JDL, three schemes are designed for classifier training and im-

age classification under different configurations: (1) local classification scheme; (2) global classification scheme; and (3) hierarchical classification scheme.

#### 5.4.1 Local Classification Scheme

Once the dictionaries for a group of visually similar categories have been trained by JDL, classifying a test image into one particular category in the group can be done effectively by making use of the residual errors provided by different category dictionaries. While this strategy has achieved good results in [161, 122], better results have been reported in [102, 104, 168] by considering the discrimination of the sparse coefficients. For example, in [102, 104], the classification decision was based on the reconstruction errors, and in [168] the discrimination of the sparse codes was exploited by calculating the distances between the coefficients and the class centroids. In addition, classifiers were trained either simultaneously with the dictionary learning process [103, 167, 174, 73] or as a second step [166, 17] to make use of the discriminative coefficients.

Given a test image, multiple versions of content representation can be obtained based on different category dictionaries learned by our JDL algorithm. As illustrated in Fig. 25, each category dictionary comprises the common dictionary  $\mathbf{D}_0$  shared by all the  $C$  classes and as well as the category-specific dictionaries  $\hat{\mathbf{D}}_i$ ,  $i = 1, \dots, C$ . To make full use of the multiple versions of representation, we train a linear SVM based on each of them, and combine the outputs of all the linear SVMs to yield the final prediction using an equal voting scheme.

For visually correlated categories, learning their inter-related dictionaries jointly

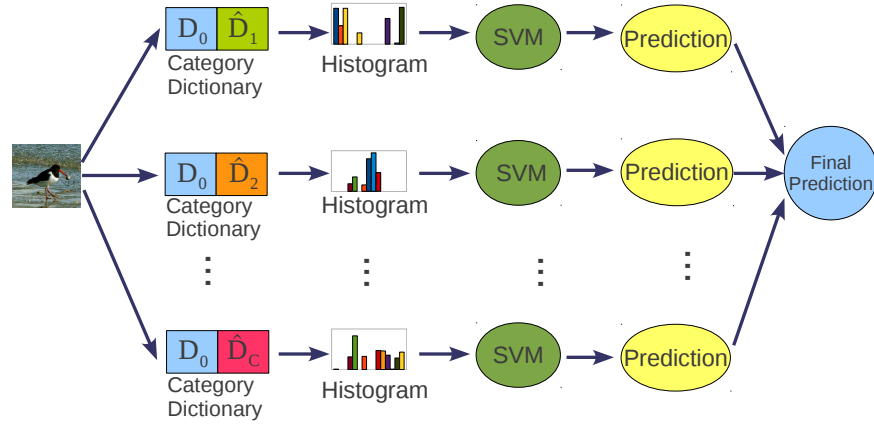


Figure 25: Illustration of the local classification scheme when the labels of test images are defined as the visually similar classes within a single group.

with the JDL algorithm can explicitly separate the common visual atoms from the category-specific ones. Therefore, more discriminative visual features can be extracted for image content representation. Thus, the local classification scheme can be used to assess the effectiveness of JDL on learning discriminative dictionaries for distinguishing a number of visually similar categories.

#### 5.4.2 Global Classification Scheme

It is too simplified to assume that the labels of test images are only defined as the classes in a single group. A large-scale visual recognition system should be able to distinguish a large number of classes from different groups. A global classification scheme is therefore developed as illustrated in Fig. 26, where the categories are clustered into  $T$  groups. First, the local features of an image are encoded using the  $T$  different group dictionaries to produce various local histograms which are then concatenated to form an image-level signature. The group dictionary of the  $t$ th group is constructed by concatenating the common dictionary  $\mathbf{D}_0^{(t)}$  and the class-specific parts  $\hat{\mathbf{D}}_i^{(t)}$ ,  $i = 1, \dots, C_t$ , where  $C_t$  is the number of categories in the  $t$ th group.



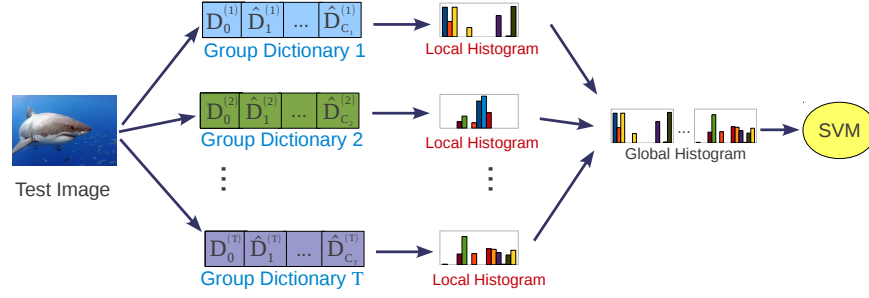


Figure 26: Illustration of the global classification scheme when the labels of test images are defined as the classes from  $T$  different groups.

Finally, the image-level signature is used as the input feature for SVM training and image classification.

#### 5.4.3 Hierarchical Classification Scheme

In the local and global classification schemes, the computational complexity of prediction grows linearly with the number of visual categories. One way to reduce the computational cost is to hierarchically organize the image categories in a tree structure according to their inter-category relations. In this section, we compare two different methods (*i.e.*, the label tree [9] and visual tree in Section 5.2) for tree structure construction. Also, we argue that the classifiers in hierarchical categorization can be trained more effectively with the dictionaries learned by JDL.

##### 5.4.3.1 Label Tree for Hierarchical Category Organization

In the label tree [9], each tree node is associated with a number of classes and a predictor. The set of classes is required to be a subset of the classes which are associated with its parent node, and the predictor is used to determine the best-matching child node to follow at the next level. Each leaf node corresponds to a particular image category. As described in Section 5.2.3,  $M$  OVR classifiers are pre-

trained to obtain the confusion matrix  $\mathbf{C}$  for learning the label tree structure. When the OVR classifiers and confusion matrix are reliable, the label tree method tends to assign visually correlated categories into the same node. However, training a large number of OVR classifiers is computationally expensive, and often suffers from the problem of huge sample imbalance. That is, the negative instances from the other  $M - 1$  categories heavily outnumber the positive samples of a given category. In addition, the negative instances may have huge visual diversity and can easily control and mislead the process of classifier training. The issue of sample imbalance may result in unreliable OVR classifiers which further produces a misleading confusion matrix for learning the structure of a label tree.

#### 5.4.3.2 Visual Tree for Hierarchical Category Organization

The proposed visual tree (Section 5.2) method can be easily modified to hierarchically organize the image categories as well. After the visual affinity matrix  $\mathbf{S} \in \mathbf{R}^{M \times M}$  of  $M$  categories is computed, a tree structure can be constructed by recursively partitioning the categories based on  $\mathbf{S}$  with any applicable clustering algorithm, such as spectral clustering[110] and AP clustering [57], to name a few. The tree structure is often application oriented, and can be explicitly controlled by specifying the branch factor (the maximum children of a tree node) and the maximum depth allowed in the visual tree.

#### 5.4.3.3 Visual Tree versus Label Tree

To compare the visual tree and label tree on hierarchical category organization, we constructed the two trees with the same configuration where branch factor was set

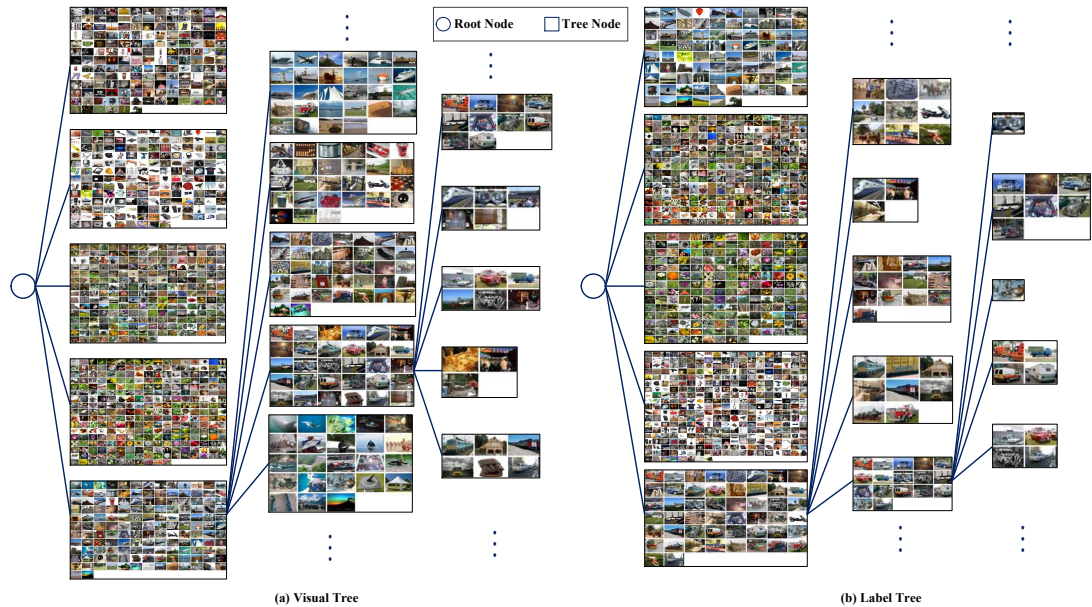


Figure 27: Visualization of the visual and label tree structures. The visual and label trees were built by recursively performing spectral clustering on the visual affinity matrix and classification confusion matrix, respectively. The branching factor is 5 and the maximum depth is 4 in both trees. The leaf nodes are not shown since each of them contains only one image category.

to be 5, maximum depth was set as 4 and the spectral clustering [110] was used for category partition. We visualize the structures of the visual tree and label tree in Fig. 27, where an icon image is selected to represent and illustrate an image category. The icon images of the categories in the same tree node are tiled together to visually illustrate it. It is seen that the proposed visual tree algorithm produces a similar tree structure as the label tree method. However, compared to the confusion matrix required by the label tree method [9], the visual affinity matrix used in the visual tree method are much cheaper to obtain in terms of computational cost, since many OVR classifiers have to be learned in advance to compute the confusion matrix.

Under the same configuration (*i.e.*, same branch factor, maximum depth and clustering method), our visual tree (Fig. 27 (a)) exhibits an even more balanced tree

structure than the label tree (Fig. 27 (b)). One of the reasons is that the confusion matrix is generally sparse with a lot of zero values, which may result in unbalanced clustering results for label tree construction. However, the zero values in the confusion matrix not necessarily indicate that the corresponding inter-category relations do not exist. On the other hand, the visual affinity matrix is guaranteed to be full which often leads to more balanced clustering results for visual tree construction.

#### 5.4.3.4 JDL for Hierarchical Image Classification

To evaluate the effectiveness of the dictionaries learned by JDL for classifier training in the setting of hierarchical image classification, the visual tree of depth *two* is used as we only partition the categories into disjoint groups for one time in Section 5.2. The hyperbolic visualization of the tree structure is shown in Fig. 28.

In the visual tree of depth *two*, there are two types of classifiers: (1) the group classifiers which serve as the predictors to determine the best-matching group for a test image; and (2) the category classifiers which are used to identify the most confident category in the group which has been selected by the group classifiers. As argued in [72], the visual features which are effective in distinguishing various super-categories (*i.e.*, groups of categories) are usually different from the features which are useful for discriminating the image categories at sub-levels. In other words, the feature space which is particularly effective for group classifier learning is often different from that for category classifier training.

Suppose that a large number of categories are clustered into  $T$  groups, and  $T$  group-based dictionaries have been learned by JDL accordingly. Each group-based

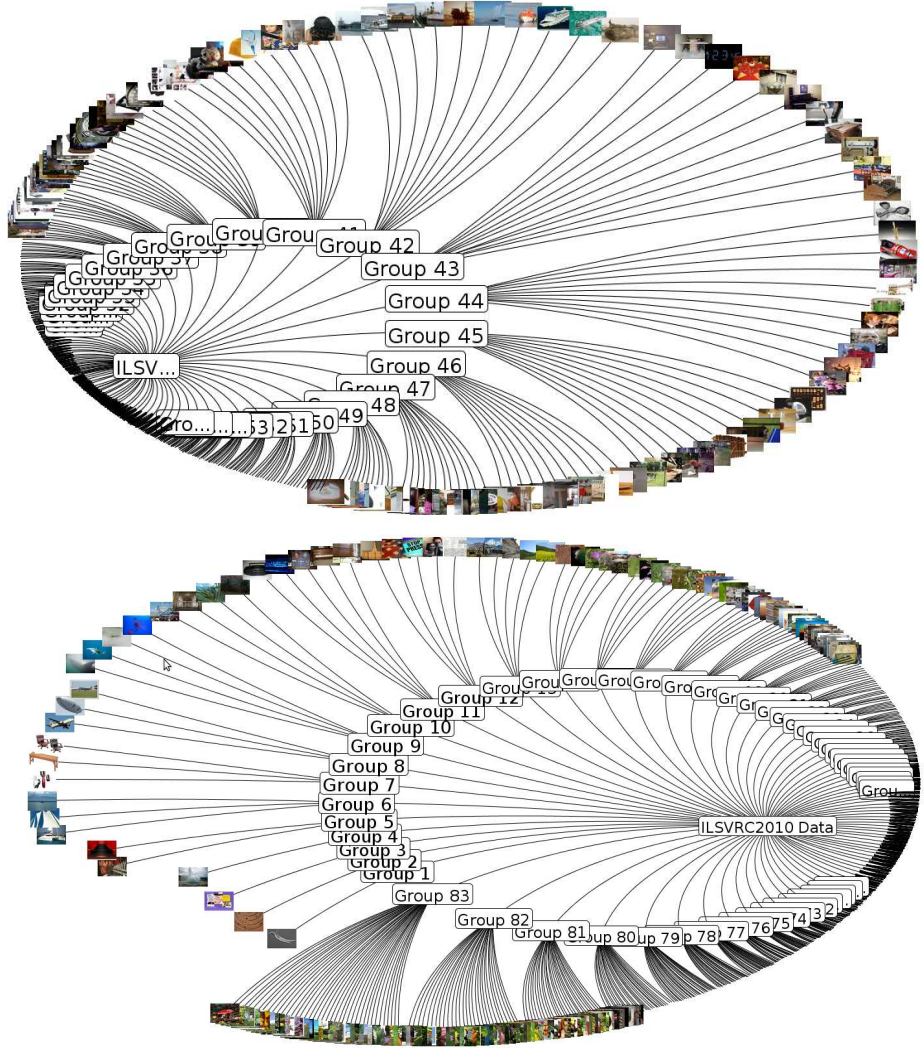


Figure 28: Hyperbolic visualization of the visual tree of depth 2 for the ILSVRC2010 data set. AP clustering was used to partition the image categories (Section 5.2.2).

dictionary has the same structure (*i.e.*, a common dictionary and multiple category-specific dictionaries) as described in Section 5.3.1. We concatenate the  $T$  common dictionaries  $\{\mathbf{D}_0^{(t)}\}_{t=1}^T$  as  $[\mathbf{D}_0^{(1)}, \dots, \mathbf{D}_0^{(T)}]$  to form the feature space for group classifier training. The category-specific dictionaries of the  $C_t$  classes in the  $t$ th group are concatenated as  $[\hat{\mathbf{D}}_1^{(t)}, \dots, \hat{\mathbf{D}}_{C_t}^{(t)}]$  to establish the feature space for training its own category classifiers. The proposed hierarchical classification scheme using the dictionaries learned by JDL is illustrated in Fig. 29.

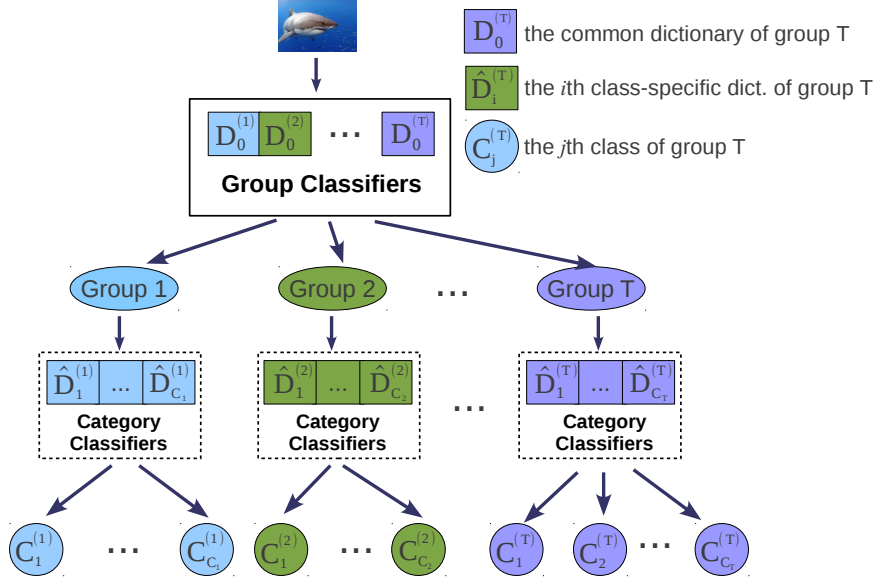


Figure 29: Illustration of the hierarchical classification scheme where the group and category classifiers are trained based on different dictionaries (*i.e.*, feature spaces).

## 5.5 Experiments

We evaluate the performance of JDL with different classification schemes based on two widely used data sets, including the Oxford flower image set of 17 classes and the ILSVRC2010 data set containing 1,000 categories. First, we adopt the local classification scheme on the Oxford flower image set to assess the effectiveness of JDL in learning discriminative dictionaries for distinguishing a number of visually similar categories. Second, to further evaluate the performance of JDL in large-scale visual recognition, we assess it using the global classification as well as the hierarchical classification schemes on the ILSVRC2010 image database. Third, we empirically investigate the convergence and discrimination of JDL. Finally, we give the time complexity of JDL.

### 5.5.1 Experimental Setup

We describe the common experimental setup in this subsection for all the experiments, including visual feature extraction and parameter settings. The SIFT [97] descriptor is used as the local feature due to its excellent performance in object recognition and image classification [17, 165, 73]. Specifically, we extract SIFT descriptors from  $16 \times 16$  patches with a step size of 6 at 3 scales. The maximum width or height of an image is re-sized as 300 pixels, and the  $l_2$  norm of each SIFT descriptor is normalized to be 1. The sparsity parameter  $\lambda$  is set to be 0.15 in all experiments, and the parameter of the discrimination promotion term  $\eta$  in the JDL model is fixed as 0.1. They are determined via cross-validation. We set the similarity threshold  $\xi$  in Algorithm 5 to be 0.9 if required.

### 5.5.2 Evaluation on Oxford Flower Image Set

The Oxford flower benchmark contains 1,360 flower images of 17 classes, and each category has 80 images. Three predefined training, testing and validation splits provided by the authors in [111] are used in our experiments. Since the 17 flower categories have strong visual correlation, we use the proposed JDL algorithm to learn discriminative dictionaries by treating them as a single group. The local classification strategy (Section 5.4.1) is thus adopted, and we compare it with a number of other dictionary learning methods, including ScSPM [165], D-KSVD [174] and FDDL [168]. Also, we include another baseline method for comparison, named independent multiple dictionary learning (IMDL), which learns multiple class-based dictionaries independently rather than jointly (See line 1 to 4 in Algorithm 5). Finally, we evaluate the

Table 7: Recognition accuracy on the 17-category Oxford flower data set (continued in Table 8).


















									
ScSPM [49]	53.33	65.00	68.33	58.33	70.00	18.33	45.00	51.67	63.33
IMDL	63.33	93.33	80.00	58.33	73.33	33.33	58.33	<b>78.33</b>	<b>86.67</b>
D-KSVD [55]	61.67	90.00	80.00	48.33	68.33	30.00	58.33	71.67	80.00
FDDL [52]	46.67	88.33	<b>88.33</b>	<b>68.33</b>	<b>78.33</b>	<b>41.67</b>	<b>71.67</b>	76.67	81.67
JDL	<b>75.00</b>	<b>95.00</b>	81.67	58.33	70.67	35.00	60.23	<b>78.33</b>	85.00

Table 8: Recognition accuracy on the 17-category Oxford flower data set (continued from Table 7).

									Avg.
ScSPM [49]	58.33	58.33	<b>50.00</b>	38.33	70.00	43.33	20.00	58.33	52.35
IMDL	<b>80.00</b>	66.67	40.00	<b>61.67</b>	<b>86.67</b>	<b>68.33</b>	35.00	70.00	66.67
D-KSVD [55]	75.00	65.00	31.67	58.33	81.67	45.00	33.33	66.67	61.47
FDDL [52]	76.67	61.67	51.67	46.67	76.67	46.67	<b>48.33</b>	80.00	66.47
JDL	75.00	<b>70.67</b>	45.00	60.00	<b>86.67</b>	65.33	45.23	<b>80.58</b>	<b>68.69</b>

necessity of explicitly separating the common visual atoms from the category-specific ones in the configuration of local classification.

#### 5.5.2.1 Comparison with Other Dictionary Learning Algorithms

Given an image, the spatial pyramid feature [79] is computed as the representation of an image by using max pooling with a three-level spatial pyramid partition. The image-level features are then used as the input for SVM training and image classification in ScSPM, IMDL and JDL. Note that the local classification scheme is also used in IMDL as multiple dictionaries are trained. Another important factor of JDL and IMDL is the dictionary size. For simplicity, we set the dictionary size for each class to be 256 in both JDL and IMDL. After the JDL algorithm converges, a common dictionary of 95 visual atoms is obtained, which is shared by all the 17 flower categories.



In D-KSVD, a linear classifier is simultaneously trained along with the dictionary learning process. In FDDL, however, the residual errors as well as the distances between sparse coefficients and class centroids are combined for classification. For a fair comparison, the dictionaries in D-KSVD and FDDL are learned based on the image-level spatial pyramid features rather than on local SIFT descriptors. Specifically, the spatial pyramid features are computed with a codebook of size 1,024 which is trained using the same method as in [83]. We further reduce the spatial pyramid features to certain dimensions with PCA before feeding them into the D-KSVD and FDDL models as stated in [174]. The dictionary size in ScSPM and D-KSVD is set the same as 2,048.

We show the experimental results in Table 7 and 8. It is clearly seen that our JDL algorithm consistently outperforms other dictionary learning algorithms, namely IMDL, D-KSVD and FDDL, in terms of the average accuracy. It proves that JDL is able to learn more discriminative dictionaries to distinguish a group of visually similar categories by separating the commonly shared visual atoms from the category-specific ones.

#### 5.5.2.2 Effectiveness of $\mathbf{D}_0$ in Local Classification

A common dictionary  $\mathbf{D}_0$  is designed in JDL to capture the common visual patterns which are shared by the visually correlated categories. As the discrimination of the shared features are often weak, separating them from the class-specific features enables JDL to learn more discriminative dictionaries. To evaluate the effectiveness of the common dictionary, we train 17 dictionaries of size 256 for the 17 flower categories

Table 9: Performance comparison of the JDL algorithms with and without separating the common visual atoms ( $\mathbf{D}_0$ ) from the category-specific ones ( $\{\hat{\mathbf{D}}_i\}_{i=1}^{17}$ ) for the 17-category Oxford flower data set.

Models	Accuracy (%)
MCLP [60]	66.74
KMTJSRC [171]	69.95
HCLSP [24]	63.15
HCLSP_ITR [24]	67.06
JDL without $\mathbf{D}_0$	67.15
JDL	68.69

without devising the common dictionary explicitly, denoted as “JDL without  $\mathbf{D}_0$ ”.

The local classification scheme is also used for classification once the dictionaries have been learned. We present the comparison in Table 9. It shows that separating the common visual atoms from the category-specific ones is effective in enhancing the discrimination of the dictionaries, and can lead to performance boosting.

Also, we compare JDL with a number of other state-of-the-art methods on this benchmark, which combine various types of visual features (color histogram, BoW, and HoG) for recognition. They include multi-class LPboost (MCLP) [60], visual classification with multi-task joint sparse representation (KMTJSRC) [171], histogram-based component-level sparse representation (HCLSP) and its extension (HCLSP\_ITR) [24]. The performance of our JDL algorithm is comparable to that of KMTJSRC, which, however, combines multiple visual features via a multi-task joint sparse representation.

### 5.5.3 Evaluation on ILSVRC2010 Image Set

The ILSVRC2010 data set contains 1.4M images of 1,000 categories. The standard training/validation/testing split is used (respectively 1.2M, 50K and 150K images).

We first present the results of different category grouping methods, and how they affect the performance of the JDL algorithm. Second, we evaluate the effectiveness of the JDL model in the setting of hierarchical image classification which provides two disjoint feature spaces for group and category classifiers training. Finally, we compare JDL with a number of state-of-the-art methods on this data set.

The final performance of a visual recognition system can be affected by many factors, such as the number of training images and the classifier training method. We follow the “good practice” in [116] to train linear SVMs in all the following trials, so that the effectiveness of different dictionary learning and category clustering methods can be seen clearly. Specifically, the OVR SVMs are adopted to support multi-class classification. We use Stochastic Gradient Descent (SGD) [16] to train the SVMs due to its efficiency in processing large-scale data. The parameters of SGD are optimized based on the validation set. For the computation, a computer cluster of 492 computing cores is used.

#### 5.5.3.1 Comparison on Image Category Clustering

The visual tree and label tree methods use two different types of information (*i.e.*, visual correlations and confusion matrix) for image category clustering. We randomly select 100 images per category as the training data to estimate them. In the visual tree method, the 100 images are used to compute the average visual representation (Section 5.2.1) for the corresponding category. In the label tree method, the 100 images are further split into training and testing at the ratio of 3:2, where the training set is used for training the OVR SVMs, and the test set is used to obtain the confusion

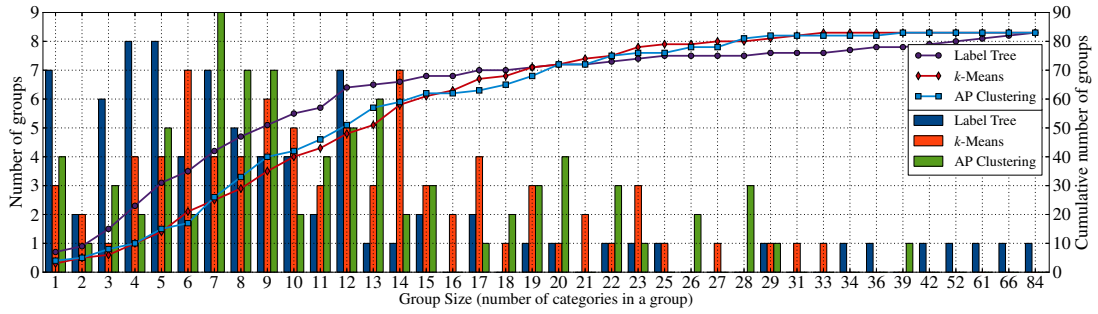


Figure 30: The number of groups (bar, units indicated in the left y-axis) and cumulative number of groups (line, units indicated in the right y-axis) of different image category clustering methods on ILSVRC2010 data set.

matrix. The number of disjoint category groups is fixed as 83 in both label tree and visual tree methods.

Fig. 30 presents the distributions of the group sizes obtained by the two methods (*i.e.*, label tree, visual tree based on  $k$ -means and AP clustering). We plot the numbers of groups across different group sizes. It shows that the sizes of the biggest groups generated by the label tree, visual tree with  $k$ -means and visual tree with AP clustering are 84, 33 and 39, respectively. The label tree method tends to produce unbalanced clustering results with many small groups (*e.g.*, the number of categories in the same group is less than 8) and some very large groups (*e.g.*, four groups have more than 50 classes). A number of groups determined by the methods are illustrated in Fig. 31 where the groups in each row have at least one overlapping category.

Given various methods for category grouping, it is desired to quantitatively evaluate how they contribute to the performance of dictionary learning algorithms. We compare our JDL algorithm with the unsupervised dictionary learning (UDL) method [83]. In UDL, three different image category clustering strategies are adopted. First, a single dictionary of size 8,192 is trained without category clustering being performed.



Figure 31: Example groups identified by different image category clustering methods on ILSVRC2010 data set. Each cell shows the sample pictures of the categories in the same group.

Second, we randomly partition the 1,000 categories into 83 groups, and learn one dictionary for each group using UDL (UDL + Random Group). Third, we cluster the categories into 83 groups by using the visual tree method with AP clustering, and then use UDL to learn the dictionary for each group. For evaluating the proposed JDL algorithm, we also implement three versions of JDL based on three different methods for category clustering, namely the label tree, the visual tree with  $k$ -means

and the visual tree with AP clustering.

Since the categories are clustered into a number of disjoint groups, an important issue is to determine the dictionary size for each group. For a given group, the dictionary size essentially depends on its visual complexity and diversity. We set the dictionary size for a group to be proportional to the number of the categories in it. Specifically, let  $C_i$  be the number of categories in a group. The dictionary size for it is decided as  $8 \times C_i$  when UDL is adopted. In JDL, the sizes of the common and class-specific dictionaries are set to be  $3 \times C_i$  and  $5 \times C_i$ , respectively. One reason for this setting is that the number of common visual atoms in a group should be smaller than that of the category-specific ones since the categories in it are still visually different from each other even though they have strong visual correlations. Obviously, the size of the common dictionary in a group can be dynamically determined by the parameter  $\xi$  in Algorithm 5. For this data set, we do not set the dictionary sizes dynamically to strictly control the total number of visual words used in JDL (*i.e.*, 8K in total). Therefore, a fair comparison could be made between the JDL and UDL algorithms.

Finally, after encoding the local features extracted from an image over the learned dictionaries, a configuration of two-level spatial partitions, *i.e.*,  $1 \times 1$  and  $2 \times 2$ , is used to pool the codes into an image-level signature. The global classification scheme is used here to obtain the accuracy rates for categorizing the 1,000 classes. In Table 10, we summarize the configurations of different dictionary learning and category clustering methods to be compared. Fig. 32 shows the comparison between UDL and JDL based on different category clustering methods. First, it is seen that clustering a large number of categories into disjoint groups improves the results even

Table 10: The configurations of different dictionary learning and image category clustering methods. # of Groups: the number of groups; Total # of Words: the total number of visual words used in all dictionaries; Feat. Dim.: the dimension of the feature vector fed to SVM; UDL: unsupervised dictionary learning.

	# of Groups	Total # of Words	Feat. Dim.
UDL + Single Dictionary	1	8192	$8192 \times 5$
UDL + Random Group	83	$8 \times 1000$	$8000 \times 5$
UDL + AP Clustering	83	$8 \times 1000$	$8000 \times 5$
JDL + Label Tree	83	$(3 + 5) \times 1000$	$8000 \times 5$
JDL + $k$ -Means	83	$(3 + 5) \times 1000$	$8000 \times 5$
JDL + AP Clustering	83	$(3 + 5) \times 1000$	$8000 \times 5$

in UDL. For image search task, Aly *et al.* [3] have reported similar results when multiple individual dictionaries were trained by randomly partitioning the training samples into a number of disjoint sets. Second, when the same category clustering method is used, JDL (JDL + AP Clustering) learns more discriminative dictionaries which lead to higher categorization accuracy rates as compared to UDL (UDL + AP Clustering). Third, the visual tree method based on  $k$ -means or AP clustering is more effective than the label tree method in clustering image categories to support the proposed JDL algorithm, and achieves slightly better classification results. Note that the experimental results of JDL on two selected groups using the local classification scheme were reported in [182].

#### 5.5.3.2 Effectiveness of $\mathbf{D}_0$ in Hierarchical Classification

As discussed in Section 5.4.3, the feature spaces which are effective for learning the group and category classifiers are usually different. One of the advantages of our JDL model is that the common dictionaries can be used to extract visual features for group classifier training while the class-specific dictionaries can be utilized to

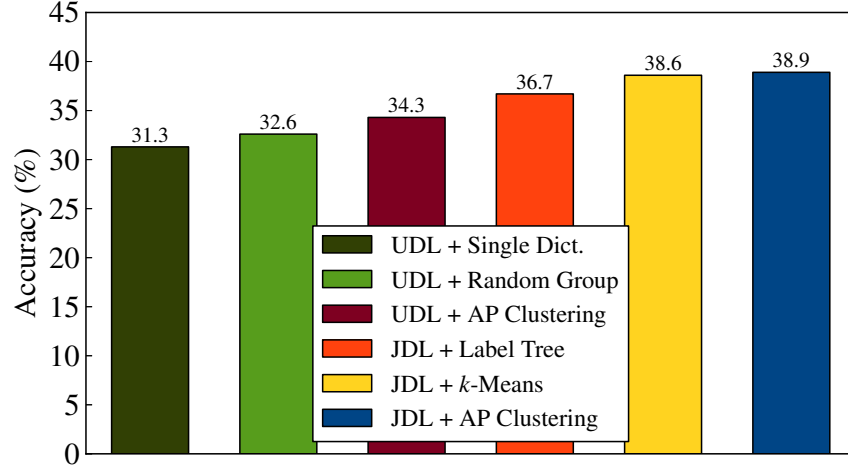


Figure 32: Classification accuracy rates using the dictionaries learned by JDL and UDL based on different image category clustering methods for ILSVRC2010 data set.

extract features for category classifier learning. To assess the effectiveness of this strategy in hierarchical image classification, we compare JDL with UDL. In UDL, a single dictionary is learned to extract features for training both group and category classifiers. The visual tree produced by AP clustering is used to organize the categories for hierarchical image classification. The common dictionaries  $\{\mathbf{D}_0^{(i)}\}_{i=1}^{83}$  learned by JDL for the 83 groups are concatenated to form a dictionary of 3,000 visual words. The dictionary is thus used as the feature space for group classifier training. For the  $t$ th group ( $t = 1, \dots, 83$ ), its category-specific dictionaries  $\{\hat{\mathbf{D}}_i^{(t)}\}_{i=1}^{C_t}$  are used to form the feature space for learning its own category classifiers, where  $C_t$  is the number of categories in the  $t$ th group.

In implementation, given the local descriptors  $\mathbf{X}$  of an image, we individually encode them over the 83 group dictionaries, and obtain 83 different versions of sparse codes, denoted as  $\{[\mathbf{A}_0^{(t)}; \hat{\mathbf{A}}^{(t)}]\}_{t=1}^{83}$ . Note that the dictionary size for each group is relatively small (*e.g.*, the biggest dictionary of the largest group only consists of



Table 11: Comparison between using the discriminative dictionaries (common and category-specific dictionaries) learned by JDL and the single dictionary trained by UDL for visual representation in hierarchical image classification. GC: group classifier; CC: category classifier; Feat. Dim.: feature dimension.

	UDL + Single Dict.	JDL + Group Dict.
Feat. Dim. for GC	$3000 \times 5$	$3000 \times 5$
Feat. Dim. for CC	$3000 \times 5$	varies across groups
Accuracy (%)	7.6	9.5

$39 \times 8 = 312$  visual words) which makes the encoding process very computationally efficient. The coefficients corresponding to the common dictionaries are concatenated as  $[\mathbf{A}_0^{(1)}; \dots, \mathbf{A}_0^{(83)}]$  to yield the visual features for group classifier training. In the  $t$ th group, the sparse codes  $\hat{\mathbf{A}}^{(t)}$  corresponding to its own category-specific dictionaries are utilized as the features to train the category classifiers in the group.

For comparison, one single dictionary of size 3,000 learned by UDL is used for both group and category classifier training. In Table 11, we tabulates the results of hierarchical image classification by using the dictionaries learned by JDL and UDL for feature extraction, respectively. It is seen that JDL outperforms UDL, since different feature spaces (*i.e.*, different dictionaries) are used for training the group and category classifiers. On the other hand, UDL uses the same feature space (*i.e.*, the same dictionary) to learn both group and category classifiers.

### 5.5.3.3 Comparison with State-of-the-Art Results

In this section, we compare our JDL algorithm with a few state-of-the-art methods on the ILSVRC2010 data set, including Fisher Vector [117], method of NEC [90], the winner team at ILSVRC2010, and the Meta-Class feature (MC) [11]. The performance comparison is shown in Table 12. The proposed JDL algorithm achieves better

Table 12: Comparison between JDL and a few state-of-the-art methods on the ILSVRC2010 data set.

	Visual Features	Coding	Feat. Dim.	Accuracy (%)
Fisher Vector [116]	SIFT	Fisher Vector	131,072	45.7
NEC [90]	LBP, HOG	LCC [155], Super-vector [184]	262,144 <sup>a</sup>	52.9
MC-Bit [11]	GIST, HoG, SSIM, SIFT	MC feature	15,458	36.4
JDL + AP Clustering	SIFT	Sparse Coding	40,000	38.9

<sup>a</sup> Only the value of the largest dimension is shown as multiple features and encoding methods were used.

result than MC-Bit [11], but does not perform as well as the Fisher Vector and the method of NEC. However, Fisher Vector takes the advantage of higher dimensional features, and NEC combines the HoG and LBP (local binary pattern) features, multiple encoding methods and fine-grained spatial pyramids to achieve better results.

#### 5.5.4 Convergence and Discrimination

The convergence of JDL indicated by the values of the objective function (33) over iterations is plotted in Fig. 33. Three groups of different sizes are randomly selected from the 83 groups which are generated by the visual tree method with AP clustering on the ILSVRC2010 data set. One can observe that after a few iterations, our JDL algorithm always empirically converged. In addition, we have quantitatively analyzed the discrimination of the sparse coefficients based on the Fisher score which is defined as  $\frac{\text{tr}(S_W)}{\text{tr}(S_B)}$ , where  $S_W$  and  $S_B$  are the within-class scatter matrix and the inter-class scatter matrix of the sparse codes, respectively. A smaller value of Fisher score implies that the current sparse representation has stronger discrimination. Fig. 34 shows the Fisher scores of JDL over iterations for three groups. The Fisher scores decrease over iterations which demonstrate that more discriminative sparse coefficients are obtained as the JDL algorithm iterates.

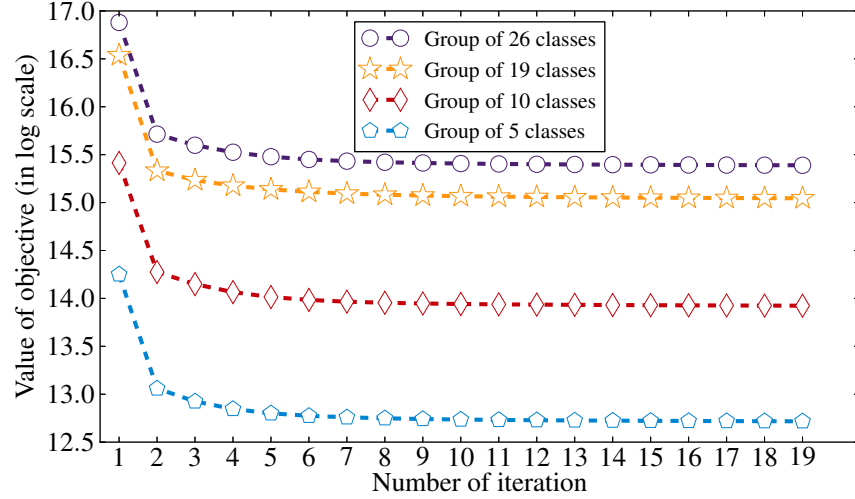


Figure 33: The values of the objective function (Eq. 33) in log scale of JDL on four image category groups of different sizes.

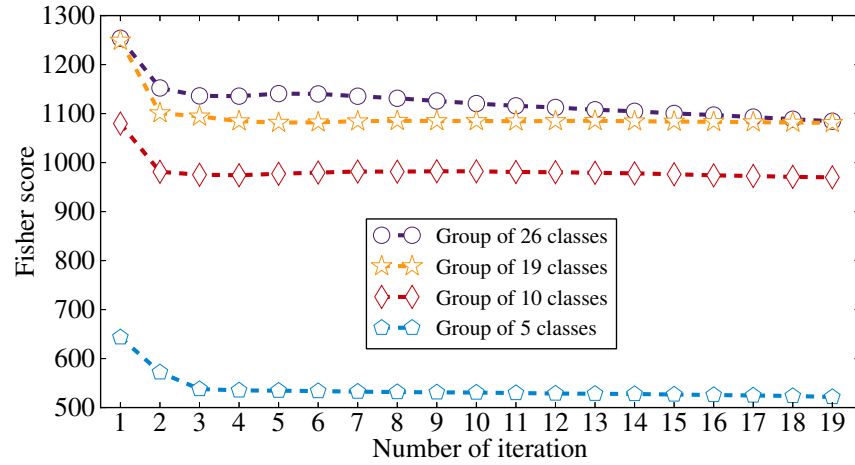


Figure 34: The Fisher scores ( $\text{tr}\left(\frac{S_W}{S_B}\right)$ ) of JDL on four image category groups of different sizes.

#### 5.5.5 Dictionary Size

We further investigate how sensitive JDL and IMDL are to the choice of the dictionary size per class  $K_i$ . Intuitively, increasing the dictionary size often leads to better results at the expense of increasing computational cost. We plot the overall categorization performance of JDL and IMDL across different choices of  $K_i$  for the

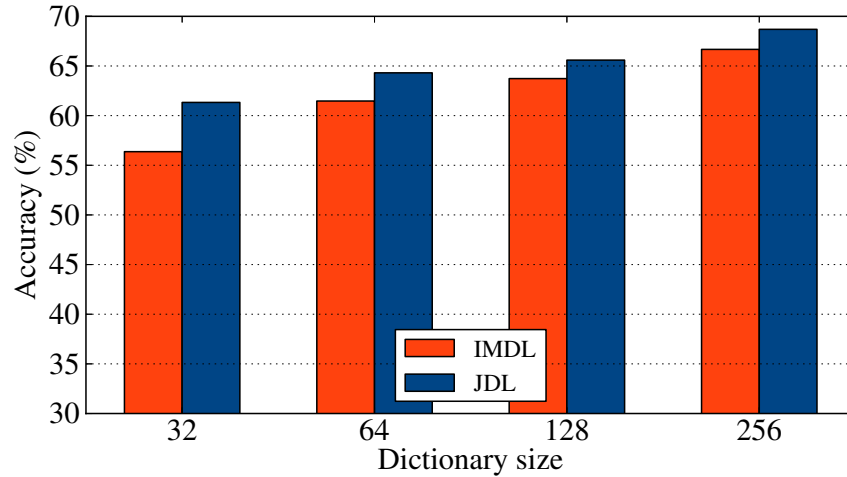


Figure 35: Comparison between JDL and IMDL using different dictionary sizes per category on the Oxford flower data set.

Oxford flower image set in Fig. 35. One can observe that JDL outperforms IMDL under the configurations of all the different the dictionary sizes, and the performance gain increases when the number of visual words decreases.

#### 5.5.6 Computational Complexity of JDL

Compared with the unsupervised dictionary learning algorithms, our JDL algorithm can extract more discriminative visual features by learning more discriminative dictionaries, and achieve better classification accuracy. The drawback of JDL is that it is computationally more complex. Although dictionary learning can be done in parallel and off-line, it is still important to see how long the dictionary learning process would take. A number of experimental parameters can affect the run time of the dictionary learning, including the number of categories, number of training samples, dictionary size and dimension of local descriptors. The run time performance of JDL is shown in Fig. 36 based on different numbers of training samples per category. The timing is based on a *single* core of an 8-core Xeon 2.67GHz server node without fully

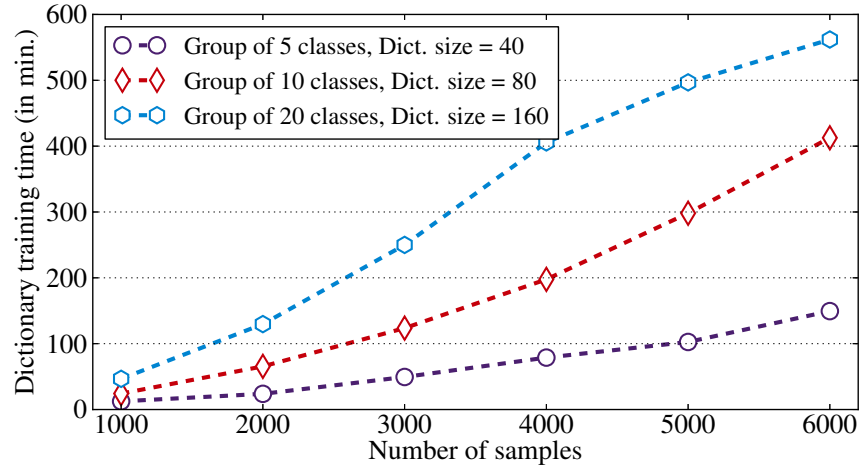


Figure 36: Dictionary training time of JDL on three image category groups of different sizes.

optimizing the code.

## 5.6 Summary

In this chapter, a novel joint dictionary learning (JDL) algorithm has been developed to learn more discriminative dictionaries by explicitly separating the common visual atoms from the category-specific ones. For a group of visually correlated classes, a common dictionary and multiple class-specific dictionaries are simultaneously modeled in JDL to enhance their discrimination power, and the processes of learning the common dictionary and multiple class-specific dictionaries have been formulated as a joint optimization by adding a discrimination promotion term based on the Fisher discrimination criterion. The visual tree as well as the label tree methods have been employed to cluster a large number of image categories into a set of disjoint groups. The process of image category clustering not only ensures that the categories in the same group are of strong visual correlation, but also makes the JDL algorithm to be computationally affordable in large-scale visual recognition applications. Three

schemes have been developed to take the advantage of the discriminative dictionaries learned by JDL for image content representation, classifier training and image classification. The experimental results have demonstrated that our JDL algorithm is superior to many unsupervised and supervised dictionary learning algorithms, especially on dealing with visually similar categories.

## CHAPTER 6: SEMANTIC GAP MODELING

### 6.1 Introduction

With the exponential growth of digital images, there is an urgent need to achieve automatic concept detection for supporting concept-based (keyword-based) image retrieval [138]. Unfortunately, there is a fundamental barrier of semantic gaps when low-level visual features are used to represent high-level image concepts. The semantic gap can be defined as the difference on the expression power between the low-level visual features (*i.e.*, computational representations of the visual content of the images from computers) and the high-level image concepts (*i.e.*, semantic interpretations of the visual content of the images from human beings) [108, 100, 68, 99, 67]. To bridge the semantic gaps, machine learning tools are usually used to learn the concept classifiers from large amounts of labeled training images (*i.e.*, learning the mapping functions between the low-level visual features and the high-level image concepts) [37, 66, 101, 178, 124, 44]. However, it is not a trivial task because the learning complexities for concept classifier training could vary with the image concepts significantly, *e.g.*, some image concepts may have lower learning complexities for concept classifier training because their semantic gaps are smaller, on the other hand, some image concepts may have higher learning complexities for concept classifier training because their semantic gaps are larger.

To achieve more effective concept classifier training, it is very important to support quantitative characterization of the semantic gaps. It is worth noting that both concept classifier training and automatic concept detection are performed in the visual feature space rather than in the label space, thus it is very attractive to develop new algorithms that can support quantitative characterization of the semantic gaps directly in the visual feature space, so that we can automatically estimate the learning complexities and select more effective inference models for concept classifier training. For a given image concept with small semantic gap, there may exist a unique mapping function (concept classifier) between its feature-based visual representation and its semantic interpretation, *e.g.*, the concept classifier for the given image concept is isolated from the concept classifiers for other image concepts in the visual feature space, which may further result in high discrimination power on concept detection. For a given image concept with large semantic gap, its concept classifier is not unique and may overlap with the concept classifiers for other image concepts in the visual feature space, which may further result in low discrimination power on concept detection. Thus the scales (numerical values) of the semantic gaps can also be treated as an effective measurement of the learning complexities for concept classifier training.

When the image concepts are visually-related, their relevant images may share some common or similar visual properties (*i.e.*, huge inter-concept visual similarity), and it could be difficult for machine learning tools to obtain unique concept classifiers for distinguishing such visually-related image concepts precisely. For instance, the visually-related image concepts may not be visually separable because huge inter-concept visual similarity may cause significant overlapping among their concept clas-



sifiers and result in low discrimination power on concept detection [43, 147, 42, 14]. Thus, the image concepts which are visually-related with many other image concepts, will have larger semantic gaps and their learning complexities for concept classifier training will be higher.

When the image concepts have huge inner-concept visual diversity among their relevant images (*i.e.*, low inner-concept visual consistency), it could be very difficult for machine learning tools to use some simple models to approximate their diverse visual properties effectively, on the other hand, using some complex models to approximate their diverse visual properties completely may cause significant overlapping between their concept classifiers and the concept classifiers for other image concepts, which may further result in low discrimination power on concept detection. Thus the image concepts, which have huge inner-concept visual diversity (*i.e.*, low inner-concept visual consistency), will have larger semantic gaps and their learning complexities for concept classifier training will be higher.

Based on these observations, an inner-concept visual homogeneity score is defined for characterizing the inner-concept visual consistency among the relevant images for the same image concept, and an inter-concept discrimination complexity score is defined for characterizing the inter-concept visual correlations among the relevant images for multiple visually-related image concepts. By simultaneously considering both the inner-concept visual homogeneity scores and the inter-concept discrimination complexity scores, a novel data-driven approach is developed for supporting quantitative characterization of the semantic gaps directly in the visual feature space.

The rest of this paper is organized as follows. Section 6.2 presents our work

on feature extraction and image similarity characterization; Section 6.3 defines the inter-concept discrimination complexity score, where a visual concept network is constructed for characterizing the inter-concept visual similarity contexts explicitly and providing a good environment to determine the visually-related image concepts automatically; Section 6.4 defines the inner-concept visual homogeneity score; Section 6.5 introduces two different approaches for supporting quantitative characterization of the semantic gaps directly in the visual feature space; Section 6.6 presents our structural learning algorithm for concept classifier training by leveraging both the scales (numerical values) of the semantic gaps and the visual concept network for automatic inference model selection; Section 6.7 describes our work on algorithm evaluation on two well-known image sets; We conclude this chapter in Section 6.8 .

## 6.2 Feature Extraction and Image Similarity Characterization

A large number of image concepts and their relevant images are used to assess the effectiveness and robustness of our data-driven algorithm on quantitative characterization of the semantic gaps. These image concepts and their relevant images are collected from two well-known image sets: NUS-WIDE [25] and ImageNet [33]. In this paper, we focus on assessing the effectiveness and robustness of our data-driven algorithm on two types of image concepts: (1) scene-based and event-based image concepts (image semantics are interpreted by the visual content of entire images); and (2) object-based image concepts (image semantics are interpreted by the visual content of object regions or object bounding boxes).

NUS-WIDE image set [25] has collected large amounts of Internet images for 81

Table 13: The 81 image concepts in NUS-WIDE [25] for algorithm evaluation.

Categories	Image Concepts
Event/Activities	Swimming Earthquake Fire-exposition Surfing Running Wedding Dancing Protest Soccer
Scene/Location	Airport Temple Castle Beach Cityscape Snow Buildings Mountain Valley Street Railroad Road Harbor Bridge Sky Clouds Garden Glacier Sunset Reflection Night-time Water Grass Moon Frost Ocean Window Plants Waterfall Lake Rainbow House Town
People	Police Military Person Tattoo
Objects	Animal Vehicle Flags Birds Tiger Bear Car Toy Tree Train Boats Cat Horse Fox Elk Cow Computer Whales Zebra Fish Dog Tower Statue Coral Rocks Sign Flowers Leaf Sand Food Sun Book Plane
Graphics	Map
Program	Sports

image concepts. For the NUS-WIDE image set, all these 81 image concepts are illustrated in Table 13, and they are used for assessing the effectiveness and robustness of our data-driven algorithm on supporting quantitative characterization of the semantic gaps.

ImageNet [33] has collected more than 9,353,897 Internet images, and it contains more than 14,791 image concepts at different semantic levels. In this paper, only 1,000 image concepts (1,000 most popular real-world object classes and scene categories), which contain large amounts of relevant images, are selected for assessing the effectiveness and robustness of our data-driven algorithm on supporting quantitative characterization of the semantic gaps directly in the visual feature space. A part of these 1,000 most popular image concepts (real-world object classes and scene categories) are given in Table 14.

For the scene-based and event-based image concepts (scene and event categories), each image is treated as one single image instance and feature extraction is done by partitioning each image (image instance) into a set of  $8 \times 8$  image patches. For each

Table 14: A part of the 1000 image concepts in ImageNet [33] for algorithm evaluation.

tree	beach	snow	mountain	scarf	earring	ring	abacus
shirt	flower	road	ladder	corridor	firework	sword	celery
broccoli	corn	tadpole	watch	garden	sunset	sailing	office
scissors	penguin	starfish	toad	zebra	monkey	mouse	home
chair	plane	rail	spiders	tiger	lion	cat	falls
street	landscape	screen	wall	sponge	mug	cobra	dolphin
rainbows	water	city	building	parks	cars	teapot	snake
roses	golf	bears	dagger	...	...	...	...

$8 \times 8$  image patch, the following visual features are extracted: (1) top 3 dominant colors; (2) 12-bin color histogram; (3) 9-dimensional Tamura texture features; and (4) SIFT features. For each  $8 \times 8$  image patch, its best-matching “visual word” is found from a pre-trained codebook with 512 visual words (codewords), and a 512-bin codeword histogram (histogram of 512 visual words) is extracted and used to represent the principal visual properties of the given image instance.

For the object-based image concepts (object classes), ImageNet [33] has provided the object bounding boxes, which are used to indicate the appearances of the object classes and their locations in the images. We treat each object bounding box as one single image instance and feature extraction is done by partitioning each object bounding box (image instance) into a set of  $8 \times 8$  image patches. For each  $8 \times 8$  image patch, the following visual features are extracted: (1) top 3 dominant colors; (2) 12-bin color histogram; (3) 9-dimensional Tamura texture features; and (4) SIFT (scale invariant feature transform) features. For each  $8 \times 8$  image patch in a given image instance (object bounding box), its best-matching “visual word” is found from a pre-trained codebook with 512 visual words (codewords), and a 512-bin codeword histogram (histogram of 512 visual words) is extracted and used to represent the principal visual properties of the given image instance.

A kernel function is defined for measuring the visual similarity context  $\kappa(x, y)$  between two image instances  $x$  and  $y$  according to their 512-bin codeword histograms  $u$  and  $v$ :

$$\kappa(x, y) = e^{-\chi^2(u, v)/\sigma} = \prod_{i=1}^{512} e^{-\chi_i^2(u(i), v(i))/\sigma_i}, \quad (44)$$

where  $\sigma = [\sigma_1, \dots, \sigma_{512}]$  is the set of the mean values of the  $\chi^2$  distances. The  $\chi^2$  distance  $\chi_i^2(u(i), v(i))$  between  $u(i)$  and  $v(i)$  is defined as:

$$\chi_i^2(u(i), v(i)) = \frac{1}{2} \cdot \frac{|u(i) - v(i)|^2}{u(i) + v(i)}, \quad (45)$$

where  $u(i)$  and  $v(i)$  are the  $i$ th bin of the codeword histograms  $u$  and  $v$  for two image instances  $x$  and  $y$ .

### 6.3 Inter-Concept Discrimination Complexity Score

A visual concept network is constructed for organizing a large number of image concepts according to their inter-concept visual correlations. The visual concept network consists of two key components: (a) *image concepts* (i.e., object classes and scene categories); and (b) *inter-concept cumulative visual similarity contexts* between their relevant image instances.

For two given image concepts  $C_i$  and  $C_j$ , their inter-concept cumulative visual similarity context  $\gamma(C_i, C_j)$  is defined as:

$$\gamma(C_i, C_j) = \frac{1}{|C_i||C_j|} \sum_{x \in C_i} \sum_{y \in C_j} \kappa(x, y), \quad (46)$$

where  $|C_i|$  and  $|C_j|$  are the total numbers of image instances for the image concepts  $C_i$  and  $C_j$ ,  $\kappa(x, y)$  is the visual similarity context between two image instances  $x$  and





Figure 38: The visual network of the 1000 image concepts on ImageNet [33] database.

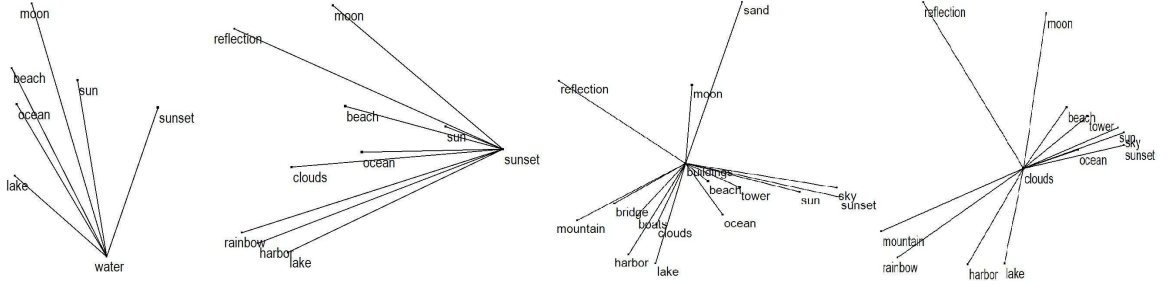


Figure 39: Some visually-related image concepts in the NUS-WIDE [25] data set.

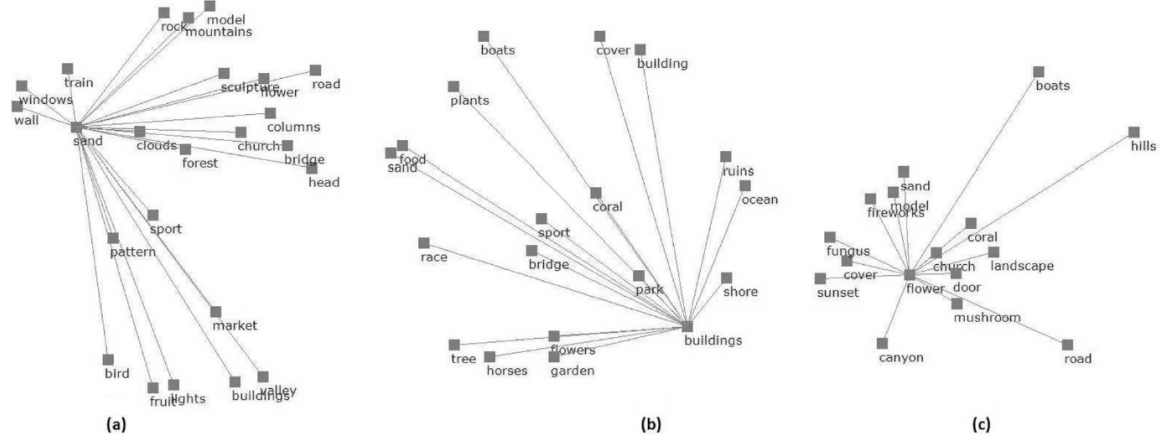


Figure 40: Some visually-related image concepts in ImageNet [33] database.

As shown in Fig. 37 and Fig. 38, the geometric closeness among the image concepts is inverse with the scales (numerical values) of their inter-concept cumulative visual similarity contexts  $\gamma(\cdot, \cdot)$ : (a) the visually-related image concepts  $\gamma(\cdot, \cdot) \neq 0$  are linked together and the visually-irrelevant image concepts  $\gamma(\cdot, \cdot) = 0$  are not linked at all; (b) the image concepts, which are closer on the visual concept network, have larger values of their inter-concept cumulative visual similarity contexts  $\gamma(\cdot, \cdot)$ ; on the other hand, the image concepts, which are far-away on the visual concept network, have smaller values of their inter-concept cumulative visual similarity contexts  $\gamma(\cdot, \cdot)$ . Thus supporting graphical representation and visualization of the visual concept network can reveal a great deal about the visual correlations among the image concepts.

The visual concept network can provide multiple advantages: (a) It can interpret



the inter-concept visual correlations explicitly as shown in Fig. 37 and Fig. 38. (b) It can provide a good environment for determining the visually-related image concepts directly in the visual feature space as shown in Fig. 39 and Fig. 40. (c) It can provide a good environment to select more effective inference models for concept classifier training, *e.g.*, integrating the image instances for multiple visually-related image concepts to learn their inter-related SVM concept classifiers jointly and training the one-against-all SVM concept classifiers independently for the isolated image concepts.

For two given image concepts  $C_i$  and  $C_j$ , if they have large value of their inter-concept cumulative visual similarity context  $\gamma(C_i, C_j)$ , their image instances will share some common or similar visual properties and there may exist significant overlapping among their concept classifiers in the visual feature space. Thus it could be hard for machine learning tools to obtain unique concept classifiers for discriminating such visually-related image concepts effectively in the visual feature space, *e.g.*, the visually-related image concepts may not be visually separable because their relevant images and concept classifiers may have significant overlapping in the visual feature space. Therefore, the image concepts, which have large values of the inter-concept cumulative visual similarity contexts  $\gamma(\cdot, \cdot)$  with many other image concepts on the visual concept network, may have large semantic gaps. On the other hand, the image concepts, which have small values or even zero values of the inter-concept cumulative visual similarity contexts  $\gamma(\cdot, \cdot)$  with other image concepts on the visual concept network (i.e., they are isolated from other image concepts in the visual feature space), may have small semantic gaps. Thus it is easy for machine learning tools to train

unique concept classifiers for discriminating the isolated image concepts (with smaller semantic gaps) from other image concepts effectively.

Given image concept  $C_l$ , two criteria can be used to quantify its inter-concept discrimination complexity score effectively: (a) the number of its visually-related image concepts on the visual concept network (some examples for the visually-related image concepts are illustrated in Fig. 39 and Fig. 40; (b) the strengths (numerical values) of the inter-concept cumulative visual similarity contexts  $\gamma(\cdot, \cdot)$  for the visually-related image concepts, *e.g.*, if two image concepts have large value of their inter-concept visual similarity context  $\gamma(\cdot, \cdot)$ , they may not be visually separable and they may have large semantic gaps.

For a given image concept  $C_l$ , its inter-concept discrimination complexity score  $\bar{\gamma}(C_l)$  is defined as the cumulative inter-concept visual similarity contexts:

$$\bar{\gamma}(C_l) = \sum_{C_j \in \Theta_l} \gamma(C_l, C_j), \quad (47)$$

where  $\Theta_l$  is a set of image concepts that are visually related with the given image concept  $C_l$  and are linked with  $C_l$  on the visual concept network,  $\gamma(C_l, C_j)$  is the strength (numerical value) of the inter-concept visual similarity context between the image concepts  $C_l$  and  $C_j$ .

If a given image concept has large value of the inter-concept discrimination complexity score  $\bar{\gamma}(\cdot)$ , it may have large semantic gap and high learning complexity for concept classifier training because the given image concept may not be visually separable from other image concepts on the visual concept network. On the other hand, if a given image concept has small value of the inter-concept discrimination complexity

score  $\bar{\gamma}(\cdot)$ , the given image concept may have small semantic gap and low learning complexity for concept classifier training because the given image concept is visually isolated from other image concepts on the visual concept network. Thus the inter-concept discrimination complexity score  $\bar{\gamma}(\cdot)$  can be used as one important factor for supporting quantitative characterization of the semantic gaps directly in the visual feature space.

#### 6.4 Inner-Concept Visual Homogeneity Score

For a given image concept  $C_l$  on the visual concept network, its inner-concept visual homogeneity score  $\Psi(C_l)$  is defined as the cumulative visual similarity contexts among all its image instances:

$$\Psi(C_l) = \frac{1}{|C_l|^2} \sum_{u \in C_l} \sum_{v \in C_l} \kappa(u, v), \quad (48)$$

where  $|C_l|$  is the total number of the images instances for the given image concept  $C_l$ ,  $\kappa(u, v)$  is the kernel-based similarity context between two image instances  $u$  and  $v$  as defined in Eq. (44).

If a given image concept has small value of the inner-concept visual homogeneity score  $\Psi(\cdot)$ , its image instances should have huge diversity on their visual properties (i.e., low inner-concept visual consistency), thus it is very hard for machine learning tools to use some simple models to approximate its diverse visual properties completely. When some complex models are used to approximate the diverse visual properties completely for the given image concept, there may not exist a unique concept classifier with high discrimination power. On the other hand, if a given image concept

has large value of the inner-concept visual homogeneity score  $\Psi(\cdot)$ , its image instances should have small diversity on their visual properties (i.e., high inner-concept visual consistency). As a result, it is much easier for machine learning tools to use some simple models to approximate its homogeneous visual properties completely and there may exist a unique concept classifier with high discrimination power. Based on these observations, the inner-concept visual homogeneity score  $\Psi(\cdot)$  can be treated as another important factor for supporting quantitative characterization of the semantic gaps directly in the visual feature space.

### 6.5 Quantitative Characterization of Semantic Gaps

For a given image concept  $C_l$  on the visual concept network, its semantic gap depends on two important factors: (1) *its inner-concept visual homogeneity score*  $\Psi(C_l)$  which is used to characterize the inner-concept visual homogeneity or inner-concept visual consistency among its relevant image instances, *e.g.*,  $\Psi(C_l)$  can be used to assess whether there exists a unique concept classifier in the visual feature space for the given image concept  $C_l$ ; (2) *its inter-concept discrimination complexity score*  $\bar{\gamma}(C_l)$  which is used to characterize its cumulative visual correlations with other image concepts on the visual concept network, *e.g.*,  $\bar{\gamma}(C_l)$  can be used to assess whether the given image concept  $C_l$  is visually separable from other image concepts in the visual feature space. It is worth noting that these two important factors are inter-related, *e.g.*, for a given image concept  $C_l$ , if it has small value of the inner-concept visual homogeneity score  $\Psi(C_l)$  (i.e., it has huge inner-concept visual diversity), it may have more opportunity to overlap with other image concepts in the visual feature space

or share some similar visual properties with other image concepts (*i.e.*, it may have large value for the inter-concept discrimination complexity score  $\bar{\gamma}(C_l)$ ).

If the given image concept  $C_l$  has large semantic gap, it may have large value for the inter-concept discrimination complexity score  $\bar{\gamma}(C_l)$  while having small value for the inner-concept visual homogeneity score  $\Psi(C_l)$  (*i.e.*,  $C_l$  has many visually-related image concepts on the visual concept network while its inner-concept visual consistency is low). On the other hand, if the given image concept  $C_l$  has small semantic gap, it may have small value for the inter-concept discrimination complexity score  $\bar{\gamma}(C_l)$  while having large value of the inner-concept visual homogeneity score  $\Psi(C_l)$  (*i.e.*,  $C_l$  has high inner-concept visual consistency while it is visually isolated from other image concepts on the visual concept network).

Based on these observations, the semantic gap for the given image concept  $C_l$  can be defined as:

$$\Upsilon(C_l) = \frac{\bar{\gamma}(C_l)}{\Psi(C_l) + \delta_0}, \quad (49)$$

where  $\delta_0$  is a constant to avoid the problem of overflow,  $\Psi(C_l)$  is the inner-concept visual homogeneity score for the given image concept  $C_l$ ,  $\bar{\gamma}(C_l)$  is the inter-concept discrimination complexity score for the given image concept  $C_l$ .

By simultaneously considering both the inner-concept visual homogeneity score and the inter-concept discrimination complexity score, the scale of the semantic gap  $\Upsilon(C_l)$  can be used to predict whether the given image concept  $C_l$  is visually separable from other image concepts in the visual feature space or whether there exists a unique concept classifier for the given image concept  $C_l$  in the visual feature space, *e.g.*,

$\Upsilon(C_l)$  can be used to estimate its learning complexity for concept classifier training. For the given image concept  $C_l$  on the visual concept network, the success for concept classifier training (i.e., whether its concept classifier can achieve high accuracy rate for automatic concept detection on test images) largely depends on the scale of its semantic gap  $\Upsilon(C_l)$ .

It is worth noting that: (1) our algorithm for supporting quantitative characterization of the semantic gaps is *data-driven* because both the inner-concept visual homogeneity score and the inter-concept discrimination complexity score are directly derived from the relevant image instances; (2) our data-driven algorithm can achieve quantitative characterization of the semantic gaps directly in the visual feature space, and the visual feature space is the common space for concept classifier training and automatic concept detection.

To assess the effectiveness and robustness of our data-driven algorithm on supporting quantitative characterization of the semantic gaps, it is very important to compare its effectiveness with other alternative approaches. Some pioneering researches have been done recently for calculating the inner-concept visual homogeneity and the inter-concept visual correlation by using the average distances [35]. Unfortunately, there is no existing approach for supporting quantitative characterization of the semantic gaps directly in the visual feature space (i.e., calculating the numerical values (scales) of the semantic gaps rather than simply guessing whether the image concepts have larger semantic gaps or not). Based on these observations, an alternative approach is developed for supporting quantitative characterization of the semantic gaps directly in the visual feature space, and it is treated as an alternative approach for effectiveness

comparison in this paper.

For a given image concept  $C_l$ , its inner-concept cumulative visual variance  $\sigma(C_l)$  is defined as:

$$\sigma(C_l) = \frac{1}{|C_l|^2} \sum_{u \in C_l} \sum_{v \in C_l} |\kappa(u, v) - \Psi(C_l)|^2, \quad (50)$$

where  $\Psi(C_l)$  is the inner-concept visual homogeneity score for the given image concept  $C_l$  as defined in Eq. (48). From this definition, one can observe that small inner-concept cumulative visual variance corresponds to large inner-concept visual homogeneity score, on the other hand, large inner-concept cumulative visual variance corresponds to small inner-concept visual homogeneity score.

If the given image concept  $C_l$  has large semantic gap, it should have large values for both the inter-concept discrimination complexity score  $\bar{\gamma}(C_l)$  and the inner-concept cumulative visual variance  $\sigma(C_l)$ . On the other hand, if the given image concept  $C_l$  has small semantic gap, it should have small values for both the inter-concept discrimination complexity score  $\bar{\gamma}(C_l)$  and the inner-concept cumulative visual variance  $\sigma(C_l)$ . Based on these observations, the semantic gap for the given image concept  $C_l$  can alternatively be defined as:

$$\Upsilon(C_l) = \lambda \cdot \bar{\gamma}(C_l) + (1 - \lambda) \cdot \sigma(C_l), \quad (51)$$

where  $\lambda = 0.6$  is a weighting factor,  $\sigma(C_l)$  is the inner-concept cumulative visual variance,  $\bar{\gamma}(C_l)$  is the inter-concept discrimination complexity score for the given image concept  $C_l$ .

The goals for supporting quantitative characterization of the semantic gaps directly

in the visual feature space are to: (1) provide a theoretical approach to estimate the learning complexity for concept classifier training; (2) provide a good environment to select effective inference models for concept classifier training which will further result in high accuracy rates on concept detection. It is worth noting that both concept classifier training and automatic concept detection are performed in the visual feature space rather than in the label space. Thus supporting quantitative characterization of the semantic gaps directly in the visual feature space plays an important role in achieving more effective concept classifier training by selecting more suitable inference models automatically.

#### 6.6 Automatic Inference Model Selection for Concept Classifier Training

Supporting quantitative characterization of the semantic gaps can allow us to estimate the learning complexity for each image concept directly in the visual feature space. With the knowledge of the learning complexity for each image concept (i.e., numerical value (scale) of the semantic gap  $\Upsilon(\cdot)$  for each image concept), more effective inference models can be selected for concept classifier training by: (a) identifying the image concepts with small semantic gaps (i.e., the isolated image concepts with good inner-concept visual consistency) and training their one-against-all SVM concept classifiers independently; (b) determining the image concepts with large semantic gaps (i.e., the visually-related image concepts with low inner-concept visual consistency) and integrating their image instances to train their inter-related SVM concept classifiers jointly; and (c) using more image instances to achieve more reliable training of the concept classifiers for the image concepts with large semantic gaps.



To bridge the semantic gaps, a structural learning algorithm is developed for concept classifier training, where both the scales of the semantic gaps  $\Upsilon(\cdot)$  and the visual concept network are used to determine the inter-related learning tasks directly in the visual feature space and select more effective inference models for concept classifier training. As compared with traditional structural SVM algorithm [14], our structural learning algorithm leverages the inter-concept visual correlations for training multiple inter-related concept classifiers jointly rather than simply performing structural output regression in the label space. As compared with traditional multi-task boosting algorithm [147], our structural learning algorithm leverages both the visual concept network and the scales of the semantic gaps for inter-task relatedness modeling and automatic inference model selection rather than simply performing concept combinations.

For a given image concept  $C_j$  on the visual concept network, its SVM classifier is defined as:

$$f_{C_j}(x) = W_j^{tr} \Phi_j(x) + \sum_{C_t \in \Theta_j} \gamma_t \cdot V_t^{tr} \Phi_t(x), \quad \sum_{C_t \in \Theta_j} \gamma_t = 1, \quad (52)$$

where  $\Theta_j$  is used to represent a set of image concepts that are visually-related with the given image concept  $C_j$  (they are linked with the given image concept  $C_j$  on the visual concept network),  $W_j$  is a *self regularization term* that is used to represent the contribution of the  $C_j$ 's image instances on the  $C_j$ 's SVM classifier  $f_{C_j}(x)$ ,  $V_t$  is the *inter-concept regularization term* that is used to represent the contribution of the  $C_t$ 's image instances on the  $C_j$ 's SVM classifier  $f_{C_j}(x)$ ,  $\gamma_t$  is the weight factor to interpret how much the  $C_t$ 's image instances can contribute on the  $C_j$ 's SVM classifier  $f_{C_j}(x)$ ,

$\Phi_j(x)$  and  $\Phi_t(x)$  are the mapping functions from the visual feature vector  $x$  to some other Euclidean space  $\mathbf{H}$ .

If the given image concept  $C_j$  is visually-related with the image concept  $C_t$  (i.e.,  $C_j$  is linked with  $C_t$  on the visual concept network),  $V_t \neq 0$ . If the given image concept  $C_j$  is visually-irrelevant with the image concept  $C_t$  (i.e.,  $C_j$  is not linked with  $C_t$  on the visual concept network),  $V_t = 0$ .

By integrating both the visual concept network and the scales of the semantic gaps for automatic inference model selection, our structural learning algorithm can achieve more effective classifier training by minimizing a joint objective function  $J$ .

$$J = \frac{1}{2}(\|W_j\|^2 + \sum_{t=1}^{|\Theta_j|} \lambda_t \|V_t\|^2) + \rho_0 \sum_{t=1}^{|\Theta_j|} \sum_{i=1}^{n_j} \xi_{ti} + \sum_{t=1}^{|\Theta_j|} \rho_t \sum_{i=1}^{n_t} \eta_{ti}, \quad (53)$$

where  $|\Theta_j|$  is the size of  $\Theta_j$ ,  $\xi_{ti} \geq 0$  and  $\eta_{ti} \geq 0$  are the error rates,  $\rho_0$  and  $\rho_t$  are the weighting factors for controlling the error penalty,  $n_j$  and  $n_t$  are the total number of image instances for the image concepts  $C_j$  and  $C_t$ ,  $\lambda_t$  is the weighting factor that is used to control the contributions of the  $C_t$ 's image instances on the  $C_j$ 's concept classifier  $f_{C_j}(x)$ .

By integrating the image instances for multiple visually-related image concepts  $\Omega = \{(x_{it}, y_{it}) | i = 1, \dots, n; t = 1, \dots, |\Theta_j|\}$  to solve the joint objective function as defined in Eq. (53), the SVM classifier for the given image concept  $C_j$  can be determined as:

$$\begin{aligned} f_{C_j}(x) = & \sum_{h,t=1}^{|\Theta_j|} \gamma_t \kappa_s(t, h) \left( \sum_{i=1}^{n_j} \beta_{hi} \kappa(x_{ji}, x) - \sum_{i=1}^{n_h} \bar{\beta}_{hi} \kappa(x_{hi}, x) \right) \\ & + \sum_{t=1}^{|\Theta_j|} \frac{\gamma_t}{\lambda_t} \left( \sum_{i=1}^{n_j} \beta_{ti} \kappa(x_{ji}, x) - \sum_{i=1}^{n_t} \bar{\beta}_{ti} \kappa(x_{ti}, x) \right), \end{aligned} \quad (54)$$

where  $\beta$  and  $\bar{\beta}$  are two different sets of the weights for the image instances,  $\kappa_s(t, h)$

is the semantic kernel for characterizing the semantic similarity context between the image concepts  $C_t$  and  $C_h$ ,  $\kappa(\cdot, \cdot)$  is the visual kernel for characterizing the visual similarity context between the image instances as defined in Eq. (44).

Our structural learning algorithm can significantly enhance the discrimination power of the concept classifiers by: (a) training the one-against-all SVM classifiers independently for the image concepts with small semantic gaps (i.e., the isolated image concepts with good inner-concept visual consistency) by automatically setting  $V_t = 0$  in Eq. (52); (b) training the inter-related SVM classifiers jointly for multiple visually-related image concepts with large semantic gaps (i.e., multiple visually-related image concepts with low inner-concept visual consistency) by automatically setting  $V_t \neq 0$  in Eq. (52); (c) learning from the image instances for other visually-related image concepts to enhance the generalization ability of the concept classifiers on test images, which may somewhat reduce the required sizes of the image instances for achieving reliable training of the concept classifiers for the image concepts with large semantic gaps.

## 6.7 Algorithm Evaluation and Experimental Results

Our experiments on algorithm evaluation are performed on two well-known image sets: NUS-WIDE [25] and ImageNet [33]. For a given image concept, our algorithm first calculates the scale (numerical value) of its semantic gap by using two alternative approaches as defined in Eqs. (49) and (51). The image concepts with small semantic gaps and the image concepts with large semantic gaps are then identified automatically according to the scales of their semantic gaps  $\Upsilon(\cdot)$ . The learning complexities

Table 15: Image concepts with small semantic gaps in NUS-WIDE [25] data set.

clouds	sky	ocean	grass	zebra	plane	airport	rocks
animal	map	soccer	vehicle	window	mountain	valley	water
flower	sun	sunset	tiger	buildings	reflection	lake	tree
whales							

Table 16: Image concepts with large semantic gaps in NUS-WIDE [25] data set.

wedding	earthquake	tattoo	statue	fox	toy	running	dancing
temple	book	bridge	glacier	castle	fire	protest	police
protest	flags	cars	town				

for concept classifier training are high for the image concepts with large semantic gaps, on the other hand, the learning complexities for concept classifier training are low for the image concepts with small semantic gaps.

#### 6.7.1 Experimental Results for NUS-WIDE Image Set

As mentioned above, a data-driven algorithm is developed for supporting quantitative characterization of the semantic gaps directly in the visual feature space, *e.g.*, calculating the numerical values (scales) of the semantic gaps for the image concepts. Thus our data-driven algorithm can automatically identify both the image concepts with small semantic gaps and the image concepts with large semantic gaps, and some experimental results are given in Table 15 and Table 16 for the NUS-WIDE [25] image set.

After the concept classifiers are obtained for all these 81 image concepts in NUS-WIDE image set, they are further used for detecting the image concepts from test images. Ideally, if an image concept has large semantic gap, its learning complexity for concept classifier training is high. As a result, the accuracy rates for detecting the image concepts with large semantic gaps may be low when the same sizes of image

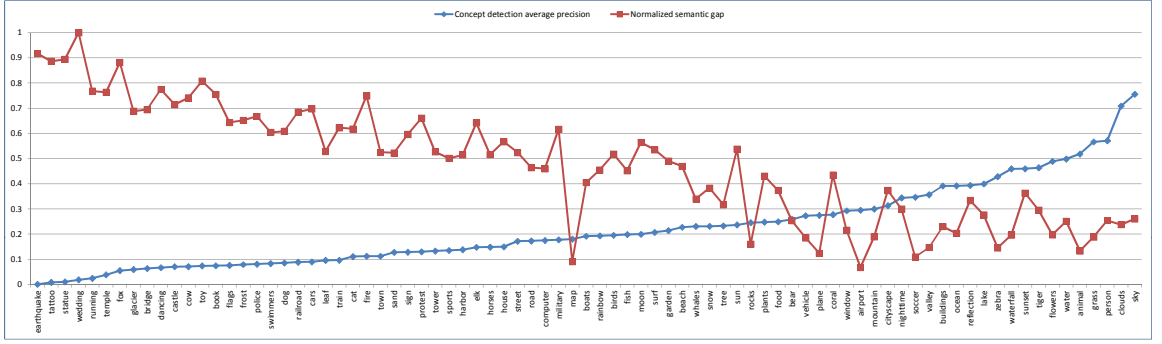


Figure 41: The consistency between the numerical values of semantic gaps  $\Upsilon(\cdot)$  for learning complexity estimation and the accuracy rates for automatic concept detection in NUS-WIDE image set.

instances are used for concept classifier training. Thus there is a good consistency between the scales (numerical values) of the semantic gaps, the strengths of the learning complexities for concept classifier training, and the accuracy rates of the concept classifiers on automatic concept detection. As shown in Table 17 and Fig. 41, our experiments have obtained good evidences for this consistency (*e.g.*, good consistency between the scales of the semantic gaps and the accuracy rates of the concept classifiers on automatic concept detection).

For the image concept “Map” in NUS-WIDE [25], a small semantic gap is obtained but the detection accuracy rate is very low rather than high. The reason for this phenomenon is that NUS-WIDE image set contains a small number of the relevant images for the image concept “Map”, which cannot sufficiently characterize both the inner-concept visual diversity for the image concept “Map” and its inter-concept visual correlations with other image concepts.

We have also compared our algorithm for quantitative characterization of the semantic gaps with the approach developed by Lu *et al.* [99, 100]. Because Lu’s ap-

Table 17: The consistency between the numerical values (scales) of semantic gaps  $\Upsilon(\cdot)$  for learning complexity estimation and the accuracy rates for automatic concept detection.

Image concept	Semantic gap	Accuracy rate	Image concept	Semantic gap	Accuracy rate
sky	0.7557	0.5609	sun	0.337	0.8381
clouds	0.7088	0.6079	tree	0.433	0.6169
person	0.5711	0.6547	snow	0.4315	0.6834
grass	0.5663	0.689	whales	0.4311	0.64
animal	0.5183	0.7337	beach	0.4276	0.7698
water	0.4986	0.7514	garden	0.4642	0.6902
flowers	0.4892	0.7975	surf	0.5072	0.535
tiger	0.5636	0.695	moon	0.4997	0.6644
sunset	0.4599	0.7635	fish	0.4988	0.5521
waterfall	0.4592	0.7967	birds	0.4955	0.6179
zebra	0.5284	0.6458	rainbow	0.4934	0.6554
lake	0.4994	0.5763	boats	0.4921	0.6053
reflection	0.5938	0.534	map	0.1806	0.0913
ocean	0.492	0.6037	military	0.4777	0.6169
buildings	0.4916	0.6301	computer	0.4752	0.5599
valley	0.4569	0.6472	road	0.4732	0.5646
soccer	0.5468	0.5095	street	0.4719	0.5539
nighttime	0.6441	0.5086	house	0.4498	0.5671
cityscape	0.5131	0.5737	horses	0.4487	0.5159
mountain	0.4999	0.6897	elk	0.448	0.6421
airport	0.4953	0.679	harbor	0.5385	0.5146
window	0.4929	0.6157	sports	0.4353	0.5315
coral	0.4778	0.6348	tower	0.4331	0.5269
plane	0.4747	0.5243	protest	0.4301	0.6107
vehicle	0.4732	0.5858	sign	0.5287	0.5962
bear	0.4591	0.5536	sand	0.4279	0.6227
food	0.4495	0.5741	fire	0.4126	0.7491
plants	0.448	0.6308	town	0.5126	0.5849
rocks	0.4457	0.7604	cat	0.5111	0.6181

proach focuses on determining the image concepts with smaller or larger semantic gaps rather than calculating the numerical values of their semantic gaps, we just provide our experimental results according to the results presented in Lu's papers [99, 100]. As shown in Table 18, our data-driven algorithm has obtained very competitive results, where the image concepts with high confidence scores (small semantic gaps as defined by Lu's method) are selected from Fig. 41 in Lu's paper [100] and our algorithm is used to calculate the numerical values of their semantic gaps directly

Table 18: Comparison on image concepts with small semantic gaps.

Image concept	Score in [100]	Semantic gap	Image concept	Score in [100]	Semantic gap
sunset	0.092	0.4599	cloud(s)	0.012	0.7088
flower(s)	0.057	0.4892	water	0.019	0.4986
sky	0.032	0.7557	garden	0.02	0.4902
tree	0.024	0.3169	beach	0.022	0.4698

in the visual feature space. One can observe that good consistency between the semantic similarity contexts among the associated text terms and the visual similarity contexts among the relevant images (high confidence scores) does not always correspond to small semantic gaps (small numerical values for the semantic gaps  $\Upsilon(\cdot)$  in the visual feature space). For some image concepts, our data-driven algorithm has obtained much better results than Lu’s approach because the associated text terms may consist of rich word vocabulary rather than only the auxiliary text terms for image semantics description. When all these auxiliary text terms are loosely used for characterizing the semantics of the social images, it is very hard if not impossible to obtain semantic consistency among the auxiliary text terms. For some image concepts, our data-driven algorithm has obtained similar results with Lu’s approach because the auxiliary text terms have good semantic consistency.

To assess the effectiveness and robustness of our data-driven algorithm on supporting quantitative characterization of the semantic gaps, we have compared the scales (numerical values) of the semantic gaps which are calculated by using two alternative approaches. As shown in Fig. 42, one can observe that two alternative approaches have obtained good consistency on supporting quantitative characterization of the semantic gaps, *e.g.*, for any two image concepts, the image concept with larger semantic gap will always have larger semantic gap under two alternative approaches, on

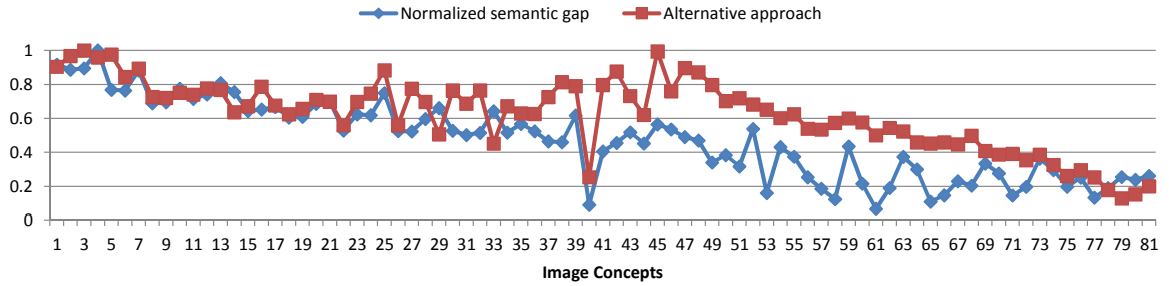


Figure 42: Comparison between two alternative approaches for supporting quantitative characterization of the semantic gaps (*i.e.*, numerical values of the semantic gaps  $\Upsilon(\cdot)$ ).

the other hand, the image concept with smaller semantic gap will still have smaller semantic gap under two alternative approaches. Thus our experimental results have demonstrated good evidences and rigorous justifications of the effectiveness and robustness of our data-driven algorithms on supporting quantitative characterization of the semantic gaps directly in the visual feature space.

### 6.7.2 Experimental Results for ImageNet Image Set

For the ImageNet [33] data set, its visual concept network is shown in Fig. 38 and some examples for the inter-related image concepts are shown in Fig. 40. The image concepts with small semantic gaps and the image concepts with large semantic gaps are identified automatically according to the scales (numerical values) of their semantic gaps  $\Upsilon(\cdot)$ , and some experimental results are shown in Table 19 and Table 20.

After the concept classifiers are obtained for all these 1,000 image concepts in ImageNet [33] data set, they are further used for detecting the image concepts from test images. When the same sized of image instances are used for concept classifier training, the accuracy rates for detecting the image concepts with large semantic gaps



Table 19: Image concepts with small semantic gaps for ImageNet [33] data set.

blue sky	water	red rose	sunset	cup	moon
missile	beach	snow	tree	glass	garden
sunset	yellow flower	car-front	car-wheel	ocean	cloud
stone-rock	red flower	watermelon	banana	licence plate	church
purple flower	white flower	sand	screen	firework	falls
leaf	butterfly	stop sign	coffee machine	coffee pot	mug
traffic light	bottle	bin	can	...	...

Table 20: Image concepts with large semantic gaps for ImageNet [33] data set.

flower	ring	fish	dog	car	crab
rail	park	gas station	gun	bank	wave
mountain	street	window	horse	building	bike
dolphin	tiger	...	...	...	...

may be low. Thus there is a good consistency between the scales (numerical values) of the semantic gaps, the strengths of the learning complexities for concept classifier training, and the accuracy rates of the concept classifiers on automatic concept detection. As shown in Fig. 43, our experiments have obtained good evidences for this consistency (*e.g.*, good consistency between the scales of the semantic gaps and the accuracy rates of the concept classifiers on automatic concept detection).

It is worth noting that our algorithm for supporting quantitative characterization of the semantic gaps is a data-driven approach, thus it is very attractive to assess its dependence with various image sets, *e.g.*, whether the scales (numerical values) of the semantic gaps for the same image concepts may vary with the image sets. As shown in Table 21, we have compared the scales (numerical values) of the semantic gaps  $\Upsilon(\cdot)$  for the same image concepts in two well-known image sets: NUS-WIDE and ImageNet. From these experimental results, one can observe that the scales (numerical values) of the semantic gaps for the same image concepts may vary with

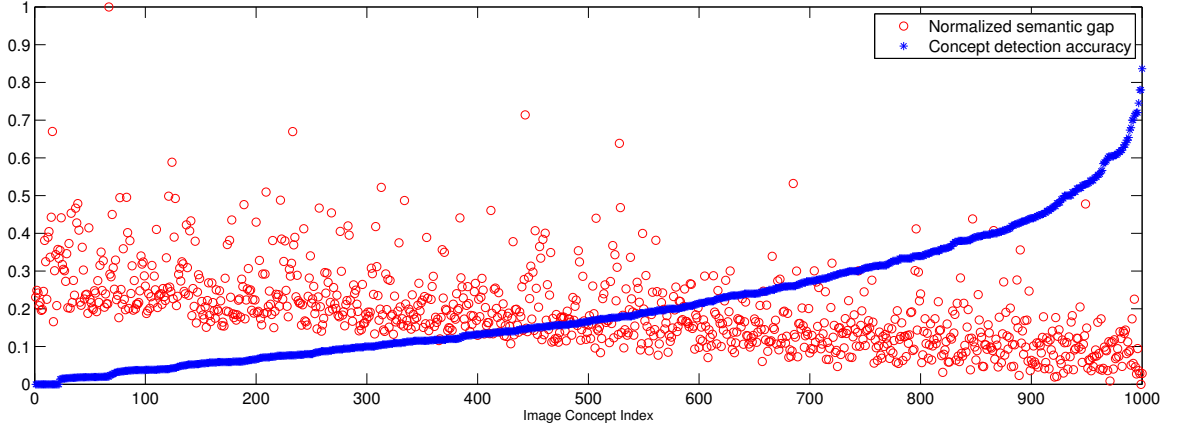


Figure 43: The consistency between the numerical values (scales) of semantic gaps  $\Upsilon(\cdot)$  for learning complexity estimation and the accuracy rates for automatic concept detection in ImageNet data set.

the image sets, but the trends of the semantic gaps are consistent, *e.g.*, for any two image concepts with different semantic gaps, the one with larger semantic gap will always have larger values of the semantic gaps in two image sets and the other with smaller semantic gap will always have smaller values of these semantic gaps in two image sets.

To evaluate the effectiveness of image representation and similarity function and their influences on the effectiveness and robustness of our data-driven algorithm for quantitative characterization of the semantic gaps, three approaches are used for image representation and similarity characterization: (a) kernel function as defined in Eq. (44) which is based on the  $\chi^2$  distances between the codewords; (b) Mahalanobis distance function where Mahalanobis distance is used to replace the  $\chi^2$  distances in Eq. (44); and (c) kernel function between the cumulative codeword histograms. This study focuses on assessing the consistency of the effectiveness and robustness of our data-driven algorithm for quantitative characterization of the semantic gaps when different approaches are used for image representation and similarity charac-

Table 21: Comparison on the scales of semantic gaps for different image sets.

Image concept	Semantic gap on NUS-WIDE	Semantic gap in ImageNet
sky	0.7557	0.7802
clouds	0.7088	0.7358
grass	0.5663	0.6132
water	0.4986	0.5202
flowers	0.4892	0.4925
sunset	0.4599	0.4803
waterfall	0.4592	0.4789
zebra	0.4284	0.4559
lake	0.3994	0.4230
ocean	0.392	0.4185
buildings	0.3916	0.4201
mountain	0.2999	0.3260
airport	0.2953	0.3138
tree	0.233	0.2900
snow	0.2315	0.2818
beach	0.2276	0.2498
garden	0.2142	0.2530
moon	0.1997	0.2180
rainbow	0.1934	0.2120
boats	0.1921	0.2018
sand	0.1279	0.1468
train	0.0966	0.1260
flags	0.7763	0.7933
book	0.7486	0.7602
toy	0.7440	0.7579
cow	0.7133	0.7411
castle	0.7090	0.7347
bridge	0.6370	0.6847
glacier	0.5962	0.6282
fox	0.5540	0.5816
temple	0.6388	0.6738

terization. As shown in Fig. 44 and Fig. 45, one can observe that our data-driven algorithm has good consistency for supporting quantitative characterization of the semantic gaps when different approaches are used for image representation and similarity characterization, *e.g.*, for any two image concepts (one has smaller semantic gap and another has bigger semantic gap), the image concept with smaller semantic gap will always have smaller semantic gap under different distance functions for simi-

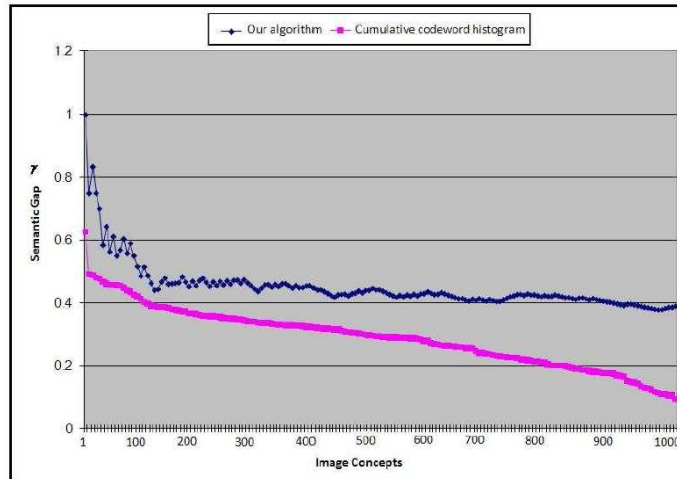


Figure 44: The consistency on the trend of the semantic gaps under different similarity functions: our algorithm using kernel function *versus* cumulative codeword histograms.

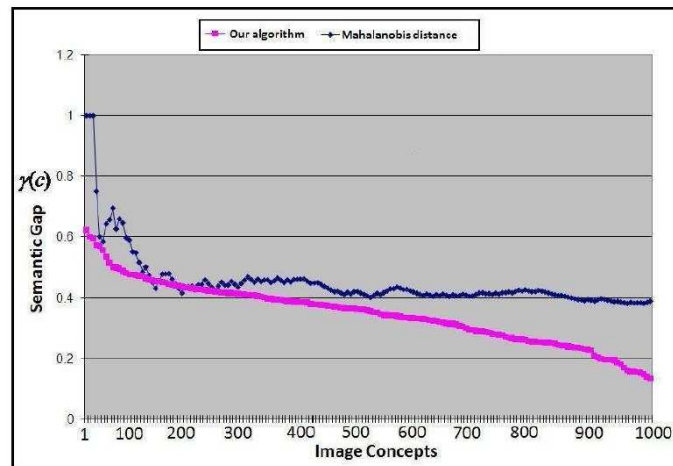


Figure 45: The consistency on the trend of the semantic gaps under different similarity functions: our algorithm using kernel function *versus* Mahalanobis distance.

larity characterization, on the other hand, the image concept with larger semantic gap will always have larger semantic gap under different distance functions for similarity characterization.

### 6.7.3 Benefits from Semantic Gap Quantification

The goal for supporting quantitative characterization of the semantic gaps is to provide a theoretical approach for: (a) estimating the learning complexity for concept classifier training directly in the visual feature space; and (b) selecting more effective inference models for concept classifier training. In order to evaluate the benefits of semantic gap quantification on concept classifier training, we have compared three approaches for concept classifier training: (1) our structural learning algorithm which leverages the inter-concept visual correlations directly in the visual feature space for automatic inference model selection, *e.g.*, leveraging both the scales (numerical values) of the semantic gaps and the visual concept network for automatic inference model selection; (2) traditional structural SVM algorithm which performs structural output regression by leveraging the inter-label (inter-concept) semantic correlations in the label space; (3) traditional multi-task boosting algorithm which leverages the inter-concept visual correlations via simple concept combinations.

For the NUS-WIDE image set, the comparison results on the detection accuracy rates for some image concepts are given in Fig. 46. For the ImageNet image set, the comparison results on the detection accuracy rates are given in Fig. 47. In our structural learning algorithm, both the scales (numerical values) of the semantic gaps and the visual concept network are leveraged to: (a) determine the inter-related learning tasks (*i.e.*, the learning tasks for the visually-related image concepts) directly in the visual feature space; (b) select more effective inference models for concept classifier training. On the other hand, the traditional structural SVM algorithm [14]

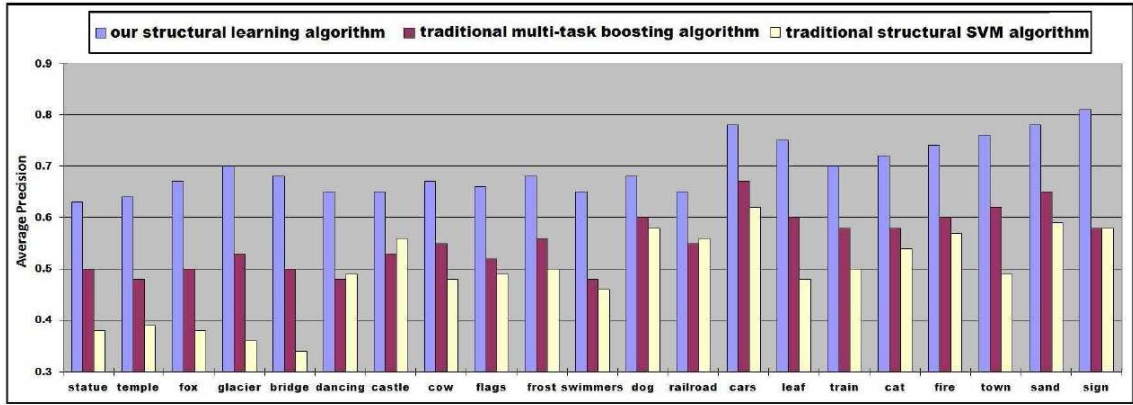


Figure 46: Performance comparison on the accuracy rates for automatic concept detection for NUS-WIDE image set: our structural learning algorithm, traditional structural SVM algorithm, traditional multi-task boosting algorithm.

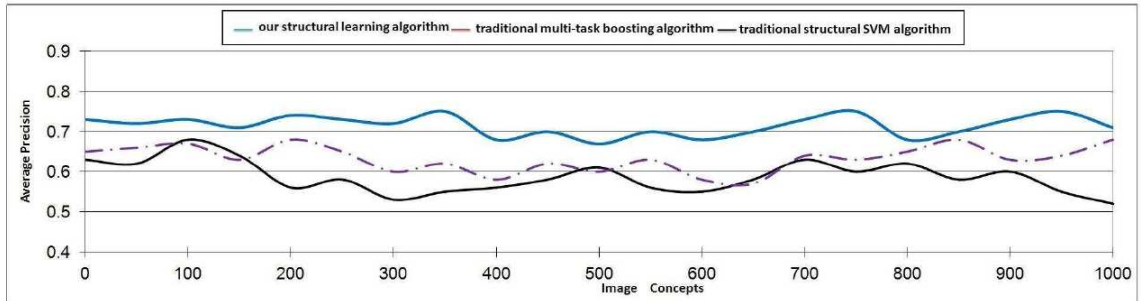


Figure 47: Performance comparison on the accuracy rates for automatic concept detection for ImageNet data set: our structural learning algorithm, traditional structural SVM algorithm, traditional multi-task boosting algorithm.

leverages the inter-concept semantic correlations in the label space via structured output regression and the traditional multi-task boosting algorithm uses simple concept combinations to exploit the inter-concept visual correlations.

The visual feature space is the common space for concept classifier training and automatic concept detection, thus characterizing the inter-concept correlations (inter-task relatedness) directly in the visual feature space and leveraging such inter-concept visual correlations for concept classifier training can significantly improve the accuracy rates of the concept classifiers on automatic concept detection. Using simple concept

combinations for modeling the inter-task relatedness may seriously suffer from the problem of huge computational complexity: there are  $2^n$  combinations for  $n$  image concepts. In addition, not all the image concepts are visually-related and simply combining the visually-irrelevant image concepts for joint concept classifier training may decrease their performance rather than improvement [43]. On the other hand, the benefits from performing structural output regression in the label space could be limited because both concept classifier training and automatic concept detection are performed in the visual feature space rather than in the label space. As shown in Fig. 46 and Fig. 47, one can observe that our structural learning algorithm can obtain higher detection accuracy rates for automatic concept detection as compared with traditional multi-task boosting algorithm and structural SVM algorithm.

In our structural learning algorithm, the visual concept network is used to determine the inter-related learning tasks automatically (*i.e.*, determine the visually-related image concepts directly in the visual feature space) and the scales of the semantic gaps are used to estimate the learning complexity and select more effective inference models for concept classifier training. To assess whether supporting quantitative characterization of the semantic gaps contributes on concept classifier training or not, we have implemented a new structural SVM algorithm, where structural output regression is performed over the visual concept network (inter-concept visual contexts in the visual feature space) rather than over the inter-label semantic contexts in the label space. As shown in Fig. 48, our structural learning algorithm can obtain the concept classifiers with higher accuracy rates on automatic concept detection as compared with this new structural SVM algorithm.

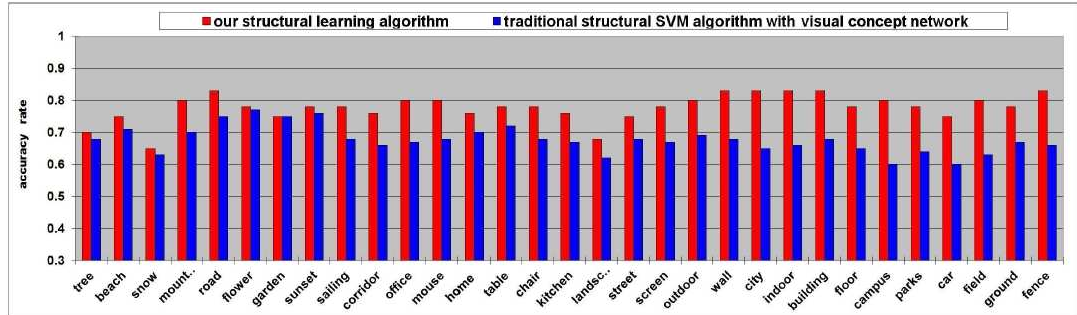


Figure 48: Performance comparison on the accuracy rates for automatic concept detection for ImageNet image set: our structural learning algorithm by using both the scales of the semantic gaps and the visual concept network for inference model selection *versus* traditional structural SVM algorithm by performing structural output regression over the visual concept network.

The goal for concept classifier training is to find a concept classifier with low generalization error on test images. Using more image instances for concept classifier training may usually improve the generalization ability of the concept classifiers and result in low generalization error rates on test images, but the sizes of the training image instances may significantly vary with the image concepts and largely depend on the scales (values) of their semantic gaps. To achieve the same accuracy rates for automatic concept detection, more training image instances should be used to train the concept classifiers for the image concepts with larger semantic gaps, but less training image instances can be used to train the concept classifiers with small semantic gaps. As shown in Fig. 49, our experimental results have demonstrated this phenomenon. From these experimental results, one can observe: (a) When the sizes of the training image instances are small, increasing the numbers of the training image instances may significantly improve the accuracy rates of the concept classifiers for automatic concept detection. When the sizes of the training image instances are large enough, adding more training image instances cannot obtain significant improvement on the



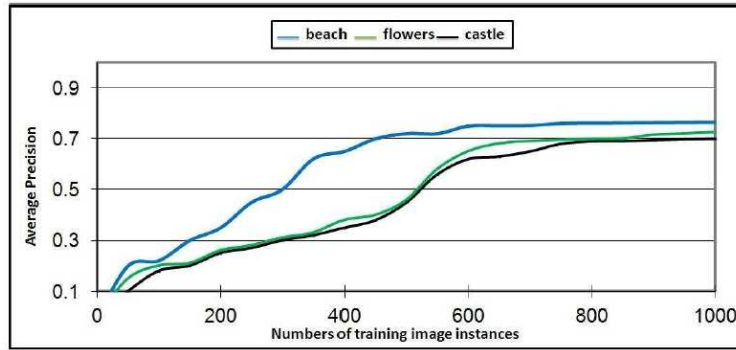


Figure 49: Our experimental results on the correlation among the accuracy rates for concept detection, the scales of semantic gaps ( $\Upsilon(\text{castle}) > \Upsilon(\text{flower}) > \Upsilon(\text{beach})$ ), and the sizes of training image instances for concept classifier training.

accuracy rates of the concept classifiers for automatic concept detection. (b) When the image concepts have larger semantic gaps, more training image instances are needed to train their concept classifiers to achieve the same accuracy rates on automatic concept detection as compared with the image concepts with smaller semantic gaps.

## 6.8 Summary

In this chapter, we presents a data-driven method to quantitatively estimate the semantic gaps of visual concepts, where both the inner-concept visual homogeneity scores and the inter-concept discrimination complexity scores are seamlessly integrated for the estimation directly in the visual feature space. A quantitative characterization of the semantic gap can allow us to estimate the learning complexity for each image concept and select more effective inference models for concept classifier training. Our experimental results on a large number of image concepts have obtained very promising results. Our future work will focus on cleansing large-scale Internet images for assessing the effectiveness and robustness of our data-driven algorithm on

supporting quantitative characterization of the semantic gaps.

## CHAPTER 7: CONCLUSION

Visual recognition is one of the fundamental problems in computer vision, and recent research efforts have been focusing on solving it in large-scale. In this dissertation, we have studied the problem of large-scale visual recognition in various aspects. In particular, we have considered more efficiently harvesting labeled image data from the Web, making visual recognition models faster, learning more effective visual dictionary for image content representation and characterizing the semantic gaps of different visual concepts more accurately. In this chapter, we first summarize our contributions and conclusions in previous chapters, and then discuss some prospective directions for further research.

### 7.1 Summary of Conclusions

In Chapter 3, we have developed an automatic image-text alignment algorithm to align Web images with their most relevant auxiliary text terms or phrases. One of the main purposes is to collect a large number of images with semantic labels from the Web for classifier training. Specifically, we have harvested a large number of cross-media web-pages which contain Web images and their auxiliary texts, and then segmented them into a collection of image-text pairs. Second, Web images have been clustered into a number of groups using near-duplicate image clustering techniques according to their visual similarity. Finally, we have performed the random walk

over a newly proposed phrase correlation network to achieve more precise image-text alignment by refining the relevance scores between Web images and their auxiliary text terms or phrases. Experimentally, we have shown that the proposed approach can effectively collect images with reliable labels using very limited human supervision.

In Chapter 4, we have proposed a visual tree model to reduce the computational complexity of a large-scale visual recognition system. The visual tree was constructed automatically for organizing a large number of image categories hierarchically according to their inter-category visual correlations. The biggest advantage of the visual tree model lies in the fact that the construction of its tree structure is very efficient compared to previous tree methods which usually require training many classifiers in advance. Our experimental results have demonstrated that the proposed visual tree model can achieve very competitive results on both the categorization accuracy and the computational speedup as being compared with other tree models.

In Chapter 5, we have proposed a joint dictionary learning (JDL) algorithm to learn more discriminative dictionaries by explicitly separating the common visual atoms from the category-specific ones. For a group of visually correlated classes, a common dictionary and multiple class-specific dictionaries are simultaneously modeled in JDL to enhance their discrimination power. The processes of learning the common dictionary and multiple class-specific dictionaries have been formulated as a joint optimization by adding a discrimination promotion term based on the Fisher discrimination criterion. To make the JDL affordable in large-scale applications, we have considered using the visual tree method described in Chapter 4 as well as the label tree [9] model to cluster a large number of image categories into a set of disjoint

groups. The process of image category clustering not only ensures that the categories in the same group are of strong visual correlation, but also makes the JDL algorithm to be computationally affordable in large-scale visual recognition applications. Accordingly, we have developed three schemes to take the advantage of the discriminative dictionaries learned by JDL for image content representation, classifier training and image classification. Our experimental results have demonstrated that the proposed JDL algorithm is superior to many unsupervised and supervised dictionary learning algorithms, especially on dealing with visually similar categories.

In Chapter 6, we have presented a data-driven method to quantitatively estimate the semantic gaps of visual concepts, where both the inner-concept visual homogeneity scores and the inter-concept discrimination complexity scores are seamlessly integrated for the estimation directly in the visual feature space. A quantitative characterization of the semantic gap can allow us to estimate the complexities of learning different visual concept classifiers, and to select more effective inference models for the training. Our experimental results on a large number of visual concepts have demonstrated the effectiveness of our method.

## 7.2 Prospective Directions for Further Research

There are many ways to further extend the methods elaborated in this dissertation. In this section, we detail a number of specific ideas for future research based on the findings of this dissertations.

**Feature Selection in Visual Tree Construction:** In the visual tree model described in Chapter 4, the visual similarities between image categories are computed only based

on the BoW features where the HoG features are used as the local features. However, in an ideal visual tree, the features used to split the tree nodes at different levels are usually different. It is worth incorporating feature selection into the tree construction process. The challenge, here, is that one has to select more effective features in an unsupervised manner since the splitting process of a tree node is often unsupervised. Domain knowledge would be helpful for feature selection.

**Trade-off between Speedup and Accuracy in Tree Models:** The visual tree presented in Chapter 4 is built in a naive way that a hard partition of image categories is adopted. Hard partition usually leads to higher speedup of inference since the image categories are only allowed to be present in one particular child node. The drawback here is that if we make a mistake on high-level nodes, we have no chance to correct it. The soft-prediction was proposed in Chapter 4 to alleviate this issue, but the effectiveness is limited. Another possible way to compensate the accuracy degradation incurred by the hard partition is to allow the child nodes of the same parent node to share some common but hard-separated classes. This can be implemented in either one single tree or multiple complementary trees. Here, the difficulty is to find a principle way to guarantee the convergence of the tree model while allow classes overlapping across sibling nodes.

**Image-based Discriminative Dictionary Learning:** Most recently proposed supervised dictionary learning algorithms including the one presented in Chapter 5 aim to learn visual words that can distinguish the class labels of local features, *e.g.* SIFT or HoG descriptors, while the true objective is to be able to discriminate images or object regions as a whole based on their distribution of visual word assignments of the

local patches sampled from them. Local visual features from the same class are not necessary similar in terms of appearance as an object itself contains a lot of different texture patterns. What is visually similar is the distribution of visual atoms encoding the local features of the same category. How to incorporate the classification loss at image or object level into the local feature dictionary training process is attractive and might lead better object recognition performance. This is a possible research direction of our future work.

## REFERENCES

- [1] M. Aharon, M. Elad, and A. Bruckstein. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, 2006.
- [2] N. Ahuja and S. Todorovic. Learning the taxonomy and models of categories present in arbitrary images. In *Proc. IEEE Conf. on Computer Vision*, pages 1–8, Oct. 2007.
- [3] M. Aly, M. Munich, and P. Perona. Multiple dictionaries for bag of words large scale image search. In *Int’l Conf. on Image Processing*, Sept. 2011.
- [4] Z. Bar-Yossef and S. Rajagopalan. Template detection via data mining and its applications. In *WWW*, pages 580–591, 2002.
- [5] K. Barnard, P. Duygulu, D. A. Forsyth, N. de Freitas, D. M. Blei, and M. I. Jordan. Matching words and pictures. *Journal of Machine Learning Research*, 3:1107–1135, 2003.
- [6] E. Bart, I. Porteous, P. Perona, and M. Welling. Unsupervised learning of visual taxonomies. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1–8, June 2008.
- [7] H. Bay, T. Tuytelaars, and L. J. V. Gool. Surf: Speeded up robust features. In *Proc. European Conf. on Computer Vision*, pages 404–417, 2006.
- [8] P. R. Beaudet. Rotationally invariant image operators. In *Proc. of Int’l Conf. on Pattern Recognition*, pages 579–583, Kyoto, Japan, Nov. 1978.
- [9] S. Bengio, J. Weston, and D. Grangier. Label embedding trees for large multi-class tasks. In *Proc. Advances in Neural Information Processing Systems*, pages 163–171. 2010.
- [10] T. L. Berg, A. C. Berg, J. Edwards, and D. A. Forsyth. Whos in the picture. In *Proc. Advances in Neural Information Processing Systems*, 2004.
- [11] A. Bergamo and L. Torresani. Meta-class features for large-scale object categorization on a budget. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 3085–3092, June 2012.
- [12] I. Biederman. Recognition-by-components: A theory of human image understanding. *Psychological Review*, 94:115–147, 1987.
- [13] J. M. Bioucas-Dias and M. A. T. Figueiredo. A new twist: Two-step iterative shrinkage/thresholding algorithms for image restoration. *IEEE Transactions on Image Processing*, 16(12):2992–3004, 2007.



- [14] M. B. Blaschko and C. H. Lampert. Learning to localize objects with structured output regression. In *Proceedings of European Conference on Computer Vision*, 2008.
- [15] L. Bo, X. Ren, and D. Fox. Multipath sparse coding using hierarchical matching pursuit. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 660–667, 2013.
- [16] L. Bottou and O. Bousquet. The tradeoffs of large scale learning. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems*, volume 20, pages 161–168. 2008.
- [17] Y.-L. Boureau, F. Bach, Y. LeCun, and J. Ponce. Learning mid-level features for recognition. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 2559–2566, 2010.
- [18] Y.-L. Boureau, J. Ponce, and Y. LeCun. A theoretical analysis of feature pooling in visual recognition. In *Intl Conf. Machine Learning*, pages 111–118, 2010.
- [19] L. Breiman. *Bagging Predictors*, volume 24. 1996.
- [20] D. Cai, X. He, Z. Li, W.-Y. Ma, and J.-R. Wen. Hierarchical clustering of WWW image search results using visual, textual and link information. In *ACM Multimedia*, pages 952–959, 2004.
- [21] E. Y. Chang, K. Goh, G. Sychay, and G. Wu. Cbsa: content-based soft annotation for multimodal image retrieval using Bayes point machines. *IEEE Trans. Circuits Syst. Video Techn.*, 13(1):26–38, 2003.
- [22] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman. The devil is in the details: an evaluation of recent feature encoding methods. In *British Machine Vision Conference*, 2011.
- [23] S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM Rev.*, 43(1):129–159, Jan. 2001.
- [24] C.-K. Chiang, C.-H. Duan, S.-H. Lai, and S.-F. Chang. Learning component-level sparse representation using histogram information for image classification. In *Proc. IEEE Conf. on Computer Vision*, 2011.
- [25] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y.-T. Zheng. Nus-wide: A real-world web image database from national university of singapore. In *Proc. of ACM Conf. on Image and Video Retrieval (CIVR’09)*, Santorini, Greece., 2009.
- [26] O. Chum, M. Perdoch, and J. Matas. Geometric min-hashing: Finding a (thick) needle in a haystack. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 17–24, June 2009.

- [27] K. Crammer and Y. Singer. *On the algorithmic implementation of multiclass kernel-based vector machines*, volume 2. JMLR.org, Mar. 2002.
- [28] F. C. Crow. Summed-area tables for texture mapping. *SIGGRAPH Comput. Graph.*, 18(3):207–212, Jan. 1984.
- [29] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 886–893, 2005.
- [30] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *Proc. of the Twentieth Annual Symposium on Computational Geometry, SCG '04*, pages 253–262, New York, NY, USA, 2004. ACM.
- [31] S. Debnath, P. Mitra, N. Pal, and C. L. Giles. Automatic identification of informative sections of web pages. *IEEE Trans. Knowl. Data Eng.*, 17(9):1233–1246, 2005.
- [32] J. Deng, A. C. Berg, K. Li, and L. Fei-Fei. What does classifying more than 10,000 image categories tell us? In *Proc. European Conf. on Computer Vision*, pages 71–84, Berlin, Heidelberg, 2010.
- [33] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2009.
- [34] J. Deng, S. Satheesh, A. C. Berg, and F. F. F. Li. Fast and balanced: Efficient label tree learning for large scale object recognition. In *Proc. Advances in Neural Information Processing Systems*, pages 567–575. 2011.
- [35] T. Deselaers and V. Ferrari. Visual and semantic similarity in imagenet. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1777–1784, June 2011.
- [36] J. J. DiCarlo, D. Zoccolan, and N. C. Rust. How does the brain solve visual object recognition? *Neuron*, 73:415–34, 2012 Feb 9 2012.
- [37] C. Dorai and S. Venkatesh. Bridging the semantic gap with computational media aesthetics. *MultiMedia, IEEE*, 10(2):15–17, 2003.
- [38] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification (2nd Edition)*. Wiley-Interscience, 2 edition, 2001.
- [39] K. Engan, S. Aase, and J. Husoy. Frame based signal compression using method of optimal directions (mod). In *Proc. of the IEEE Intl. Symp. on Circuits and Systems*, volume 4, pages 1–4, July 1999.

- [40] P. Enser and C. Sandom. Towards a comprehensive survey of the semantic gap in visual image retrieval. In *Proceedings of the 2nd international conference on Image and video retrieval*, CIVR'03, pages 291–299, Berlin, Heidelberg, 2003. Springer-Verlag.
- [41] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010.
- [42] T. Evgeniou, C. A. Micchelli, and M. Pontil. Learning multiple tasks with kernel methods. *Journal of Machine Learning Research*, 6:615–637, 2005.
- [43] J. Fan, Y. Gao, and H. Luo. Integrating concept ontology and multitask learning to achieve more effective classifier training for multilevel image annotation. *IEEE Transactions on Image Processing*, 17(3):407–426, 2008.
- [44] J. Fan, Y. Gao, H. Luo, and R. Jain. Mining multilevel image semantics via hierarchical classification. *IEEE Transactions on Multimedia*, 10(2):167–187, 2008.
- [45] J. Fan, Y. Gao, H. Luo, and G. Xu. Statistical modeling and conceptualization of natural images. *Pattern Recogn.*, 38(6):865–885, June 2005.
- [46] J. Fan, X. He, **N. Zhou**, J. Peng, and R. Jain. Quantitative characterization of semantic gaps for learning complexity estimation and inference model selection. *IEEE Transactions on Multimedia*, 14(5):1414–1428, Oct. 2012.
- [47] J. Fan, D. A. Keim, Y. Gao, H. Luo, and Z. Li. Justclick: personalized image recommendation via exploratory search from large-scale flickr images. *IEEE Trans. Cir. and Sys. for Video Technol.*, 19(2):273–288, Feb. 2009.
- [48] J. Fan, Y. Shen, C. Yang, and **N. Zhou**. Structured max-margin learning for inter-related classifier training and multilabel image annotation. *IEEE Transactions on Image Processing*, 20(3):837–854, 2011.
- [49] J. Fan, Y. Shen, **N. Zhou**, and Y. Gao. Harvesting large-scale weakly-tagged image databases from the web. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 802–809, 2010.
- [50] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1778–1785, 2009.
- [51] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories. *Comput. Vis. Image Underst.*, 106(1):59–70, Apr. 2007.
- [52] C. Fellbaum. *WordNet An Electronic Lexical Database*. The MIT Press, Cambridge, MA ; London, May 1998.

- [53] J. Feng, B. Ni, Q. Tian, and S. Yan. Geometric  $l_p$ -norm feature pooling for image classification. In *Proc IEEE Conf. on Computer Vision and Pattern Recognition*, pages 2609–2704, June 2011.
- [54] S. Feng, V. Lavrenko, and R. Manmatha. Multiple bernoulli relevance models for image and video annotation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume 2, pages 1002–1009, 2004.
- [55] Y. Feng and M. Lapata. Automatic image annotation using auxiliary text information. In *ACL*, pages 272–280, 2008.
- [56] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. Learning object categories from google’s image search. In *Proc. IEEE Conf. on Computer Vision*, pages 1816–1823, 2005.
- [57] B. J. Frey and D. Dueck. Clustering by passing messages between data points. *Science*, 315:972–976, 2007.
- [58] W. Frstner and E. Glch. A fast operator for detection and precise location of distict point, corners and centres of circular features. pages 281–305, 87.
- [59] B. Gao, T.-Y. Liu, T. Qin, X. Zheng, Q. Cheng, and W.-Y. Ma. Web image clustering by consistent utilization of visual features and surrounding texts. In *ACM Multimedia*, pages 112–121, 2005.
- [60] P. Gehler and S. Nowozin. On feature combination for multiclass object classification. In *Proc. IEEE Conf. on Computer Vision*, pages 221–228, 29 2009-oct. 2 2009.
- [61] J. C. Gemert, J.-M. Geusebroek, C. J. Veenman, and A. W. Smeulders. Kernel codebooks for scene categorization. In *Proc. European Conf. on Computer Vision*, pages 696–709, Berlin, Heidelberg, 2008. Springer-Verlag.
- [62] K. Grauman and T. Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *Proc. IEEE Conf. on Computer Vision*, pages 1458–1465, 2005.
- [63] K. Grauman and B. Leibe. *Visual Object Recognition*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2011.
- [64] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology, 2007.
- [65] G. Griffin and P. Perona. Learning and using taxonomies for fast visual categorization. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1–8, June 2008.

- [66] J. S. Hare, P. A. S. Sinclair, P. H. Lewis, K. Martinez, P. G. B. Enser, and C. J. S. Bridging the semantic gap in multimedia information retrieval: Top-down and bottom-up approaches. In *In 3rd European Semantic Web Conference (ESWC-06), LNCS 4011*. Springer Verlag, 2006.
- [67] A. Hauptmann, R. Yan, and W.-H. Lin. How many high-level concepts will fill the semantic gap in news video retrieval? In *Proceedings of the 6th ACM international conference on Image and video retrieval, CIVR '07*, pages 627–634, New York, NY, USA, 2007. ACM.
- [68] A. Hauptmann, R. Yan, W.-H. Lin, M. Christel, and H. Wactlar. Can high-level concepts fill the semantic gap in video retrieval? a case study with broadcast news. *Multimedia, IEEE Transactions on*, 9(5):958–966, 2007.
- [69] W. H. Hsu, L. S. Kennedy, and S.-F. Chang. Video search reranking via information bottleneck principle. In *Proceedings of the 14<sup>th</sup> annual ACM international Conf. on Multimedia, MULTIMEDIA '06*, pages 35–44, New York, NY, USA, 2006. ACM.
- [70] W. H. Hsu, L. S. Kennedy, and S.-F. Chang. Video search reranking through random walk over document-level context graph. In *ACM Multimedia*, pages 971–980, 2007.
- [71] K. Huang and S. Aviyente. Sparse representation for signal classification. In *Proc. Advances in Neural Information Processing Systems*, pages 609–616. MIT Press, Cambridge, MA, 2007.
- [72] S. J. J. Hwang, K. Grauman, and F. Sha. Learning a tree of metrics with disjoint visual features. In *Proc. Advances in Neural Information Processing Systems*, pages 621–629. 2011.
- [73] Z. Jiang, Z. Lin, and L. Davis. Learning a discriminative dictionary for sparse coding via label consistent k-SVD. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1697–1704, June 2011.
- [74] Y. Jin, L. Khan, L. Wang, and M. Awad. Image annotations by combining multiple evidence & wordnet. In *Proc. of the 13<sup>th</sup> annual ACM international Conf. on Multimedia, Hilton, Singapore Nov. 06–11*, 2005.
- [75] Y. Jing and S. Baluja. Visualrank: Applying pagerank to large-scale image search. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(11):1877–1890, Nov. 2008.
- [76] U. H.-G. Kressel. *Advances in kernel methods*. pages 255–268, Cambridge, MA, USA, 1999. MIT Press.
- [77] C. H. 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*, pages 951–958, 2009.

- [78] V. Lavrenko, R. Manmatha, and J. Jeon. A model for learning the semantics of pictures. In *Proc. Advances in Neural Information Processing Systems*, 2003.
- [79] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume 2, pages 2169–2178, 2006.
- [80] C. Leacock and M. Chodorow. *Combining local context and WordNet similarity for word sense identification*, pages 305–332. In C. Fellbaum (Ed.), MIT Press, 1998.
- [81] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel. Handwritten digit recognition with a back-propagation network. In *Proc. Advances in Neural Information Processing Systems*, pages 396–404, 1989.
- [82] Y. LeCun, P. Haffner, L. Bottou, and Y. Bengio. Object recognition with gradient-based learning. In *Shape, Contour and Grouping in Computer Vision*, pages 319–, 1999.
- [83] H. Lee, A. Battle, R. Raina, and A. Y. Ng. Efficient sparse coding algorithms. In *Proc. Advances in Neural Information Processing Systems*, pages 801–808, 2006.
- [84] C. W. Leong and R. Mihalcea. Explorations in automatic image annotation using textual features. In *Linguistic Annotation Workshop*, pages 56–59, 2009.
- [85] F.-F. Li, P. Perona, and C. I. of Technology. A Bayesian hierarchical model for learning natural scene categories. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 524–531, 2005.
- [86] J. Li and J. Z. Wang. Automatic linguistic indexing of pictures by a statistical modeling approach. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(9):1075–1088, 2003.
- [87] L.-J. Li, C. Wang, Y. Lim, D. Blei, and L. Fei-fei. Building and using a semantivisual image hierarchy. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conf. on*, pages 3336–3343, June 2010.
- [88] L.-J. Li, G. Wang, and F.-F. L. 0002. Optimol: automatic online picture collection via incremental model learning. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2007.
- [89] X. Li, C. Snoek, and M. Worring. Learning social tag relevance by neighbor voting. *Multimedia, IEEE Transactions on*, 11(7):1310–1322, Nov. 2009.
- [90] Y. Lin, F. Lv, S. Zhu, M. Yang, T. Cour, K. Yu, L. Cao, and T. Huang. Large-scale image classification: Fast feature extraction and SVM training. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1689–1696, June 2011.

- [91] T. Lindeberg. Junction detection with automatic selection of detection scales and localization scales. In *ICIP (1)*, pages 924–928, 1994.
- [92] T. Lindeberg. Feature detection with automatic scale selection. *Int'l Journal of Computer Vision*, 30(2):79–116, 1998.
- [93] B. Liu, F. Sadeghi, M. F. Tappen, C. Liu, and O. Shamir. Probabilistic label trees for efficient large scale image classification. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conf. on*, June 2013.
- [94] D. Liu, X.-S. Hua, L. Yang, M. Wang, and H.-J. Zhang. Tag ranking. In *WWW*, pages 351–360, 2009.
- [95] J. Liu, W. Lai, X.-S. Hua, Y. Huang, and S. Li. Video search re-ranking via multi-graph propagation. In *Proceedings of the 15<sup>th</sup> international Conf. on Multimedia*, MULTIMEDIA '07, pages 208–217, New York, NY, USA, 2007. ACM.
- [96] Y. Liu, T. Mei, and X.-S. Hua. Crowdreranking: exploring multiple search engines for visual search reranking. In *Proceedings of the 32<sup>nd</sup> international ACM SIGIR Conf. on Research and development in information retrieval*, SIGIR '09, pages 500–507, New York, NY, USA, 2009. ACM.
- [97] D. G. Lowe. Object recognition from local scale-invariant features. In *Proc. IEEE Conf. on Computer Vision*, pages 1150–1157, 1999.
- [98] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int'l Journal of Computer Vision*, 60(2):91–110, 2004.
- [99] Y. Lu, L. Zhang, J. Liu, and Q. Tian. Constructing concept lexica with small semantic gaps. *IEEE Transactions on Multimedia*, 12(4):288–299, 2010.
- [100] Y. Lu, L. Zhang, Q. Tian, and W.-Y. Ma. What are the high-level concepts with small semantic gaps? In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, 2008.
- [101] H. Ma, J. Zhu, M.-T. Lyu, and I. King. Bridging the semantic gap between image contents and tags. *Multimedia, IEEE Transactions on*, 12(5):462–473, 2010.
- [102] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Discriminative learned dictionaries for local image analysis. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2008.
- [103] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Supervised dictionary learning. In *Proc. Advances in Neural Information Processing Systems*, pages 1033–1040, 2008.

- [104] J. Mairal, M. Leordeanu, F. Bach, M. Hebert, and J. Ponce. Discriminative sparse image models for class-specific edge detection and image interpretation. In *Proc. European Conf. on Computer Vision*, pages 43–56, 2008.
- [105] S. Maji. *Algorithms and Representations for Visual Recognition*. PhD thesis, EECS Department, University of California, Berkeley, May 2012.
- [106] M. Marszalek and C. Schmid. Semantic hierarchies for visual object recognition. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conf. on*, pages 1–7, June 2007.
- [107] M. Marszalek and C. Schmid. Constructing category hierarchies for visual recognition. In *Proc. European Conf. on Computer Vision*, volume 5305, pages 479–491, Marseille, France, 2008. Springer-Verlag.
- [108] M. Naphade, J. Smith, J. Tesic, S.-F. Chang, W. Hsu, L. Kennedy, A. Hauptmann, and J. Curtis. Large-scale concept ontology for multimedia. *MultiMedia, IEEE*, 13(3):86–91, 2006.
- [109] A. P. Natsev, M. R. Naphade, and J. R. Smith. Semantic representation: Search and mining of multimedia content. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pages 641–646, New York, NY, USA, 2004. ACM.
- [110] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Proc. Advances in Neural Information Processing Systems*, pages 849–856. MIT Press, 2001.
- [111] M.-E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *Proc. of Indian Conf. on Computer Vision, Graphics and Image Processing*, Dec. 2008.
- [112] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *Int'l Journal of Computer Vision*, 42(3):145–175, 2001.
- [113] B. A. Olshausen and D. J. Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision Research*, 37(23):3311–3325, 1997.
- [114] O. Parkhi, A. Vedaldi, A. Zisserman, and C. Jawahar. Cats and dogs. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3498–3505, 2012.
- [115] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2(6):559–572, 1901.
- [116] F. Perronnin, Z. Akata, Z. Harchaoui, and C. Schmid. Towards good practice in large-scale learning for image classification. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 3482–3489, June 2012.



- [117] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *Proc. European Conf. on Computer vision*, pages 143–156, Berlin, Heidelberg, 2010.
- [118] P. T. Pham, M.-F. Moens, and T. Tuytelaars. Cross-media alignment of names and faces. *IEEE Transactions on Multimedia*, 12(1):13–27, 2010.
- [119] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1–8, June 2008.
- [120] N. Pinto, D. D. Cox, and J. J. Dicarlo. Why is real-world visual object recognition hard? *PLoS Computational Biology*, 2008.
- [121] J. C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *ADVANCES IN LARGE MARGIN CLASSIFIERS*, pages 61–74. MIT Press, 1999.
- [122] I. Ramírez, P. Sprechmann, and G. Sapiro. Classification and clustering via dictionary learning with structured incoherence and shared features. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 3501–3508, 2010.
- [123] M. Ranzato, Y.-L. Boureau, and Y. LeCun. Sparse feature learning for deep belief networks. In *Proc. Advances in Neural Information Processing Systems*, 2007.
- [124] N. Rasiwasia, P. Moreno, and N. Vasconcelos. Bridging the gap: Query by semantic example. *Multimedia, IEEE Transactions on*, 9(5):923–938, 2007.
- [125] E. Rosch, C. B. Mervis, W. D. Gray, D. M., and P. Boyes-braem. Basic objects in natural categories. *Cognitive Psychology*, 1976.
- [126] H. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 203–208, 1996.
- [127] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. Labelme: A database and web-based tool for image annotation. *Int. J. Comput. Vision*, 77(1-3):157–173, May 2008.
- [128] D. G. Saari. Explaining all three-alternative voting outcomes. *Journal of Economic Theory*, 87(2):313–355, 1999.
- [129] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA, 1986.

- [130] J. Sanchez and F. Perronnin. High-dimensional signature compression for large-scale image classification. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1665–1672, June 2011.
- [131] S. Santini, A. Gupta, and R. Jain. Emergent semantics through interaction in image databases. *IEEE Trans. on Knowl. and Data Eng.*, 13(3):337–351, May 2001.
- [132] S. Satoh, Y. Nakamura, and T. Kanade. Name-it: Naming and detecting faces in news videos. *IEEE MultiMedia*, 6(1):22–35, 1999.
- [133] C. Schmid, R. Mohr, and C. Bauckhage. Evaluation of interest point detectors. *Int'l Journal of Computer Vision*, 37(2):151–172, 2000.
- [134] A. T. Schreiber, B. Dubbeldam, J. Wielemaker, and B. J. Wielinga. Ontology-based photo annotation. *IEEE Intelligent Systems*, 16(3):66–74, 2001.
- [135] F. Schroff, A. Criminisi, and A. Zisserman. Harvesting image databases from the web. In *Proc. IEEE Conf. on Computer Vision*, pages 1–8, 2007.
- [136] F. Schroff, A. Criminisi, and A. Zisserman. Harvesting image databases from the web. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(4):754–766, Apr. 2011.
- [137] J. Sivic, B. Russell, A. Zisserman, W. Freeman, and A. Efros. Unsupervised discovery of visual object class hierarchies. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1–8, June 2008.
- [138] A. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380, 2000.
- [139] C. G. M. Snoek, M. Worring, J.-M. Geusebroek, D. Koelma, F. Seinstra, and A. Smeulders. The semantic pathfinder: Using an authoring metaphor for generic multimedia indexing. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(10):1678–1689, 2006.
- [140] A. Sorokin and D. Forsyth. Utility data annotation with amazon mechanical turk. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW '08. IEEE Computer Society Conference on*, pages 1–8, 2008.
- [141] M. J. Swain and D. H. Ballard. Color indexing. *Int. J. Comput. Vision*, 7(1):11–32, Nov. 1991.
- [142] J. Tang, S. Yan, R. Hong, G.-J. Qi, and T.-S. Chua. Inferring semantic concepts from community-contributed images and noisy tags. In *Proceedings of the 17<sup>th</sup> ACM international Conf. on Multimedia*, MM '09, pages 223–232, New York, NY, USA, 2009. ACM.

- [143] S. Thorpe, D. Fize, and C. Marlot. Speed of processing in the human visual system. *Nature*, 381:520, 1996.
- [144] S. J. Thorpe. The speed of categorization in the human visual system. *Neuron*, 62(2):168 – 170, 2009.
- [145] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society (Series B)*, 58:267–288, 1996.
- [146] A. Torralba, R. Fergus, and W. Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(11):1958–1970, 2008.
- [147] A. Torralba, K. Murphy, and W. Freeman. Sharing visual features for multiclass and multiview object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(5):854–869, 2007.
- [148] T. Tuytelaars and K. Mikolajczyk. Local invariant feature detectors: a survey. *Found. Trends. Comput. Graph. Vis.*, 3(3):177–280, July 2008.
- [149] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek. Evaluating color descriptors for object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1582–1596, 2010.
- [150] V. Vapnik. *Statistical learning theory*. Wiley, 1998.
- [151] P. Viola and M. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, 2004.
- [152] L. Von Ahn and L. Dabbish. Labeling images with a computer game. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '04, pages 319–326, New York, NY, USA, 2004. ACM.
- [153] C. Wang, F. Jing, L. Zhang, and H.-J. Zhang. Image annotation refinement using random walk with restarts. In *Proceedings of the 14<sup>th</sup> annual ACM international Conf. on Multimedia*, MULTIMEDIA '06, pages 647–650, New York, NY, USA, 2006. ACM.
- [154] C. Wang, L. Zhang, and H.-J. Zhang. Learning to reduce the semantic gap in web image retrieval and annotation. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '08, pages 355–362, New York, NY, USA, 2008. ACM.
- [155] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 3360–3367, June 2010.
- [156] X.-J. Wang, W.-Y. Ma, G.-R. Xue, and X. Li. Multi-model similarity propagation and its application for web image retrieval. In *ACM Multimedia*, pages 944–951, 2004.

- [157] Y. Wang and S. Gong. Refining image annotation using contextual relations between words. In *Proc. of the 6<sup>th</sup> Intl. Conf. on Image and Video Retrieval (CIVR'07)*, pages 425–432, 2007.
- [158] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010.
- [159] J. Weston and C. Watkins. Multi-class support vector machines. Technical report, 1998.
- [160] J. M. Winn, A. Criminisi, and T. P. Minka. Object categorization by learned universal visual dictionary. In *Proc. IEEE Conf. on Computer Vision*, pages 1800–1807, 2005.
- [161] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(2):210–227, 2009.
- [162] L. Wu, X.-S. Hua, N. Yu, W.-Y. Ma, and S. Li. Flickr distance. In *Proceedings of the 16th ACM international conference on Multimedia*, MM '08, pages 31–40, New York, NY, USA, 2008. ACM.
- [163] K. Yanai. Generic image classification using visual knowledge on the web. In *Proceedings of the eleventh ACM international conference on Multimedia*, pages 167–176, New York, NY, USA, 2003. ACM.
- [164] A. Y. Yang, A. Ganesh, Z. Zhou, S. Shankar Sastry, and Y. Ma. A review of fast  $l_1$ -minimization algorithms for robust face recognition. *ArXiv e-prints*, July 2010.
- [165] J. Yang, K. Yu, Y. Gong, and T. S. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1794–1801, 2009.
- [166] J. Yang, K. Yu, and T. S. Huang. Supervised translation-invariant sparse coding. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 3517–3524, 2010.
- [167] L. Yang, R. Jin, R. Sukthankar, and F. Jurie. Unifying discriminative visual codebook generation with classifier training for object category recognition. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2008.
- [168] M. Yang, L. Zhang, X. Feng, and D. Zhang. Fisher discrimination dictionary learning for sparse representation. In *Proc. IEEE Conf. on Computer Vision*, 2011.

- [169] H.-F. Yu, C.-J. Hsieh, K.-W. Chang, and C.-J. Lin. Large linear classification when data cannot fit in memory. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '10, pages 833–842, New York, NY, USA, 2010. ACM.
- [170] S. Yu, D. Cai, J.-R. Wen, and W.-Y. Ma. Improving pseudo-relevance feedback in web information retrieval using web page segmentation. In *WWW*, pages 11–18, 2003.
- [171] X.-T. Yuan and S. Yan. Visual classification with multi-task joint sparse representation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 3493–3500, June 2010.
- [172] L. Zelnik-Manor and P. Perona. Self-tuning spectral clustering. In *Proc. Advances in Neural Information Processing Systems*, pages 1601–1608. Cambridge, MA, 2005.
- [173] D. Zhang and G. Lu. Review of shape representation and description techniques. *Pattern Recognition*, 37(1):1–19, 2004.
- [174] Q. Zhang and B. Li. Discriminative k-SVD for dictionary learning in face recognition. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 2691–2698, June 2010.
- [175] R. Zhang, Z. M. Zhang, M. Li, W.-Y. Ma, and H. Zhang. A probabilistic semantic model for image annotation and multi-modal image retrieval. In *Proc. IEEE Conf. on Computer Vision*, pages 846–851, 2005.
- [176] S. Zhang, Y. Zhan, and D. N. Metaxas. Deformable segmentation via sparse representation and dictionary learning. *Medical Image Analysis*, 16(7):1385–1396, 2012.
- [177] W. Zhang, A. Surve, X. Fern, and T. G. Dietterich. Learning non-redundant codebooks for classifying complex objects. In *Intl Conf. Machine Learning*, page 156, 2009.
- [178] R. Zhao and W. I. Grosky. Narrowing the semantic gap - improved text-based web document retrieval using visual features. *Multimedia, IEEE Transactions on*, 4(2):189–200, 2002.
- [179] **N. Zhou**, W. K. Cheung, G. Qiu, and X. Xue. A hybrid probabilistic model for unified collaborative and content-based image tagging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(7):1281–1294, 2011.
- [180] **N. Zhou** and J. Fan. Jointly learning visually correlated dictionaries for large-scale visual recognition applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP(99):1–1, 2013.

- [181] **N. Zhou**, Y. Shen, and J. Fan. Automatic image annotation by using relevant keywords extracted from auxiliary text documents. In *Proceedings of the international workshop on Very-large-scale multimedia corpus, mining and retrieval*, VLS-MCMR '10, pages 7–12, New York, NY, USA, 2010. ACM.
- [182] **N. Zhou**, Y. Shen, J. Peng, and J. Fan. Learning inter-related visual dictionary for object recognition. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 3490–3497, June 2012.
- [183] **N. Zhou**, Y. Shen, J. Peng, X. Feng, and J. Fan. Leveraging auxiliary text terms for automatic image annotation. In *Proceedings of the 20<sup>th</sup> international Conf. companion on World wide web*, WWW '11, pages 175–176, New York, NY, USA, 2011. ACM.
- [184] X. Zhou, K. Yu, T. Zhang, and T. S. Huang. Image classification using super-vector coding of local image descriptors. In *Proc. European Conf. on Computer Vision*, pages 141–154, Berlin, Heidelberg, 2010.