

SEMANTIC IMAGE UNDERSTANDING: FROM THE WEB, IN
LARGE SCALE, WITH REAL-WORLD CHALLENGING DATA

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Jia Li

November 2011

© 2011 by Jia Li. All Rights Reserved.

Re-distributed by Stanford University under license with the author.



This work is licensed under a Creative Commons Attribution-Noncommercial 3.0 United States License.

<http://creativecommons.org/licenses/by-nc/3.0/us/>

This dissertation is online at: <http://purl.stanford.edu/qk372kq7966>

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Fei-Fei Li, Primary Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Daphne Koller

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Andrew Ng

Approved for the Stanford University Committee on Graduate Studies.

Patricia J. Gumpert, Vice Provost Graduate Education

This signature page was generated electronically upon submission of this dissertation in electronic format. An original signed hard copy of the signature page is on file in University Archives.

Abstract

Human can effortlessly perceive rich amount of semantic information from our visual world including objects within it, the scene environment, and event/activity taking place etc.. Such information has been critical for us to enjoy our life. In computer vision, an important, open problem is to endow computers/intelligent agents the ability to extract semantically meaningful information as human does.

The primary goal of my research is to design and demonstrate visual recognition algorithms to bridge the gap between visual intelligence and human perception. Towards this goal, we have developed rigid statistical models to represent the large scale real-world challenging data especially those from Internet. Visual features are the starting-point of computer vision algorithms. We propose a novel high-level image representation to encode the abundant semantic and structural information within an image.

We first focus on introducing principle generative models for modeling our rich visual world, from recognizing objects in an image, to a detailed understanding of scene/activity images, to inferring the relationship among large scale user images and related textual data. We propose a non-parametric topic model, hierarchical Dirichlet Process (HDP), in a robust noise rejection system for object recognition, learning the object model and re-ranking noisy web images containing the objects in an iterative online fashion. It learns the object model in a fully automatic way, freeing the researchers from heavy human labor in labeling training examples for recognizing objects. This framework has been tested on a large scale corpus of over 400 thousand images and also won the Software Robot first Prize in the 2007 Semantic Visual Recognition Competition.

Understanding our visual world is beyond simply recognizing objects. We then present a generative model for understanding complex scenes that involve objects, humans and scene backgrounds to interact together. For detailed understanding of an image, we propose the very first model for event recognition in a static image by combining the objects appear in the event and the scene environment, where the event takes place. We are not only interested in the category prediction of an unknown image, but also in how pixels form coherent objects and the semantic concepts related to them. We propose the first principled graphical model that tackles three very challenging vision tasks in one framework: image classification, object annotation, and object segmentation. Our statistical model encodes the relationships of pixel visual properties, object identities, textual concepts and the image class. It is a much larger scale departure from the previous work, using real-world challenging user photos such as noisy, Flickr images and user tags to learn the model in an automatic framework. Interpreting single images is an important corner stone for inferring relationships among large scale images to effectively organize them. We propose a joint visual-textual model based upon the nested Chinese Restaurant Process (nCRP) model. Our model combines textual semantics (user tags) with image visual contents, which learns a semantically and visually meaningful image hierarchy on thousands of Flickr user images with noisy user tags. The hierarchy performs significantly better on image classification and annotation performance as a knowledge base comparing to the state-of-the-art algorithms.

Visual recognition algorithms start from representation of the images, the so-called image feature. While the goal of visual recognition is to recognize object and scene contents that are semantically meaningful, all previous work have relied on low-level feature representations such as filter banks, textures, and colors, creating the well known semantic gap. We propose a fundamentally new image feature, Object Bank, which uses hundreds and thousands of object sensing filters (i.e. pre-trained object detectors) to represent an image. Instead of representing an image based on its color, texture or likewise, Object Bank depicts an image by objects appearing in the image and their locations. Encoding rich descriptive semantic and structural information of an image, Object Bank is extremely robust and powerful for complex

scene understanding, including classification, retrieval and annotation.

To my family

Acknowledgements

This thesis would not have been possible without the support, guidance and love from so many people around me. I would like to especially thank my advisor, Professor Fei-Fei Li for her encouragement, insight and patience. I was fortunate to be the first couple of students in Professor Fei-Fei Li’s research group, and have appreciated her seemingly unlimited supply of clever and kindness. I have learned from her not only how to conduct high-quality research, but also invaluable advices in life.

I am also grateful to my other thesis committee members, Prof. Daphne Koller, Prof. Andrew Ng, and Prof. Christopher Manning for giving me advices in research, career and life. I would like to thank Prof. Kalanit Grill-Spector for willing to be my committee chair without even knowing me.

During my PhD study, it has been my great honor to collaborate with Prof. David Blei and Prof. Eric Xing, from whom, I learned incredible machine learning knowledge. My research has also been greatly benefited from the collaboration with (in chronological order) Gang Wang, Juan Carlos Niebles, Richard Socher, Jia Deng, Chong Wang, Yongwhan Lim, Barry Chai, Hao Su and Jun Zhu. Thank you for making my research journey exciting and enjoyable.

I am also thankful to my fellow students and members in the Vision Lab for the insightful research discussions and great moments: Liangliang Cao, Min Sun, Bangpeng Yao, Louis Chen, Kevin Tang, Chris Baldassano, Marius Catalin Iordan and Dave Jackson. I am particularly grateful to Olga Russakovsky, who is always ready to help me. I will make sure to be there to support her when it is her turn.

I would like to thank my parents for their unconditional love and support. My appreciation to them is beyond what I can express in words. Last but not least, I

especially want to thank my husband, Yujia Li, for his understanding, encouragement and love.

Contents

Abstract	iv
Acknowledgements	viii
I Introduction	1
II Probabilistic Models for Semantic Image Understanding	6
1 Learning object model from noisy Internet images	8
1.1 Related works	10
1.2 Overview of the OPTIMOL Framework	15
1.3 Theoretical Framework of OPTIMOL	17
1.3.1 Our Model	17
1.3.2 Learning	20
1.3.3 New Image Classification and Annotation	24
1.3.4 Discussion of the Model	27
1.4 Walkthrough for the accordion category	28
1.5 Experiments & Results	33
1.5.1 Datasets Definitions	34
1.5.2 Exp.1: Analysis Experiment	35
1.5.3 Exp.2: Image Collection	42

1.5.4	Exp.3: Classification	44
1.6	Discussion	45
2	Towards Total Scene Understanding	49
2.1	Introduction and Motivation	50
2.2	Related Work	52
2.3	Event Image Understanding by Scene and Object Recognition	53
2.3.1	The Integrative Model	54
2.3.2	Labeling an Unknown Image	59
2.3.3	Learning the Model	60
2.3.4	System Implementation	61
2.3.5	Dataset and Experimental Setup	62
2.3.6	Results	63
2.3.7	Discussion	65
2.4	A Probabilistic Model Towards Total Scene Understanding: Simultaneous Classification, Annotation and Segmentation	67
2.4.1	The Hierarchical Generative Model	67
2.4.2	Properties of the Model	70
2.4.3	Learning via Collapsed Gibbs Sampling	71
2.4.4	Automatic Initialization Scheme	73
2.4.5	Learning Summary	74
2.4.6	Inference: Classification, Annotation and Segmentation	75
2.4.7	Experiments and Results	76
2.5	Discussion	80
3	Probabilistic Model for Automatic Image Organization	83
3.1	Related Work	85
3.2	Building the Semantivisual Image Hierarchy	86
3.2.1	A Hierarchical Model for both Image and Text	86
3.2.2	Learning the Semantivisual Image Hierarchy	89
3.2.3	A Semantivisual Image Hierarchy	91
3.3	Using the Semantivisual Image Hierarchy	96

3.3.1	Hierarchical Annotation of Images	96
3.3.2	Image Labeling	97
3.3.3	Image Classification	98
3.4	Discussion	100

III High Level Image Representation for Semantic Image Understanding 101

4	Introduction 102
4.1	Background and Related Work 103
5	The Object Bank Representation of Images 106
5.1	Construction of Object Bank 106
5.2	Implementation Details of Object Bank 107
5.3	High Level Visual Recognition by Using Different Visual Representations 110
5.3.1	Object Bank on Scene Classification 111
5.3.2	Object Bank on Object Recognition 113
5.4	Analysis: Role of Each Ingredient 114
5.4.1	Comparison of Different Types of Detectors 115
5.4.2	Role of View Points 116
5.4.3	Role of Scales 117
5.4.4	Role of Spatial Location 120
5.4.5	Comparison of Different Pooling Methods 121
5.4.6	Role of Objects 122
5.4.7	Relationship of Objects and Scenes Discovered by OB 129
5.4.8	Relationship of Different Objects Discovered by OB 131
5.5	Analysis: Guideline of Using Object Bank 131
5.5.1	Robustness to different classification models 131
5.5.2	Dimension Reduction by Using PCA 133
5.5.3	Dimension Reduction by Combining Different Views 134
5.6	Discussion 135

6	Semantic Feature Sparsification of Object Bank	136
6.1	Scene Classification and Feature/Object Compression via Structured Regularized Learning	137
6.2	Experiments and Results	139
6.2.1	Semantic Feature Sparsification Over OB	139
6.2.2	Interpretability of the compressed representation	142
6.3	Discussion	144
7	Multi-Level Structured Image Coding on Object Bank	146
7.1	Introduction and Background	146
7.2	Object Bank Revisit: a Structured High Dimensional Image Representation	148
7.3	Multi-Level Structured Image Coding	149
7.3.1	Basic Sparse Coding	149
7.3.2	The MUSIC Approach	149
7.3.3	Optimization Algorithm: Coordinate Descent	154
7.4	Experiment	157
7.4.1	Analysis of MUSIC	157
7.4.2	Applications in High-level Image Recognition	161
7.5	Discussion	165
IV	Conclusion	167
	Bibliography	171

List of Tables

2.1	Comparison of precision and recall values for annotation with Alipr, corr-LDA and our model. Detailed results are given for seven objects, but means are computed for all 30 object categories (Experiment D).	79
2.2	Results of segmentation on seven object categories and mean values for all 30 categories (Experiment E).	80
5.1	Object classification performance by using different high level representations. . .	113
5.2	Classification performance by using different spatial location structure.	120
5.3	Classification performance of different methods.	133
7.1	Classification accuracy of different models.	161
7.2	Classification performance by using different classifiers and self-taught learning (we learn a MUSIC on the MIT indoor data and apply it to infer image codes for images in the UIUC sports data) on the image codes inferred by MUSIC.	163

List of Figures

- 1.1 **Illustration of the framework of the Online Picture collecTion via Incremental MOdel Learning (OPTIMOL) system.** This framework works in an incremental way: Once a model is learned, it can be used to classify images from the web resource. The group of images classified as being in this object category are regarded as related images. Otherwise, they are discarded. The model is then updated by a subset of the newly accepted images in the current iteration. In this incremental fashion, the category model gets more and more robust. As a consequence, the collected dataset becomes larger and larger. 16
- 1.2 **Graphical model of HDP.** Each node denotes a random variable. Bounding boxes represent repetitions. Arrows indicate conditional dependency. Dark node indicates it is observed. 19
- 1.3 **Influence of the threshold(Eq.1.11) on the number of images to be appended to the dataset and held in the cache set on 100 “accordion” images.** x-axis is the value of threshold represented in percentile. The validation ratio thresholds are 1, 5, 10, 30, 50, 70, 90, 100, which are equivalent to -1.94, 2.10, 14.24, 26.88, 44.26, 58.50, 100.22 and 112.46 in log likelihood ratio thresholds respectively for the current classifier. y-axis denotes the number of images. Blue region represents number of images classified as unrelated. Yellow region denotes the number of images that will be appended to the object dataset. Pink region represents number of images held in the “cache set”. The “true” bar represents the proportion of true images in the 100 testing images. These bars are generated using the initial model learned from 15 seeds images. The higher the threshold is, the fewer number of images will be appended to the permanent dataset and held in the “cache set”. 29

1.4	Downloading part in Step 1. A noisy “accordion” dataset is downloaded using “accordion” as query word in Google image, Yahoo! image and Picsearch. Downloaded images will be further divided into groups for the iterative process.	30
1.5	Preprocessing in Step 1. Top: Regions of interest found by Kadir&Brady detector. The circles indicate the interest regions. The red crosses are the centers of these regions. Bottom: Sample codewords. Patches with similar SIFT descriptors are clustered into the same codeword, which are presented using the same color.	30
1.6	Initial batch learning. In the first iteration, a model is learned from the seed images. The learned model performs fairly well in classification as shown in Fig. 1.9.	31
1.7	Classification. Classification is performed on a subset of raw images collected from the web using the “accordion” query. Images with low likelihood ratios measured by Eq.1.11 are discarded. For the rest of the images, those with low entropies are incorporated into the permanent dataset, while the high entropy ones stay in the “cache set”.	32
1.8	Incremental Learning. The model is updated using only the images held in the “cache set” from the previous iteration.	32
1.9	Batch vs. Incremental Learning (a case study of the “inline skate” category with 4835 images). Left: The number of images retrieved by the incremental learning algorithm, the batch learning algorithm and the base model. Detection rate is displayed on top of each bar. x-axis represents batch learning with 15 seed images, batch learning with 25 seed images, incremental learning with 15 seed images, incremental learning with 25 seed images, the base model learned from 15 seed images and 25 seed images respectively. Middle: Running time comparison of the batch learning method, the incremental learning method and the base model learned as a function of number of training iterations. The incrementally learned model is initialized by applying the batch learning algorithm on 15 or 25 training images, which takes the same amount of time as the corresponding batch method does. After initialization, incremental learning is more efficient compared to the batch method. Right: Recognition accuracy of the incrementally learned, batch learned models and the base model evaluated by using Receiver Operating Characteristic (ROC) Curves.	35

1.10	Average image of each category, in comparison to the average images of <i>Caltech101</i> . The grayer the average image is, the more diverse the dataset is.	37
1.11	Diversity of our collected dataset. Left: Illustration of the diverse of clusters in the “accordion” dataset. Root image is the average image of all the images in “accordion” dataset collected by OPTIMOL. Middle layers are average images of the top 3 “accordion” clusters generated from the learned model. Leaf nodes of the tree structure are 3 example images attached to each of the cluster average image. Right: Illustration of the diverse clusters in the “euphonium” dataset.	38
1.12	Left: Number of global clusters shared within each category as a function of the value of γ . Right: Average number of clusters in each image as a function of the value of α . The standard deviation of the average numbers are plotted as vertical bars centered at the data points.	38
1.13	Left: Data collection results of OPTIMOL with different likelihood ratio threshold values for the “accordion” dataset. X-axis denotes the likelihood ratio threshold values. Y-axis represents the number of collected images. The number on the top of each bar represents the detection rate for OPTIMOL with that entropy threshold value. Right: Data collection results of OPTIMOL with different likelihood ratio threshold values for the “euphonium” dataset.	39
1.14	Left: Illustration of images in the permanent dataset, the “cache set” and the “junk set”. X-axis represents the likelihood while y-axis represents the entropy. If the likelihood ratio of an image is higher than some threshold, it is selected as a related image. This image will be further measured by its entropy. If the image has low entropy, it will be appended to the permanent dataset. If it has high entropy, it will stay in the “cache set” to be further used to train the model. Right: Examples of high and low entropy images in “accordion” and “inline-skate” classes.	40
1.15	Sampled images from dataset collected by using different entropy threshold values. Left: Example images from dataset collected with entropy threshold set at top 100% (all images). Right: Example images from dataset collected with entropy threshold set at top 30%.	41
1.16	Detection performance. x-axis is the number of seed images. y-axis represents the detection rate.	41

1.17	Polysemy discovery using OPTIMOL. Two polysemous query words are used as examples: “mouse” and “bass”. Left: Example images of “mouse” and “bass” from image search engines. Notice that in the “mouse” group, the images of animal mouse and computer mouse are intermixed with each other, as well as with other noisy images. The same is true for the “bass” group. For this experiment, 50 seed images are used for each class. Right: For each query word, example images of the two main topic clusters discovered by OPTIMOL are demonstrated. We observe that for the “mouse” query, one cluster mainly contains images of the animal mouse, whereas the other cluster contains images of the computer mouse.	43
1.18	Left: Randomly selected images from the image collection result for “accordion” category. False positive images are highlighted by using red boxes. Right: Randomly selected images from the image collection result for “inline-skate” category.	44
1.19	Confusion table for Exp.3. We use the same training and testing datasets as in [51]. The average performance of OPTIMOL is 74.82%, whereas [51] reports 72.0%.	45
1.20	Image collection and annotation results by OPTIMOL. Each row in the figure contains two categories, where each category includes 4 sample annotation results and a bar plot. Let us use “Sunflower” as an example. The left sub-panel gives 4 sample annotation results (bounding box indicates the estimated locations and sizes of the “Sunflower”). The right sub-panel shows the comparison of the number of images in “Sunflower” category given different datasets. The blue bar indicates the number of “Sunflower” images in LabelMe dataset, the yellow bar the number of images in Caltech 101-Human. The OPTIMOL results are displayed using the red, green, and cyan bars, representing the numbers of images retrieved for the “Sunflower” category in Caltech 101-Web, Web-23 and Princeton-23 dataset respectively. The gray bar in each figure represents the number of images retrieved by the base model trained with only seed images. The number on top of each bar represents the detection rate for that dataset.	47

1.21	Image collection and annotation results by OPTIMOL. Notation is the same as Fig. 1.20. Since the pictures in the “face” category of Caltech 101-Human were taken by camera instead of downloading from the web, the raw Caltech images of the “face” category are not available. Hence, there is no result for “face” by 101 (OPTIMOL). All of our results have been put online at http://vision.stanford.edu/projects/OPTIMOL.htm	48
2.1	An example of what our total scene model can understand given an unknown image. At the scene level, the image is classified as a “polo” scene/event. A number of objects can be inferred and segmented by the visual information in the scene, hierarchically represented by object regions and feature patches. In addition, several tags can be inferred based on the scene class and the object correspondence.	50
2.2	Telling the <i>what, where and who</i> story. Given an <i>event</i> (rowing) image such as the one on the left, our system can automatically interpret what is the event, where does this happen and who (or what kind of objects) are in the image. The result is represented in the figure on the right. A red name tag over the image represents the event category. The scene category label is given in the white tag below the image. A set of name tags are attached to the estimated centers of the objects to indicate their categorical labels. As an example, from the image on the right, we can tell from the name tags that this is a rowing sport event held on a lake (scene). In this event, there are rowing boat, athletes, water and trees (objects).	54
2.3	Graphical model of our approach. E, S, and O represent the event, scene and object labels respectively. X is the observed appearance patch for scene. A and G are the observed appearance and geometry/layout properties for the object patch. The rest of the nodes are parameters of the model. For details, please refer to Sec. 2.3.1	56
2.4	Our dataset contains 8 sports event classes: rowing (250 images), badminton (200 images), polo (182 images), bocce (137 images), snowboarding (190 images), croquet (236 images), sailing (190 images), and rock climbing (194 images). Our examples here demonstrate the complexity and diversity of this highly challenging dataset.	60

2.5	<p>Left: Confusion table for the 8-class event recognition experiment. The average performance is 73.4%. Random chance would be 12.5%. Right: Performance comparison between the full model and the three control models. The x-axis denotes the name of the model used in each experiment, where “full model” indicates the proposed integrative model (see Fig. 2.3). The y-axis represents the average 8-class discrimination rate, which is the average score of the diagonal entries of the confusion table of each model.</p>	64
2.6	<p>(This figure is best viewed in color and with PDF magnification.) Image interpretation via event, scene, and object recognition. Each row shows results of an event class. Column 1 shows the event class label. Column 2 shows the object classes recognized by the system. Masks with different colors indicate different object classes. The name of each object class appears at the estimated centroid of the object. Column 3 is the scene class label assigned to this image by our system. Finally Column 4 shows the sorted object distribution given the event. Names on the x-axis represents the object class, the order of which varies across the categories. y-axis represents the distribution.</p>	66
2.7	<p>A graphical model representation of our generative model. Nodes represent random variables and edges indicate dependencies. The variable at the right lower corner of each box denotes the number of replications. The box indexed by N_r represents the visual information of the image, whereas the one indexed by N_t represents the textual information (i.e. tags). $N_c, N_o, N_x, N_{f_i}, i \in 1, 2, 3, 4$ denote the numbers of different scenes, objects, patches and regions for region feature type i respectively. Hyperparameters of the distributions are omitted for clarity.</p>	68
2.8	<p>Probabilities of different objects. Words such as “horse” or “net” have higher probability because users tend to only tag them when the objects are really present, largely due to their clear visual relevance. On the contrary, words such as “island” and “wind” are usually related to the location or some other visually irrelevant concept, and usually not observable in a normal photograph.</p>	71

- 2.9 Walk-through of the learning process. **(a):** The original image. **(b):** The original tags from Flickr. Visually irrelevant tags are colored in red. **(c):** Output of Step 1 of Algorithm 1: Tags after the WordNet pruning. **(d):** Output of Step 2 of Algorithm 1: The image is partly annotated using the initialization scheme. Different object concepts are colored differently. Note that there is a background class in our initialization scheme, which is denoted in black in this figure. Since the criterion for being selected as an initial image is very conservative, the image annotations are clean but many regions are not annotated (missing tags). **(e):** Output of Step 3 of Algorithm 1: After training the hierarchical model, the image is completely and more precisely segmented. **(f):** Final annotation proposed by our approach. Blue tags are predicted by the visual component ($S = \text{visual}$). Green tags are generated from the top down scene information learned by the model ($S = \text{non-visual}$). 74
- 2.10 Comparison of classification results. **Left: Overall performance.** Confusion table for the 8-way scene classification. Rows represent the models for each scene while the columns represent the ground truth classes. The overall classification performance is 54%. **Right: Comparison with different models (Experiment A).** Performance of four methods. Percentage on each bar represents the average scene classification performance. 3rd bar is the modified Corr-LDA model [16]. 77
- 2.11 **Left: Influence of unannotated data (Experiment B).** Classification performance as a function of number of unannotated images. The y axis represents the average classification performance. The x axis represents the number of unlabeled images. It shows the unannotated images also contribute to the learning process of our model. **Right: Effect of noise in tags (Experiment C).** Performance of different models as a function of noise percentage in the tags. The y axis is average classification performance. The x axis represents the percentage of noisy tags. While the performance of corr-LDA decreases with the increase of percentage of noise, our model performs robustly by selectively learning the related tags. . . . 78

2.12	Comparison of object segmentation results with or without the top down scene class influence. Each triplet of images show results of one scene class (Experiment F). The left image shows object segmentation result without the top down contextual information, i.e. by setting the probability distribution of object given scene class to a fixed uniform distribution. The center image shows object segmentation result by using the full model. We observe objects are more accurately recognized and delineated. The right image shows the probability of the 5 most likely objects per scene class. This probability encodes the top down contextual information.	81
3.1	Traditional ways of organizing and browsing digital images include using dates or filenames, which can be a problem for large sets of images. Images organized by semantically meaningful hierarchy could be more useful.	84
3.2	Schematic illustration of associating a training image in the semantivisual hierarchical model (left) and assigning a test image to a node on a given path of the hierarchy (right). The hierarchical model is summarized in variable T , where only one path is explicitly drawn from C_1 to C_n . Left of the model: Two training images and their Flickr tags are shown. Each image is further decomposed into regions. Each region is characterized by the features demonstrated in the bounding box on the left. A region is assigned to a node that best depicts its semantic meaning. Right of the model: A query image is assigned to a path based on the distribution of the concepts it contains. To further visualize the image on a particular node of the path, we choose the node that corresponds to the dominating region concepts in the image.	87
3.3	The graphical model (Top) and the notations of the variables(Bottom). 88	
3.4	Example images from each of the 40 Flickr classes.	91

3.5	Visualization of the learned image hierarchy. Each node on the hierarchy is represented by a colored plate, which contains four randomly sampled images associated with this node. The color of the plate indicates the level on the hierarchy. A node is also depicted by a set of tags, where only the first tag is explicitly spelled out. The top subtree shows the root node “photo” and some of its children. The rest of this figure shows six representative sub-trees of the hierarchy: “event”, “architecture”, “food”, “garden”, “holiday” and “football”.	94
3.6	Evaluation of the hierarchy. Top: “How meaningful is the path” experiment. Top Left: The AMT users are provided with a list of words. The users need to identify the words that are not related to the image to the left. Top Right: Quantitative results of our hierarchy and nCRP[15]. Our hierarchy performs the best by incorporating the visual information associated to the tags. Bottom: “How meaningful is the hierarchy” experiment. Bottom Left: The AMT users are provided with all permutations of candidate words from the path corresponding to the image that correctly represents the hierarchical structure. Bottom Right: Quantitative results of our hierarchy, nCRP [15] and Flickr. All three algorithms use exactly the same tag input to construct the hierarchy.	95
3.7	Results of the hierarchical annotation experiment. Three sample images and their hierarchical annotations by our algorithm and the original Flickr tags are shown. The table presents quantitative comparison on our hierarchy and nCRP[15]. The performance is measured by the modified Damerau-Levenshtein distance between the proposed hierarchical annotation by each algorithm and the human subjects’ result. .	96
3.8	Results of the image labeling experiment. We show example images and annotations by using our hierarchy, the Corr-LDA model [16] and the Alipr algorithm [89]. The numbers on the right are quantitative evaluations of these three methods by using an AMT evaluation task.	98

- 3.9 Comparison of classification results. **Top Left: Overall performance.** Confusion table for the 40-way Flickr images classification. Rows represent the models for each class while the columns represent the ground truth classes. **Top Right: Comparison with different models.** Percentage on each bar represents the average scene classification performance. Corr-LDA also has the same tag input as ours. **Bottom: classification example.** Example images that our algorithm correctly classified but all other algorithms misclassified. . . . 99
- 5.1 (Best viewed in colors and magnification.) Illustration of the object filter representations. Given an input image, we first run a large number of object detectors at multiple scales to obtain the object responses, the probability of objects appearing at each pixel. For each object at each scale, we apply a three-level spatial pyramid representation of the resulting object filter map, resulting in `No.Objects` \times `No.Scales` \times $(1^2 + 2^2 + 4^2)$ grids. An Object Bank representation of an image is a concatenation of statistics of object responses in each of these grids. We consider three ways of encoding the information. The first is the *max response representation (OB-Max)*, where we compute the maximum response value of each object, resulting in a feature vector of `No.Objects` length for each grid. The second is the *average response representation (OB-Avg)*, where we extract the average response value in each grid. The resulting feature vector has the same length as the maximum response. The third is the *histogram representation (OB-Hist)*. Here for each of the object detectors, we keep track of the percent of pixels on a discretized number of response values, resulting in a vector of `No.BinnedResponseValues` \times `No.Objects` length for each grid. 108

5.2	(Best viewed in colors and magnification.) Left: The frequency (or popularity) of objects in the world follows Zipf’s law trend: a small proportion of objects occurs much more frequently than the majority. While there are many ways of measuring this, e.g., by ranking object names in popular corpora such as the American National Corpora [73] and British National Corpus [38], we have taken a web-based approach by counting the number of downloadable images corresponding to object classes in WordNet on popular search engines such as Google, Ask.com and Bing. We show here the distribution of the top 2000 objects. Right: Rough grouping of the chosen object filters based loosely on the WordNet hierarchy [103]. The size of each unshaded node corresponds to the number of images returned by the search.	109
5.3	(Best viewed in colors and magnification.) Comparison of Object Bank representation with two low-level feature representations, GIST and SIFT-SPM of two types of images, mountain vs. city street. For each input image, we first show the selected filter responses in the GIST representation [109]. Then we show a histogram of the SPM representation of SIFT patches [85] at level 2 of the SPM representation where the codeword map is also shown as a histogram. Finally, we show a selected number of object filter responses.	110
5.4	(Best viewed in colors and magnification.) Comparison of classification performance of different features (GIST vs. BOW vs. SPM vs. Object Bank) and classifiers (SVM vs. LR) on (top to down) 15 scene, LabelMe, UIUC-Sports and MIT-Indoor datasets. In the LabelMe dataset, the “ideal” classification accuracy is 90%, where we use the human ground-truth object identities to predict the labels of the scene classes. Previous state-of-the-art performance are displayed by using the green bars.	112
5.5	Left: Detection performance comparison of different detection methods on ImageNet objects. Right: Classification performance of different detection methods on UIUC sports dataset and MIT Indoor dataset.	115
5.6	Diverse views of rowing boats in different images. Images are randomly selected from the UIUC sports dataset.	116

5.7	<p>Left: Classification performance of Object Bank generated from detectors trained on images with different view points on the UIUC sports dataset and the MIT Indoor dataset. Right: Classification performance of Object Bank generated from different views on images with different view points.</p>	117
5.8	<p>Left: Classification performance on the UIUC sports event dataset by using Object Bank representation corresponding to each single scale. Right: Classification performance on the MIT Indoor dataset by using Object Bank representation corresponding to each single scale. X axis is the index of the scale from fine to coarse. Y axis represents the average precision of a 8-way classification.</p>	118
5.9	<p>Left: Classification performance on UIUC sports event dataset by using Object Bank representation corresponding to each single scale. Right: Classification performance on MIT Indoor dataset by using Object Bank representation corresponding to each single scale. X axis is the index of the scale from fine to coarse. Y axis represents the average precision of a 8-way classification.</p>	118
5.10	<p>Binary classification experiment for each individual scale and the combination of them. Each grid is filled in with “ball” in different size. Each row represents a model trained on images with relatively similar scale (from small to large. Last row is the combination of all scales). Each column represents a test set with relatively similar scale. The more transparent the mask is, the better the classification accuracy is.</p>	119
5.11	<p>Left: Heat map of possible locations estimated from classification performance of Object Bank representation generated from different spatial locations. Right: Example images with the possible location map overlaid on the original image. . .</p>	121
5.12	<p>Left: Classification performance of different pooling methods. Right: Classification performance of PCA projected representations by using different pooling methods. Dimension is fixed to the minimum number (~ 100 dimensions) of principal components to preserve 99% of the unlabeled data variance of the three representations. Average and maximum response values within each spatial pyramid grid are extracted as the Object Bank feature in average pooling and max pooling respectively. We discretize values within each spatial pyramid to construct the histogram pooling representation.</p>	122

5.13	<p>Left: Classification performance on UIUC sports event dataset by using Object Bank representation corresponding to each single object. Right: Classification performance on MIT Indoor dataset by using Object Bank representation corresponding to each single object. X axis is the index of the object sorted by using the detection performance on object datasets from ImageNet. Y axis represents the average precision of a 8-way classification.</p>	123
5.14	<p>Classification performance on UIUC sports event dataset and MIT Indoor dataset by using Object Bank representation corresponding to accumulative object. X axis is the number of objects. Y axis represents the average precision of a 8-way classification.</p>	124
5.15	<p>Model comparison of “sail boat” and “human” models trained on UIUC training images (Left) and ImageNet images (Right).</p>	125
5.16	<p>Detection performance comparison of models trained on UIUC training images and ImageNet images.</p>	126
5.17	<p>Classification performance on UIUC sports event dataset by using Object Bank representation generated from sailboat and human models trained on UIUC training images and ImageNet images respectively. Y axis represents the average precision of a 8-way classification.</p>	126
5.18	<p>Left: Classification performance on UIUC sports event dataset by using UIUC-25 (customized Object Bank), ImageNet-177 (generic Object Bank), ImageNet-25 (25 objects randomly selected from ImageNet object candidates), randomly generated filters (Pseudo Object Bank) and state-of-the-art algorithm on UIUC sports dataset. The blue bar in the last panel is the performance of “pseudo” Object Bank representation extracted from the same number of “pseudo” object detectors. The values of the parameters in these “pseudo” detectors are generated without altering the original detector structures. In the case of linear classifier, the weights of the classifier are randomly generated from a Gaussian distribution instead of learned. “Pseudo” Object Bank is then extracted with exactly the same setting as Object Bank. Right: Classification performance on UIUC sports event dataset by using different appearance models: UIUC-22, UIUC-25, and ImageNet-177. Numbers at the top of each bar indicates the corresponding feature dimension.</p>	127

5.19	Comparison of different models and response maps generated. Column 1: Models visualized by using learned weights of histogram of oriented gradients. Here, “Random Gaussian” represents a randomly generated model by using random numbers sampled from a Gaussian distribution. “Best Gaussian” refers to the randomly generated model which performs best in classifying images containing “sailboat” from other images in the UIUC sports dataset. Column 2-4: Original images and the corresponding response maps. Each row corresponds to the response maps of images in the first row generated by the model showed in the first column. . . .	128
5.20	Most related scene type for each object. Rows are objects and column represent scene types. Classification scores of individual objects are used as the measurement of relationship between objects and scene types. The higher the classification accuracy, the more transparent the mask is in the intersecting grid of the object and scene type.	130
5.21	Relationship of objects. Classification scores of individual objects are used as the feature to measure the distance among objects.	132
5.22	Left: Best classification performance of projected representations by using different pooling methods. All dimensions are below 150. Middle: Training time comparison of the original Object Bank and the compressed Object Bank using OB-Max as an example. Right: Testing time comparison of the original Object Bank and the compressed Object Bank for each image.	134
5.23	Classification performance of different pooling methods for dimension reduction. We select feature dimensions corresponding to the view point with higher average value, maximum value and maximum variance respectively for classification. This corresponds to 1/2 dimension reduction.	134

6.1	(a) Classification performance (and s.t.d.) w.r.t number of training images. Each pair represents performances of LR1 and LRG respectively. X-axis is the ratio of the training images over the full training dataset (70 images/class). (b) Classification performance w.r.t feature dimension. X-axis is the size of compressed feature dimension, represented as the ratio of the compressed feature dimension over the full Object Bank representation dimension (44604). (c) Same as (b), represented in Log Scale to contrast the performances of different algorithms. (d) Classification performance w.r.t number of object filters. X-axis is the number of object filters. 3 rounds of randomized sampling is performed to choose the object filters from all the object detectors.	140
6.2	Object-wise coefficients given scene class. Selected objects correspond to non-zero β values learned by LRG.	143
6.3	Illustration of the learned β^{OF} by LRG1 within an object group. Columns from left to right correspond to “building” in “church” scene, “tree” in “mountain”, “cloud” in “beach”, and “boat” in “sailing”. Top Row: weights of Object Bank dimensions corresponding to different scales, from small to large. The weight of a scale is obtained by summing up the weights of all features corresponding to this scale in β^{OF} . Middle: Heat map of feature weights in image space at the scale with the highest weight (purple bars above). We project the learned feature weights back to the image by reverting the Object Bank representation extraction procedure. The purple bounding box shows the size of the object filter at this scale, centered at the peak of the heat map. Bottom: example scene images masked by the feature weights in image space (at the highest weighted scale), highlighting the most relevant object dimension.	144

7.1	<p>Top: an illustration of MUSIC for inferring a compact image code from high-dimensional Object Bank features. Here, $\mathbf{x}_o \in \mathbb{R}^G$ is the response of an object filter in the original OB, \mathbf{s}_o is the code of object o whose dimension is much lower than G (Sec. 7.3.2), θ is the image code (Sec. 7.3.2) that aggregates \mathbf{s}_o to achieve a single compressed representation for entire image, and B_o represents all the bases needed for reconstructing signals from object o. In B_o, the colored grids (column-wise) represent object-specific bases while the shaded grids represents shared bases. Across all variables, grids in the same color are directly correlated. Bottom: The learned structured object dictionary β. We show one example basis for each object and one shared basis. (This figure is best viewed in colors and with pdf magnification.)</p>	151
7.2	<p>(a) A heat matrix of the bases in the structured dictionary. Each column corresponds to a basis, and each row corresponds to a spatial location (i.e., a grid), which are grouped as “Top”, “Middle” and “Bottom” locations in the image. Object names are displayed at the bottom. A high value of a basis-element in a row indicates that the object is likely to appear in the corresponding grid. The values of each object basis are standardized for salient visualization. (b) A heat matrix of the average image codes θ of images from different classes are displayed on the right. Names of the object-specific bases are displayed at the bottom.</p>	158
7.3	<p>(a) Comparison of classification performance to the methods that use existing low-level representations or the original Object Bank representation and state-of-the-art approaches on UIUC sports data. (b) Comparison of classification performance to the methods that use existing low-level representations, the original Object Bank representation, and state-of-the-art approaches on MIT Indoor data.</p>	162
7.4	<p>Left: Content based image retrieval: precision of the the top ranked images by using GIST, BOW, SPM, original OB, and image code on the UIUC sports event dataset. Cosine distance is used as the distance measurement. Right: Average precision of the top N images in Caltech 256 dataset. “Cq1Rocchio” and “Csvm” are obtained by applying Rocchio algorithm [27] and SVM to the Classesemes [131], whereas “BoWRocchio” and “BoWsvm” are from Bag-of-Words representation. Performance scores are cited from Fig.4 in [131]</p>	163

7.5 Example image annotation results by MUSIC. Proposed tags are listed on the right side of the image. Incorrect tags are highlighted in red. The average number of tags proposed is ~ 10 . For those images with more than 7 tags predicted, only the top 7 annotated tags with highest empirical frequencies in the tag list of that image are shown. 165

Part I

Introduction

One of the most exciting revolutions in recent human history is the information revolution. The largest component of the digital universe is images, captured by more than 1 billion devices in the world, from digital cameras and camera phones to medical scanners and security cameras. With this exponential growth of image data, an important question that faces today's computer engineers and scientists is how to take advantage of this resource to further human knowledge and advance human society. Successful solutions to this question have numerous applications including automatic multimedia library indexing, retrieval and organization, inferring social interaction through seamless sharing photos, educational, and clinical assistive technology, and security systems.

Visual signals are notoriously complex and variable due to both photometric changes of images (such as varying illuminations and shadows) as well as geometric changes (such as view point variations, occlusions, etc.). Nevertheless, humans can readily perceive the semantic meaning of an image: what objects are present, what the scene environment is, and what kind activities are happening. It brings forward the problem of semantic image understanding, i.e. developing computer vision algorithms to effectively extract useful human understandable meaning from the vast amount of visual information.

In this work, we focus on using rigorous machine learning techniques to tackle the challenging problems of large-scale semantic image understanding, especially focusing on model learning and image representation. We develop algorithms that greatly benefited from the use of large-scale noisy Internet images. On the model representation side, we have been working on modeling images at multiple depths: object recognition, scene understanding and hierarchical image structure learning, which we discuss in Part II. On the image representation side, we propose a fundamentally novel high level image representation for high level visual recognition tasks. We elaborate the discussion of this high level image representation in Part III.

In Chapter 1, we discuss a fundamental task in image understanding, object recognition. Abundant data helps to train a robust recognition system, and a good recognition system can help in collecting a large number of relevant web images which can be subsequently be used to refine the system. Humans continuously update the

knowledge of objects when new examples are observed. We propose a novel object recognition algorithm called OPTIMOL by adapting a non-parametric latent topic model and an incremental learning framework emulating the human learning process. This algorithm is capable of automatically collecting much larger object category datasets while learning robust object category models from noisy Internet images and performing meaningful image annotation in real world scenarios. OPTIMOL is one of the pioneering work in harnessing noisy Internet resources towards large-scale recognition of real-world cluttered images.

While recognizing isolated objects and object classes is a critical component of visual recognition, a lot more is needed to achieve a complete understanding of a visual scene. Take a picture of a polo game as an example. Often within a single glance, humans are able to classify this image as a polo game (high-level image classification), recognize different objects such as horse and grass within the scene (annotation), and localize and delineate where the objects are in the scene (segmentation). Classification, annotation and segmentation are each by itself a challenging task. However, they are mutually beneficial to each other. Leveraging on this fact, in Chapter 2, we discuss our unified framework to recognize, annotate and segment the objects within an image, allowing for improved scene categorization performance. It is the first principled graphical model that tackles these three very challenging vision tasks in one framework. Our approach is the pioneering work in event and complex scene understanding in static images. Learning scalability is a critical issue when considering practical applications of computer vision algorithms. We design a framework to tackle the challenging recognition problems on real world Internet images, which offers, for the first time, a principled method to account for noise related to either erroneous or missing correspondences between visual concepts and textual annotations. It performs automatic learning from Internet images and tags, hence offering a scalable approach with no additional human labor.

Semantic image understanding of individual images is helpful for inferring the relationship among images based on visual content. A meaningful image hierarchy can ease the human effort in organizing thousands and millions of pictures (e.g., personal albums). Two types of hierarchies have recently been explored in computer

vision for describing the relationship among images: language-based hierarchy and low-level visual feature-based hierarchy. Pure language-based lexicon taxonomies, such as WordNet [103], are useful to guide the meaningful organization of images. However, they ignore important visual information that connects images together. On the other hand, purely visual feature-based hierarchies [7, 123] are difficult to interpret, and arguably not as useful. In Chapter 3, we propose to automatically construct a semantically and visually meaningful hierarchy of texts and images on the Internet. We introduce a non-parametric hierarchical model which jointly models the images and their textual counterparts. The hierarchical model encourages a flexible data structure imposed by its non-parametric property. The quality of the hierarchy is quantitatively evaluated by human subjects. Furthermore, we demonstrate that a good image hierarchy can serve as a knowledge ontology for end tasks such as image retrieval, annotation and classification.

Besides the model representation which has shown effectiveness in representing our complex visual world, another important problem we address in this thesis is the image feature representation. Any visual recognition task using computer vision algorithms starts with feature representation, the process of turning pixels into a vector of numbers for further computation and inference. A great deal of research has been conducted in this area, most of which are low level based image representations such as some variant of image gradients, textures and/or colors. Robust low-level image features have been proven to be effective representations for a variety of tasks such as object recognition and scene classification; however, such image representation carry little semantic meanings, creating what is known as a semantic gap for high-level visual tasks. In Part III, we propose a high-level image representation, called the Object Bank, where an image is represented as a scale-invariant response map of a large number of pre-trained generic object detectors, universally applicable to any testing dataset or visual task. Object Bank is a fundamentally new image representation, which encodes semantic and spatial information of the objects within an image. It is a sharp departure from all previous image representations and provides essential information for semantic image understanding. Semantically meaningful information of images can be effectively inferred based upon their Object Bank representation

in an unsupervised fashion. Using very simple, of-the-shelf classifiers such as linear support vector machines and logistic regression, we show that this high-level image representation can be used effectively for high level visual tasks including object and scene image classification, image annotation, content based image retrieval and semantic based image retrieval. The results are superior to reported state-of-the-arts performance on a number of standard benchmark datasets. To achieve more efficient and scalable representation while discover semantically meaningful feature patterns of complex scene images, we propose to perform content-based compression on Object Bank via a regularized logistic regression method and an unsupervised feature learning method.

Part II

Probabilistic Models for Semantic Image Understanding

The proliferation of digital camera allows people to easily capture the exciting moments in their lives, upload to online photo sharing websites, and share with friends. With the rapid growth of the Internet and digital camera, large amount of visual data such as images has emerged as a central player of the information age. It also confronts vision researchers the problem of finding effective ways to navigate the vast amount of visual information in the digital world. To solve this problem, semantic image understanding plays a vital role. We introduce the probabilistic models we developed to tackle semantic image understanding of challenging real-world images in this part. Specifically, in Chapter 1, we discuss a novel approach for object recognition, an important classical problem in semantic image understanding, by harnessing the vast amount of images on the Internet despite their noisiness. In Chapter 2, we introduce a more ambitious and less investigated problem: total scene understanding towards a complete understanding of images. Lastly, in Chapter 3, we extend the probabilistic model in total scene understanding to a non-parametric hierarchical one to infer image relationships among large scale user photos and automatically construct meaningful hierarchy encoded with semantic meaning and visual similarity.

Chapter 1

Learning object model from noisy Internet images

One of the holy grail of computer vision research is object recognition, i.e., describe the identities of objects and localize them in the images. Over the years, significant amount of efforts have been paid to develop algorithms for learning and modeling generic objects [107, 13, 43, 48, 51, 54, 83, 86, 87]. In order to develop effective object categorization algorithms, researchers rely on a critical resource: an accurate object class dataset. A good dataset serves as training data as well as an evaluation benchmark. A handful of large scale datasets currently serve such a purpose, such as *Caltech101/256* [41][64], the UIUC car dataset [2], LotusHill [145], LableMe [119], [34] etc. Sec. 1.1 will elaborate on the strengths and weaknesses of these datasets. In short, all of them, however, have a rather limited number of images and offer no possibility of expansion other than with extremely costly manual labor. Considering the numerous types of objects exist in our world and how fast the number grows everyday, how the traditional object recognition algorithms going to be scalable to recognize them with the existing datasets?

The explosion of the Internet provides us with a tremendous amount of images shared online, which can serve as abundant training data for training the object models. However, this data does not come for free. Type the word “airplane” in your favorite Internet search image engine, say Google Image (or Yahoo!, Bing, flickr.com,

etc.). What do you get? Of the thousands of images these search engines return, only a small fraction would be considered good airplane images ($\sim 15\%$ [51]). It is fair to say that for most of today’s average users surfing the web for images of generic objects, the current commercial state-of-the-art results are far from satisfying.

So far the story is a frustrating one. We are facing a chicken and egg problem here: Users of the Internet search engines would like better search results when looking for objects; developers of these search engines would like more robust visual models to improve these results; vision researchers are developing visual object models for this purpose; but in order to do so, it is critical to have large and diverse object datasets for training and evaluation; this, however, goes back to the same problem that the users face.

Lots of progress have been made on this problem in the past decade. Among the solutions, one of the major trends is to manually collect and annotate a ground truth dataset (LotusHill [145], LableMe [119] and Pascal Challenge [1]). Due to the vast number of object classes in our world, however, manually collecting images for all the classes is currently impossible. Recently, researchers have developed approaches utilizing images retrieved by image search softwares to learn statistical models to collect datasets automatically. Yet, learning from these images is still challenging:

- Current commercial image retrieval software is built upon text search techniques using the keywords embedded in the image link or tag. Thus, retrieved image is highly contaminated with visually irrelevant images. Extracting the useful information from this noisy pool of retrieved images is quite critical.
- The intra-class appearance variance among images can be large. For example, the appearance of wrist watches are different than the pocket watches in the watch category. The ability of relying on knowledge extracted from one of them (e.g. wrist watch) to distinguish the other (e.g. pocket watch) from unrelated images is important.
- Polysemy is common in the retrieved images, e.g. a “mouse” can be either a “computer mouse” or an “animal mouse”. An ideal approach can recognize the different appearances and cluster each of the objects separately.

In this work, we provide a framework to simultaneously learn object class models

and collect object class datasets. This is achieved by leveraging on the vast amount of images available on the Internet. The sketch of our idea is the following. Given a very small number of seed images of an object class (either provided by a human or automatically), our algorithm learns a model that best describes this class. Serving as a classifier, the algorithm can extract from the text search result those images that belong to the object class. The newly collected images are added to the object dataset, serving as new training data to improve the object model. With this new model, the algorithm can then go back to the web and extract more relevant images. Our model uses its previous prediction to teach itself. This is an iterative process that continuously gathers an accurate image dataset while learning a more and more robust object model. We will show in our experiments that our automatic, online algorithm is capable of collecting object class datasets of more images than *Caltech 101* [41] or LabelMe [119]. To summarize, we highlight here the main contributions of our work.

- We propose an iterative framework that collects object category datasets and learns the object category models simultaneously. This framework uses Bayesian non-parametric model as its theoretical base.
- We have developed an incremental learning scheme that uses only the newly added images for training a new model. This memory-less learning scheme is capable of handling an arbitrarily large number of images, which is a vital property for collecting large image datasets.
- Our experiments show that our algorithm is capable of both learning highly effective object category models and collecting object category datasets significantly larger than that of *Caltech 101* or LabelMe.

1.1 Related works

Image Retrieval from the Web: Content-based image retrieval (CBIR) [148, 35, 24, 90, 26, 74, 6, 5, 76] has been long an active field of research. One major group of research [6, 5, 76] in CBIR treats images as a collection of blobs or blocks, each

corresponding to a word or phrase in the caption (with some considerable variations). The goal of such algorithms is to assign proper words and/or phrases to a new image, and hence to retrieve similar ones in a database that contains such annotations. Another group of approaches focuses on comparing the query image with exemplar images and retrieving images based on image similarity [24, 26, 35]. However, our work is significantly different from the conventional frameworks of CBIR. Instead of learning to annotate images with a list of words or comparing the similarity of images, our algorithm collects the most suitable images from the web resources given a single word or phrase. One major difference between our work and the traditional CBIR is the emphasis on visual model learning. When collecting images of a particular object category, our algorithm continues to learn a better and better visual model to classify this object.

A few recent approaches in this domain are closer to our current framework. Feng and Chua propose a method to refine images returned by search engine using co-training [50]. They employ two independent segmentation methods as well as two independent sets of features to co-train two “statistically independent” SVM classifiers and co-annotate unknown images. Their method, however, does not offer an incremental training approach to boost the training efficiency in the co-train and co-annotate process. Moreover, their approach needs user interaction at the beginning of training and also when both the classifiers are uncertain about the decision.

Berg and Forsyth [11] develop a lightly supervised system to collect animal pictures from the web. Their system takes advantage of both the text surrounding the web images and the global feature statistics (patches, colors, textures) of the images to collect a large number of animal images. Their approach involves a training and a testing stage. In the training stage, a set of visual exemplars are selected by clustering the textual information. In the testing stage, textual information as well as visual cues extracted from these visual exemplars are incorporated in the classifier to find more visually and semantically related images. This approach requires supervision to identify the clusters of visual exemplars as relevant or background. In addition to this, there is an optional step for the user to swap erroneously labeled exemplars between the relevant and background topics in training.

Similar to [11], Schroff et al. [120] also employ the web meta data to boost the performance of image dataset collection. The images are ranked based on a simple Bayesian posterior estimation, i.e. the probability of the image class given multiple textual features of each image. A visual classifier, trained on the top ranked images, is then applied to re-rank the images.

Another method close in spirit to ours is by Yanai and Barnard [143]. They also utilize the idea of refining web image result with a probabilistic model. Their approach consists of a collection stage and a selection stage. In the collection stage, they divide the images into relevant and unrelated groups by analyzing the associated HTML documents. In the selection stage, a probabilistic generative model is applied to select the most relevant images among those from the first stage. Unlike ours, their method focuses on image annotation. Furthermore, their experiments show that their model is effective for “scene” concepts but not for “object” concepts. Hence, it is not suitable for generic object category dataset collection.

While the three approaches above rely on both visual and textual features of the web images returned by search engines, we would like to focus on visual cue only to demonstrate how much it can improve the retrieval result.

Multiple instance learning (MIL) research [97, 147, 133] has recently been explored in image retrieval and classification. MIL has shown effectiveness in learning from noisy Internet images in a weakly supervised category learning setting [133]. Similar to our framework, it takes advantage of the imperfect textual based image retrieval result of the traditional image search engines while properly accounting for the anticipated noise and ambiguity in the retrieval result. Our method significantly different than these MIL based approaches in our selective learning of related images while discarding the noisy images which constitutes majority of the search engine retrieved images. Therefore our framework is much more efficient in the scenario of learning from noisy Internet images and is capable of collecting large scale image dataset.

Finally, our approach is inspired by two papers by Fergus et al. [51, 53]. They introduce the idea of training a good object class model from web images returned by search engines, hence obtaining an object filter to refine these results. [53] extends the constellation model [140, 52] to include heterogeneous parts (e.g. regions of pixels

and curve segments). The extended model is then used to re-rank the retrieved result of image search engine. In [51], the authors extend a latent topic model (pLSA) to incorporate spatial information. The learned model is then applied to classify object images and to re-rank the images retrieved by Google image search. Although these two models are accurate, they are not scalable. Without an incremental learning framework, they need to be re-learned with all available images whenever new images are added.

All the above techniques achieve better search results by using either a better visual model or a combination of visual and text models to re-rank the rather noisy images from the web. We show later that by introducing an iterative framework of incremental learning, we are able to embed the processes of image collection and model learning efficiently into a mutually reinforcing system.

Object Classification: The recent explosion of object categorization research makes it possible to apply such techniques to partially solve many challenging problems such as improving the image search result and product images organization. Due to the vast number of object categorization approaches [43, 48, 51, 54, 83, 86, 87, 94], it is out of the scope of this thesis to discuss all of them. Here we will focus on two major branches that are closely related to our approach, specifically, latent topic model based on the “bag of words” representation and incremental learning of statistic models.

A number of systems based on the bag of words model representation have shown to be effective for object and scene recognition [51, 122, 46, 127, 18, 31, 124]. Sivic et al. [122] apply probabilistic Latent Semantic Analysis (pLSA), a model introduced in the statistical text literature, to images. Treating images and categories as documents and topics respectively, they model an image as a mixture of topics. By discovering the topics embedded in each image, they can find the class of the image. pLSA, however, can not perform satisfactorily on unknown testing images since it is not a well defined generative model. Furthermore, the number of parameters in pLSA grows linearly with the number of training images, making the model prone to overfitting. Fei-Fei et al. [46] apply an adapted version of a more flexible model called Latent Dirichlet Allocation (LDA) model [17] proposed by Blei et al. to natural scene categorization. LDA overcomes problems of pLSA by modeling the topic mixture proportion as a

latent variable regularized by its Dirichlet hyper-parameter.

The models mentioned above are all applied in a batch learning scenario. If the training data grows, as in our framework, they have to be retrained with all previous data and the new data. This is not an efficient approach, especially when learning from large datasets. Hence, we would like to apply incremental learning to our model.

A handful of object recognition approaches have applied incremental learning to object recognition tasks. The most notable ones are [83] and [41]. Krempp et al. [83] use a set of edge configurations as parts, which are learned from the data. By presenting the object categories sequentially to the system, it is optimized to accommodate the new classes by maximally reusing parts. Fei-Fei et al. [41] adopt a generative probabilistic model called constellation model [140, 52] to describe the object categories. Following Neal and Hinton's adaptation of conventional EM [105], a fully Bayesian incremental learning framework is developed to boost the learning speed.

Our approach combines the merits of these two branches:

- Bag of words representation enables the model to handle occlusion and rotation, which are common for web images. It is also computationally efficient, a desired property for the computation of large image dataset. On the other hand, latent topic model provides natural clustering of data, which helps solving the polysemy problem in image retrieval. We choose a nonparametric latent topic model so that the model can adjust its internal structure, specifically the number of clusters of the data, to accommodate new data.
- Given large intra-class variety of the online images, it is difficult to prepare good training examples for every subgroup of each image class. We employ an iteratively learning and classification approach to find the good training examples automatically. In each iteration, the object model is taught by its own prediction. In such iterative process, incremental learning is important to make learning in every iteration more efficient.

Object Datasets: One main goal of our proposed work is to suggest a framework that can replace most of the current human effort in object dataset collection. A few

popular object datasets exist today as the major training and evaluation resources for the community such as *Caltech 101* and LabelMe. *Caltech 101* consists of 101 object classes each of which contains 31 to 800 images [41]. It was collected by a group of students spending on average three or four hours per 100 images. While it is regarded as one of the most comprehensive object category datasets, it is limited in terms of the variation in the images (big, centered objects with few viewpoint changes), numbers of images per category (at most a few hundred) as well as the number of categories. For a long time, datasets are collected in this way relying on extensive human labor. Similar datasets are Caltech-256 [64], PASCAL [1], LotusHill [145] and Fink et. al [57]).

Recently, LabelMe has offered an alternative way of collecting datasets of objects by having users upload their images and label them [119]. This dataset is much more diverse than *Caltech 101*, potentially serving as a better benchmark for object detection algorithms. But since it relies on people uploading pictures and making uncontrolled annotations, it is difficult to use it as a generic object dataset. In addition, while some classes have many images (such as 20304 images for “car”), others have too few (such as 7 images for “watch”).

A few other object category datasets such as [34, 2] are also used by researchers. All of the datasets mentioned above require laborious human effort to collect and select the images. In addition, while serving as training and test datasets for researchers, they are not suitable for general search engine users. Our proposed work offers a first step towards a unified way of automatically collecting data useful both as a research dataset as well as for answering user queries.

1.2 Overview of the OPTIMOL Framework

We would like to tackle simultaneously the problem of model learning for object categories and automatic dataset collection, taking advantage of the vast but highly contaminated data from the web. We use Fig. 1.1 and Alg.1 to illustrate the overall framework of OPTIMOL. For every object category we are interested in, say, “panda”, we *initialize* our image dataset with a handful of seed images. This can be done either

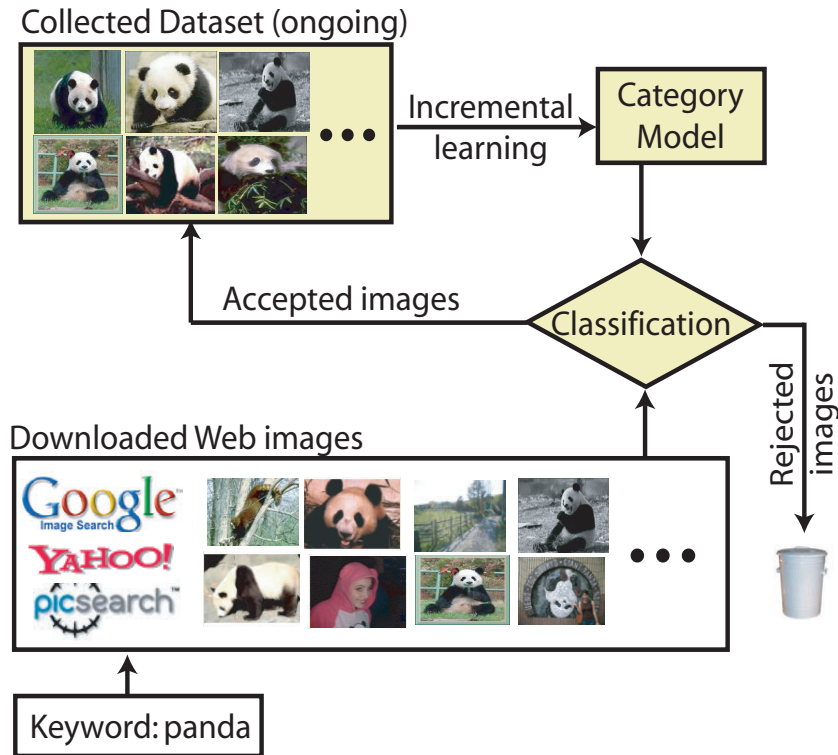


Figure 1.1: Illustration of the framework of the Online Picture collection via Incremental Model Learning (OPTIMOL) system. This framework works in an incremental way: Once a model is learned, it can be used to classify images from the web resource. The group of images classified as being in this object category are regarded as related images. Otherwise, they are discarded. The model is then updated by a subset of the newly accepted images in the current iteration. In this incremental fashion, the category model gets more and more robust. As a consequence, the collected dataset becomes larger and larger.

manually or automatically¹. With this small dataset, we begin the iterative process of model learning and dataset collection. *Learning* is done via an incremental learning process that we introduce in Sec. 1.3.2. Given the current updated model of the object class, we perform a binary *classification* on a subset of images downloaded from the web (e.g. “panda” vs. background). If an image is accepted as a “panda” image based on some statistical criteria (see Sec. 1.3.2), we *augment* our existing “panda” dataset by appending this new image. We then update our “panda” model

¹To automatically collect a handful of seed images, we use the images returned by the first page of Google image search, or any other state-of-the-art commercial search engines given the object class name as query word.

Algorithm 1 Incremental learning, classification and data collection

Download from the Web a large reservoir of images obtained by searching with keyword(s)
Initialize the object category dataset with seed images (manually or automatically)
repeat
 Learn object category model with the latest accepted images to the dataset
 Classify a subset of downloaded images using the current object category model
 Augment the dataset with accepted images
until user satisfied or images exhausted

with a subset of the newly accepted images (see Sec. 1.3.3 for details of the “cache set”). Note that the already existing images in the dataset no longer participate in this iteration of learning. In the meantime, the background model will also be updated using a constant resource of background images². We *repeat* this process till a sufficient dataset is collected or we have exhausted all downloaded images.

1.3 Theoretical Framework of OPTIMOL

1.3.1 Our Model

In this section, we describe the model used in OPTIMOL in detail. Specifically, we first review the probabilistic classification approaches especially generative models. We then introduce briefly the “bag of words” image representation combined with the latent topic model. Finally, we discuss the nonparametric latent topic model (i.e. Hierarchical Dirichlet Process (HDP)) in OPTIMOL.

Generative Model

Classification approaches can be grossly divided into generative models, discriminative models and discriminant functions [14]. For generative models, such as Gaussian mixture models ([140, 52, 40]), Markov random fields [84], latent topic model

²The background class model is learnt by using a published “background” image dataset [52, 42]. The background class model is updated together with the object class model. In this way, it can accommodate the essential changes of the new training data.

([122, 46, 139]) etc., both the input distribution and the output distribution are modeled. While for discriminative models, which include boosting ([59, 60]), support vector machines [19], conditional random field [100] etc., the posterior probabilities are modeled directly. The simplest approaches are called discriminant functions (e.g. Fisher’s linear discriminant [8]), which are projections mapping the input data to class labels. Comparing to the other two approaches, generative models are able to handle the missing data and noisy data problems better since all variables are jointly modeled in a relatively equal manner. When such problems are encountered, the performance will not be affected dramatically. This property is desired for semi-supervised learning from Internet images where only a small amount of labeled data is provided. Here, we would like to adopt generative model given this ideal property for OPTIMOL’s iterative incremental learning framework. Previous success of generative model in object recognition [40, 122] and content based image retrieval [143, 51] ensure the potential ability of generative model in our framework.

Object category model

We would like to emphasize that our proposed framework is not limited to the particular object model used here. Any model that can be cast into an incremental learning framework is suitable for our protocol. Of the many possibilities, we have chosen to use a variant of the HDP (Hierarchical Dirichlet Process) [129] model based on the “bag of words” [122, 46, 31, 124] representation of images. HDP is particular suitable here because of the natural clustering and computationally efficient properties respectively. “Bag of words” model is frequently used in natural language processing and information retrieval of text documents. In “bag of words” model, each document is represented as an unordered collection of words. When applied to image representation, it describes each image as a bag of visual words (node x in Fig. 1.2). We are particularly interested in the application of latent topic models to such representation [69, 17, 129]. Similar to [126, 139], we adapt a nonparametric generative model, Hierarchical Dirichlet process (HDP) [129], for our object category model. Compared to parametric latent topic models such as LDA [17] or pLSA [69], HDP offers a way

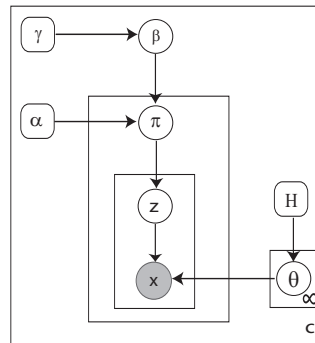


Figure 1.2: **Graphical model of HDP.** Each node denotes a random variable. Bounding boxes represent repetitions. Arrows indicate conditional dependency. Dark node indicates it is observed.

to sample an unbounded number of latent topics, or clusters, for each object category model. This property is especially desirable for OPTIMOL. Since the data for training keeps growing, we would like to retain the ability to “grow” the object class model when new clusters of images arise. Before we move on to introduce the HDP object category model in more detail, we define the notations in Fig. 1.2 here.

- A *patch* x is the basic unit of an image. Each patch is represented as a codeword of a visual vocabulary of codewords indexed by $\{1, \dots, T\}$.
- An *image* is represented as N unordered patches denoted by $\mathbf{x} = (x_{j1}, x_{j2}, \dots, x_{jN})$, where x_{ji} is the i th patch of the j th image.
- A *category* is a collection of I images denoted by $\mathbf{D} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_I)$.

Hierarchical Dirichlet process

We represent an image as a document constituted by a bag of visual words. Each category consists of a variable number of latent topics corresponding to clusters of images with similar visual words attributes. We model both object and background classes with HDP [129]. Fig. 1.2 shows the graphical model of HDP. In the HDP model, θ corresponds to the distributions of visual words given different latent topics shared among images. Let x_{ji} be the i th patch in j th image. For each patch x_{ji} , there is a hidden variable z_{ji} denoting the latent topic index. β is the stick-breaking

weights [121] and π_j represents the mixing proportion of z for the j th image. We now go through the graphical model (Fig. 1.2) and show how we generate each patch in an image. For each image class c ,

- Sample $\beta \sim \text{GEM}(\gamma)$. GEM is the stick-breaking process:

$$\beta'_k \sim \text{Beta}(1, \gamma) \quad \beta_k = \beta'_k \prod_{l=1}^{k-1} (1 - \beta'_l) \quad \beta = (\beta_1, \beta_2, \dots, \beta_\infty) \quad (1.1)$$

- Sample θ_k from the Dirichlet prior distribution H .
- Given the stick-breaking weights γ and global cluster θ , we generate each image in this class.
 - We first sample $\pi_j, \pi_j | \alpha, \beta \sim \text{DP}(\alpha, \beta)$. DP denotes the Dirichlet Process introduced by Ferguson in 1973 [55]:

$$\pi'_{jk} \sim \text{Beta} \left(\alpha \beta_k, \alpha \left(1 - \prod_{l=1}^k \beta_l \right) \right) \quad \pi_{jk} = \pi'_{jk} \prod_{l=1}^{k-1} (1 - \pi'_{jl}) \quad (1.2)$$

where $\pi_j = (\pi_{j1}, \pi_{j1}, \dots, \pi_{j\infty})$.

- Given π_j , we are ready to generate each image patch x_{ji}
 - * Sample the latent topic index z_{ji} for patch x_{ji} from a multinomial distribution $\pi_j: z_{ji} | \pi_j \sim \pi_j$
 - * Sample x_{ji} given z_{ji} from a class dependent multinomial distribution $F: x_{ji} | z_{ji}, \theta_k \sim F(\theta_{z_{ji}}^c)$

1.3.2 Learning

We have described our hierarchical model in details. We now turn to learning its parameters. In this subsection, we first describe the batch learning algorithm of our hierarchical model. We then introduce semi-supervised learning of this model. Finally, the efficient semi-supervised incremental learning is introduced for learning from large image dataset.

Markov Chain Monte Carlo sampling

In this section, we describe how we learn the parameters by Gibbs sampling [63] of the latent variables. We choose the *Chinese restaurant franchise* [129] metaphor to describe this procedure. Imagine multiple Chinese restaurants serving the same set of dishes in the menu. At each table of each restaurant, a dish is shared by the customers sitting at that table. Metaphorically, we describe each image as one restaurant and the local cluster for the customer x_{ji} as the table t_{ji} . Similarly, the global cluster for the t th table in the j th restaurant is represented as the dish k_{jt} :

$$t_{ji}|t_{j1}, \dots, t_{ji-1}, \alpha, G_0 \sim \sum_{t=1}^{T_j} n_{jt} \delta_{t_{ji}=t} + \alpha G_0 \quad (1.3)$$

$$k_{jt}|k_{11}, k_{12}, \dots, k_{21}, \dots, k_{jt-1}, \gamma \sim \sum_{k=1}^K m_k \delta_{k_{jt}=k} + \gamma H \quad (1.4)$$

where $G_0 \sim DP(\gamma, H)$. n_{jt} denotes the number of customers for table t . T_j is the current number of tables. m_k represents the number of tables ordered dish k . K denotes the current total number of dishes. All these statistics are calculated without considering the current data point. A new table and new dish can also be generated from G_0 and H , respectively, if current data does not fit in any of the previous table or dish. For standard mixture models, the *Chinese restaurant franchise* can be easily connected to the stick breaking process by having $z_{ji} = k_{jt}$.

Sampling the table. According to Eq.1.3 and Eq.1.4, the probability of a new customer x_{ji} being assigned to table t is:

$$P(t_{ji} = t | x_{ji}, t_{-ji}, \mathbf{k}) \propto \begin{cases} \alpha p_{t_{new}} & \text{for } t = t_{new} \\ n_{jt} f(x_{ji} | \theta_{k_{ji}}) & \text{for used } t \end{cases} \quad (1.5)$$

We have

$$p_{t_{new}} = \sum_{k=1}^K \frac{m_k}{\sum_{k=1}^K m_k + \gamma} f(x_{ji}|\theta_{k_{ji}}) + \frac{\gamma}{\sum_{k=1}^K m_k + \gamma} f(x_{ji}|\theta_{k_{new}})$$

$f(x_{ji}|\theta_{k_{ji}})$ is the conditional density of patch x_{ji} given all data items associated with k except itself. The probability of assigning a newly generated table t_{new} to a global cluster is proportional to:

$$\begin{cases} m_k f(x_{ji}|\theta_{k_{ji}}) & \text{for used } k \\ \gamma f(x_{ji}|\theta_{k_{new}}) & \text{for new } k \end{cases} \quad (1.6)$$

Sampling the global latent topic. For the existing tables, the dish can change according to all customers at that table. The global cluster k_{jt} can be obtained from:

$$\begin{cases} m_k f(\mathbf{x}_{jt}|\theta_{k_{jt}}) & \text{for used } k \\ \gamma f(\mathbf{x}_{jt}|\theta_{k_{new}}) & \text{for new } k \end{cases} \quad (1.7)$$

Where \mathbf{x}_{jt} represents all patches associated with image level mixture component t in image j except the current one. $f(\mathbf{x}_{jt}|\theta_{k_{jt}})$ is the conditional density of \mathbf{x}_{jt} given all patches associated with topic k except themselves. n_{jt} and m_k will be updated respectively regarding the table index and global latent topic assigned. Given $z_{ji} = k_{jt_{ji}}$, we in turn update $F(\theta_{z_{ji}}^c)$ for the category c .

Semi-supervised Learning

Due to the large variation of web images, it requires large number of representative images to train a robust model. Manually selecting these images is time consuming and biased. In the framework of OPTIMOL, we employ a semi-supervised learning approach, specifically self training, to propagate the initial knowledge [150]. As a wrapper algorithm, self training can be easily applied to existing models. It has

been used in natural language processing to perform tasks such as parsing strings of words [101]. In computational biology, self training is employed for gene prediction [12]. Recently, it is also applied in computer vision by Rosenberg et al. [118] to help object detection. All of the approaches show that, by employing a self training framework, one can achieve comparable result to state-of-the-art approach with less labeled training data. We will demonstrate later in Fig. 1.9 that with semi-supervised learning framework, OPTIMOL shows superior performance in comparison to the fully supervised learning framework using the same number of seed images. The basic idea of self training is:

- First, an initial model is trained with a limited amount of reliable labeled data.
- This model is applied to estimate the labels of the unlabeled data.
- The estimated labels is used to retrain the model.
- Repeat the training and classification procedure.

With this idea, self training allows the model to teach itself iteratively with new classification results. In each iteration of the self training, one can incorporate the new data to retrain the model either via the batch learning mode described at the beginning of Sec. 1.3.2 or an incremental learning mode introduced in the next paragraph. In the self training framework, data that are far away from the initial training set are unlikely to be selected to update the model. However, such data are very useful for generalization of the model. Thus, we design a “cache set” to solve this problem in Sec. 1.3.3.

Incremental learning of a latent topic model

Having introduced the object class model and the batched learning approach, we propose an incremental learning scheme for OPTIMOL. This scheme let OPTIMOL update the model at every iteration of the dataset collection process more efficiently. Our goal here is to perform incremental learning by using only new images selected at current iteration. We will illustrate in Fig. 1.9 (Middle) that this is much more efficient than performing a batch learning by using all images in the current dataset at

every iteration. Meanwhile, it still retains the accuracy of the model as shown in Fig. 1.9. Let Θ denote the model parameters, and I_j denote the j th image represented by a set of patches x_{j1}, \dots, x_{jn} . For each patch x_{ji} , there is a hidden variable z_{ji} denoting the latent topic index. The model parameters and hidden variable are updated iteratively using the current model and the input image I_j in the following fashion:

$$z_j \sim p(z|\Theta^{j-1}, I_j) \quad \Theta^j \sim p(\Theta|z_j, \Theta^{j-1}, I_j) \quad (1.8)$$

where Θ^{j-1} represents the model parameters learned from the previous $j - 1$ images. Neal & Hinton [105] provide a theoretical ground for incrementally learning mixture models via sufficient statistics updates. We follow this idea by keeping only the sufficient statistics of the parameters associated with the existing images in an object dataset. Learning is then achieved by updating these sufficient statistics with those provided by the new images. One straightforward method is to use all the new images accepted by the current classification criterion. But this method will favor those images with similar appearances to the existing ones, hence resulting in an over-specialized object models. To avoid such a problem, we take full advantage of the non-parametric HDP model by using a subset of the related images denoted as “cache set” to update our model. Here, “related images” refer to images classified as belonging to the object class by the current model. We detail the selection of the “cache set” in Sec. 1.3.3.

1.3.3 New Image Classification and Annotation

In the OPTIMOL framework, learning and classification are conducted iteratively. We have described the learning step in Sec. 1.3.2. In this subsection, we introduce the classification step in our framework. We first describe how our model judges which images are related images against others. Then we introduce the criterion to select the “cache set”, a subset of the related images to be used to train our model. Finally, we detail the annotation method at the end of this section.

Image Classification

For every iteration of the dataset collection process, we have a binary classification problem: classify unknown images as a foreground object or a background image. In the current model, we have $p(z|c)$ parameterized by the distribution of global latent topics given each class in the Chinese restaurant franchise and $p(x|z, c)$ parameterized by $F(\theta_z^c)$ learned for each category c . A testing image I is represented as a collection of local patches x_i , where $i = \{1, \dots, M\}$ and M is the number of patches. The likelihood $p(I|c)$ for each class is calculated by:

$$P(I|c) = \prod_i \sum_z P(x_i|z, c)P(z|c) \quad (1.9)$$

Classification decision is made by choosing the category model that yields the higher probability. From a dataset collection point of view, incorporating an incorrect image into the dataset (false positive) is much worse than missing a correct image (false negative). Hence, a risk function is introduced to penalize false positives more heavily:

$$\begin{aligned} R_i(A|I) &= \lambda_{Ac_f}P(c_f|I) + \lambda_{Ac_b}P(c_b|I) \\ R_i(R|I) &= \lambda_{Rc_f}P(c_f|I) + \lambda_{Rc_b}P(c_b|I) \end{aligned} \quad (1.10)$$

Here A represents acceptance of an image into our dataset. R denotes rejection. As long as the risk of accepting an image is lower than rejecting it, it is accepted. Image classification is finally decided by the likelihood ratio:

$$\frac{P(I|c_f)}{P(I|c_b)} > \frac{\lambda_{Ac_b} - \lambda_{Rc_b}}{\lambda_{Rc_f} - \lambda_{Ac_f}} \frac{P(c_b)}{P(c_f)} \quad (1.11)$$

where the c_f is the foreground category while the c_b is the background category. $\frac{\lambda_{Ac_b} - \lambda_{Rc_b}}{\lambda_{Rc_f} - \lambda_{Ac_f}}$ is automatically adjusted by applying the likelihood ratio measurement to a reference dataset³ at every iteration. New images satisfying Eq.1.11 are regarded

³To achieve a fully automated system, we use the original seed images as the reference dataset. As the training dataset grows larger, the direct effect of the original training images diminishes in terms of the object model. It therefore becomes a good approximation of a validation dataset.

as related images. They will be either appended to the permanent dataset or used to train the new model upon further criterion.

The Cache Set

In the self training setting, the model teaches itself by using the predicted related images. It is critical to distinguish random noisy images from difference caused by intra-class difference. How to extract the most useful information from the new classification result automatically? We use a “cache set” of images to incrementally update our model. The “cache set” is a less “permanent” set of good images compared to the actual image dataset. At each iteration, if all “good” images are used for model learning, it is highly likely that many of these images will look very similar to the previously collected images, hence reinforcing the model to be even more specialized in selecting such images for the next iteration. Furthermore, it will also be computationally expensive to train with all “good” images. So the usage of the “cache set” is to retain a group of images that tend to be more diverse than the existing images in the dataset. For each new image passing the classification criterion (Eq.1.11), it is further evaluated by Eq.1.12 to determine whether it should belong to the “cache set” or the permanent set.

$$H(I) = - \sum_z p(z|I) \ln p(z|I) \quad (1.12)$$

In Fig. 1.14, we demonstrate how to select the “cache set”. According to Shannon’s definition of entropy, Eq.1.12 relates to the amount of uncertainty of an event associated with a given probability distribution. Images with high entropy are more uncertain and more likely to have new topics. Thus, these high likelihood and high entropy images are ideal for model learning. In the meantime, images with high likelihood but low entropy are regarded as confident foreground images and will be incorporated into the permanent dataset.

Image Annotation

The goal of OPTIMOL is not only to collect a good image dataset but also to provide further information about the location and size of the objects contained in the dataset images. Object annotation is carried out by first calculating the likelihood of each patch given the object class c_f :

$$p(x|c_f) = \sum_z p(x|z, c_f)p(z|c_f) \quad (1.13)$$

The region with the most concentrated high likelihood patches is then selected as the object region. A bounding box is drawn to enclose the selected patches according Eq.1.13. Sample results are shown in Fig. 1.20.

1.3.4 Discussion of the Model

We discuss here several important properties of OPTIMOL in this section.

Dataset Diversity Our goal is to collect a diverse image dataset which has ample intra class variation. Furthermore, the ideal model should be capable of collecting all possible object classes associated with different semantic meanings of a polysemous word. OPTIMOL is able to achieve both goals given its facility of accommodating new training data different from the previous ones. This is largely attributed to the property of object model (i.e. HDP) that is capable of generating unbounded number of topics to describe data with different aspects. Later, we show in Fig. 1.10 that our framework can collect a large and more diverse image dataset compared to *Caltech 101*. Moreover, Fig. 1.11 demonstrates that OPTIMOL collects image in a semantic way by assigning visually different images to different clusters.

Concept Drift Self training helps OPTIMOL to accumulate knowledge without human interaction. However, it is prone to concept drift when the model is updated by unrelated images. The term ‘‘Concept Drift’’ refers to the phenomenon of a target variable changing over time. In the OPTIMOL framework, we are mostly concerned

with the object model drifting from one category to another (e.g. from accordions to grand pianos). To avoid model drift, our system needs to decide whether an image should be discarded, appended to the permanent dataset or kept in the “cache set” to retrain the model. Using a constant number threshold for Eq.1.11 to make this decision is not feasible since the model is updated in every iteration. The rank of the images is not a good choice either since there might not be any related images in current iteration. In the OPTIMOL framework, we use a threshold calculated dynamically by measuring the likelihood ratio of the updated model on a validation set. Those with likelihood ratio lower than the threshold are assumed to be “unrelated” and hence discarded. Among those “related” images, a proportion of images with high entropies are selected to be held in the “cache set” according to Eq.1.12. Fig. 1.3 shows the influence of the threshold on the number of images to be accepted, discarded or used for training. Basically, the number of images to be incorporated into the permanent dataset decreases along with the increase of the threshold. The same applies to the number of images to be held in the “cache set”. If fewer images are kept in the “cache set” and are used to update the model, the model tends to be similar to the initial model. In the extreme case, if the threshold equals 1, no image will be incorporated to the dataset. Neither will new images be used to retrain the model. The model will stay the same as the initial model. Hence the incorporated images will be highly similar to the initial dataset and the collected dataset will not be very diverse (Fig. 1.15 Left). To the other extreme, if the threshold equals 0, unrelated images will be accepted. Some of these unrelated images with high entropies will be used to update the model. In this scenario, self training will reinforce the error in each iteration and tend to drift. We demonstrate this concept drift issue by showing the dataset collection performance of OPTIMOL with different likelihood ratio thresholds in Fig. 1.13.

1.4 Walkthrough for the accordion category

As an example, we describe how OPTIMOL collects images for the “accordion” category following Alg.1 and Fig. 1.1. We use Fig. 1.4-1.8 to show the real system.

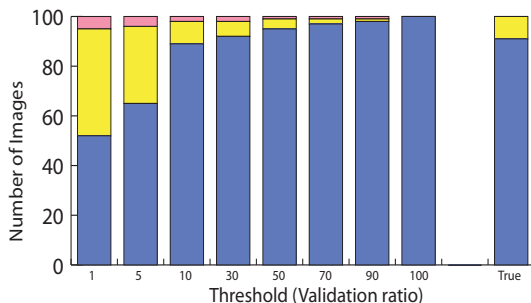


Figure 1.3: Influence of the threshold (Eq.1.11) on the number of images to be appended to the dataset and held in the cache set on 100 “accordion” images. x-axis is the value of threshold represented in percentile. The validation ratio thresholds are 1, 5, 10, 30, 50, 70, 90, 100, which are equivalent to -1.94, 2.10, 14.24, 26.88, 44.26, 58.50, 100.22 and 112.46 in log likelihood ratio thresholds respectively for the current classifier. y-axis denotes the number of images. Blue region represents number of images classified as unrelated. Yellow region denotes the number of images that will be appended to the object dataset. Pink region represents number of images held in the “cache set”. The “true” bar represents the proportion of true images in the 100 testing images. These bars are generated using the initial model learned from 15 seeds images. The higher the threshold is, the fewer number of images will be appended to the permanent dataset and held in the “cache set”.

- Step 1 (Downloading and preprocessing): As shown in Fig. 1.4, 1659 images are downloaded as our image pool by typing the query word “accordion” in image search engines such as Google image, Yahoo image and Picsearch. We use the first 15 images from the web resource as our seed images, assuming that most of them are related to the “accordion” concept. The remaining (non-seed) images are divided into 17 groups. The first 16 groups have 100 images each and the last, 17th group has 44 images. The OPTIMOL framework will process one group per iteration. Each image is represented as a set of unordered local patches. Kadir and Brady [82] salient point detector offers compact representations of the image, which makes computation more efficient for our framework. We apply this detector to find the informative local regions that are salient over both location and scale. Considering the diversity of images on the web, a 128-dim rotationally invariant SIFT vector is used to represent each region [94]. We build a 500-word codebook by applying K-means clustering to the 89058 SIFT vectors extracted from the 15 seeds images of each of the 23 object categories. Each patch in an image is then described by using the most similar codeword in the codebook via vector quantization. In Fig. 1.5, we show examples of detected regions of interest and some codeword samples.

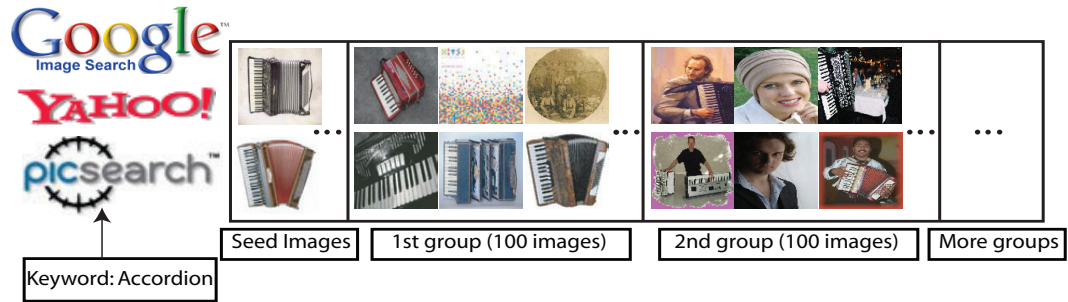


Figure 1.4: **Downloading part in Step 1.** A noisy “accordion” dataset is downloaded using “accordion” as query word in Google image, Yahoo! image and Picsearch. Downloaded images will be further divided into groups for the iterative process.

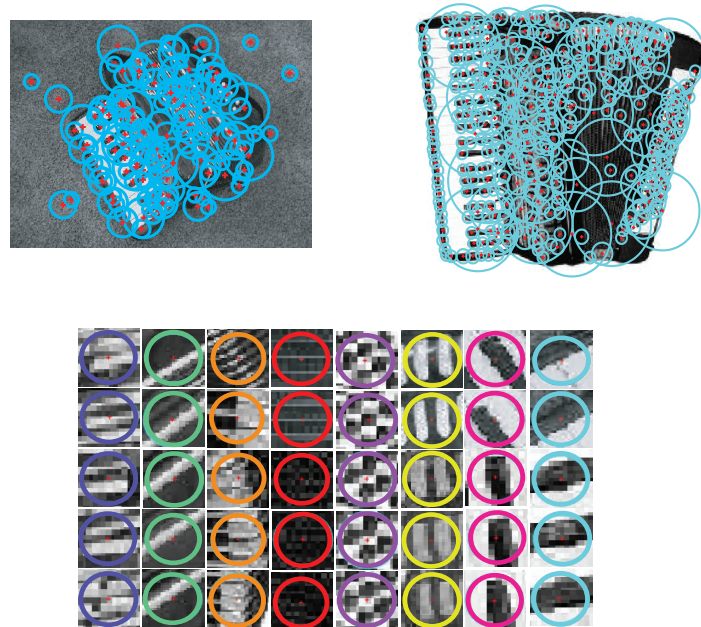


Figure 1.5: **Preprocessing in Step 1.** **Top:** Regions of interest found by Kadir&Brady detector. The circles indicate the interest regions. The red crosses are the centers of these regions. **Bottom:** Sample codewords. Patches with similar SIFT descriptors are clustered into the same codeword, which are presented using the same color.

- Step 2 (Initial batch learning): As shown in Fig. 1.6, a batch learning algorithm described in Sec. 1.3.2 is applied on the seed images to train an initial “accordion” model. Meanwhile, same number of background images are used to train a background model. In model learning, the hyper-parameters γ and α are constant numbers

1 and 0.01 respectively acting as smooth factors. We will show later in Fig. 1.12 how they influence the model. According to Eq.1.4, given a high γ value, we expect to obtain more dishes (global topics in a category) in the model. Similarly, following Eq.1.3, a higher α value populates more tables (local clusters in each image). After Step 2, we obtain a fairly good object category model, which can perform reasonable classification.

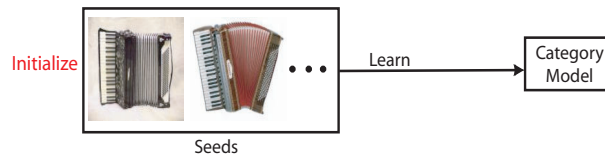


Figure 1.6: **Initial batch learning.** In the first iteration, a model is learned from the seed images. The learned model performs fairly well in classification as shown in Fig. 1.9.

- **Step 3 (Classification):** Models obtained from the learning step are used to classify a group of images from the pool, typically 100 images. By measuring the likelihood ratio, this group is divided into unrelated images and related images as shown in Fig. 1.7. The unrelated images will be discarded. Related images are further measured by their entropy. High entropy ones will be held in the “cache set”. Low entropy images will be appended to the permanent dataset. In classification, our threshold is selected as “30%”. This is a conservative choice that allows only the top 30% validation images with highest likelihood ratio to be classified as foreground images. This threshold is equivalent to likelihood ratio threshold 26.88. As shown in Fig. 1.3, this criterion agrees (conservatively) with the observation that an average estimate of 15% of images returned by the search engine are related to the query word(s). 10% of the related images with high entropies will be kept in the “cache set” for the incremental learning of the model. These images also participate in next 2 iterations in classification. After three iterations, images still left in the “cache set” will be discarded.
- **Step 4 (Incremental Learning)** As shown in Fig. 1.8, incremental learning is only applied to images held in the “cache set”. In the meantime, the same number of new background images are used to update the background model. In this step, we keep the same set of learning parameters as those in Step 2.

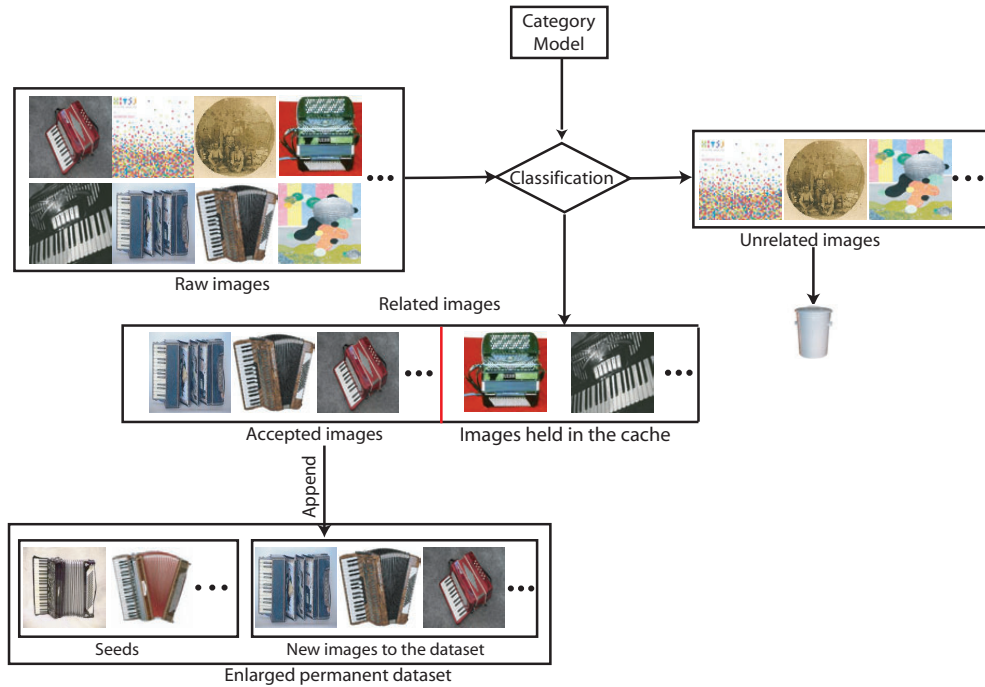


Figure 1.7: **Classification.** Classification is performed on a subset of raw images collected from the web using the “accordion” query. Images with low likelihood ratios measured by Eq.1.11 are discarded. For the rest of the images, those with low entropies are incorporated into the permanent dataset, while the high entropy ones stay in the “cache set”.

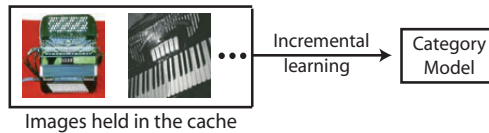


Figure 1.8: **Incremental Learning.** The model is updated using only the images held in the “cache set” from the previous iteration.

- Repeat Step 3 and 4 till the user terminates the program or images in the downloaded image pool are exhausted.

1.5 Experiments & Results

We conduct three experiments to demonstrate the effectiveness of OPTIMOL. Exp.1 consists of a set of analysis experiments.

- A performance comparison of the batch vs. incremental learning methods in terms of the number of collected images, processing time and the recognition accuracy.
- Diversity of our collected dataset. More specifically, comparison between the average images of our collected dataset and the *Caltech 101* dataset. In addition, we show average images of different clusters for the “accordion” and “euphonium” categories as examples to provide more insights into the model.
- Influence of the hyper-parameters γ and α on the model. γ and α control the number of global and local clusters of the images respectively.
- Dataset collection comparison of OPTIMOL using different likelihood threshold values to demonstrate the issue of concept drift.
- Illustration of images in the permanent dataset, the “cache set” and the “junk set”.
- Illustration of images collected by OPTIMOL using different entropy thresholds.
- Detection performance comparison of OPTIMOL using different numbers of seed images.
- Polysemous class analysis(a case study of the polysemous words “mouse” and “bass”).

Exp.2 demonstrates the superior dataset collection performance of OPTIMOL over the existing datasets. In addition to dataset collect, it also provides satisfying annotations on the collected images. Exp.3 shows that OPTIMOL is on par with the state-of-the-art object model learned from internet images [51] for multiple object categories classification.

We first introduce the various datasets used in the experiments. Then we show experiment settings and results for the three experiments respectively.

1.5.1 Datasets Definitions

We define the following four different datasets used in our experiments:

1. Caltech 101-Web & Caltech 101-Human

Two versions of the Caltech 101 dataset are used in our experiment. Caltech 101-Web is the original raw dataset downloaded from the web containing a large portion of visually unrelated images in each category. The number of images in each category varies from 113 (winsor-chair) to 1701 (watch). Caltech 101-Human is the clean dataset manually selected from Caltech 101-Web. The number of images in each category varies from 31 (inline-skate) to 800 (airplanes). By using this dataset, we show that OPTIMOL achieves superior retrieval performance to human labeled results.

2. Web-23

We downloaded 21 object categories from online image search engines by using query words randomly selected from object category names in Caltech 101-Web. In addition, “face” and “penguin” categories are included in Web-23 for further comparison. The number of images in each category ranges from 577 (stop-sign) to 12414 (face). Most of the images in a category are unrelated images (e.g. 352 true “accordions” out of 1659 images).

3. Princeton-23 [28]

This dataset includes the same categories as used in Web-23. However, it is a more diverse dataset which contains more images in every category. The images are downloaded using words generated by WordNet [103] synset as the query input for image search engines. To obtain more images, query words are also translated into multiple languages, accessing the regional website of the image search engines. The number of images in each category varies from 4854 (inline-skate) to 38937 (sunflower).

4. Fergus ICCV’05 dataset

A 7-Category dataset provided by [51]. Object classes are: airplane, car, face, guitar, leopard, motorbike and watch.

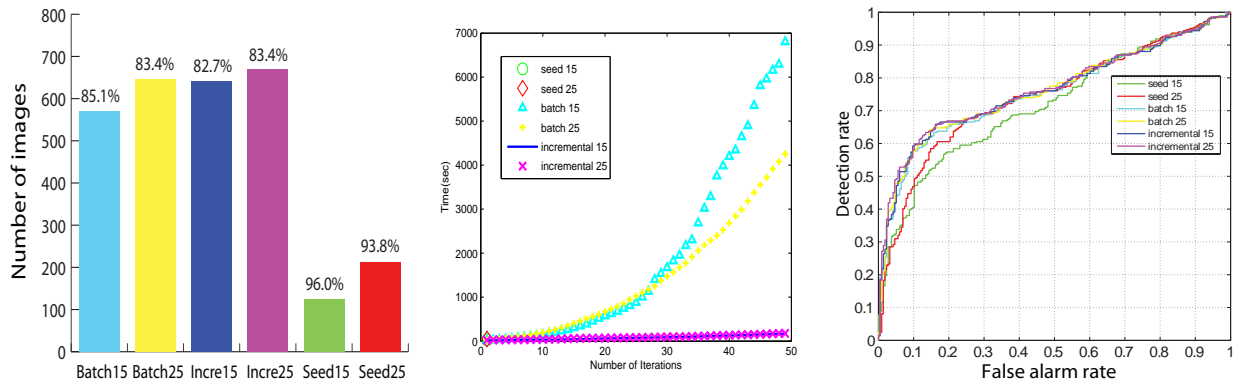


Figure 1.9: **Batch vs. Incremental Learning** (a case study of the “inline skate” category with 4835 images). **Left:** The number of images retrieved by the incremental learning algorithm, the batch learning algorithm and the base model. Detection rate is displayed on top of each bar. x-axis represents batch learning with 15 seed images, batch learning with 25 seed images, incremental learning with 15 seed images, incremental learning with 25 seed images, the base model learned from 15 seed images and 25 seed images respectively. **Middle:** Running time comparison of the batch learning method, the incremental learning method and the base model learned as a function of number of training iterations. The incrementally learned model is initialized by applying the batch learning algorithm on 15 or 25 training images, which takes the same amount of time as the corresponding batch method does. After initialization, incremental learning is more efficient compared to the batch method. **Right:** Recognition accuracy of the incrementally learned, batch learned models and the base model evaluated by using Receiver Operating Characteristic (ROC) Curves.

1.5.2 Exp.1: Analysis Experiment

Comparison of incremental learning and batch learning. In this experiment, we compare the computation time and accuracy of the incremental learning algorithm, the batch learning algorithm as well as the base model learned from initial seed images (Fig. 1.9). To keep a fair comparison, the images used for batch learning and incremental learning are exactly the same in each iteration. For all three algorithms, background models are updated together with the foreground models by using the same number of images. All results shown here are collected from the “inline skate” dataset; other datasets yield similar behavior. Fig. 1.9 (Left) shows that the incremental learning method is comparable to the batch method in the number of collected images. Both of them outperform the base model learned from the seed images only. Fig. 1.9 (Middle) illustrates that by incrementally learning from the new

images at every iteration, OPTIMOL is more computationally efficient than a batch method. Finally, we show a classification performance comparison among OPTIMOL, the batch method and the model learned from seed images in Fig. 1.9 (Right) by a set of ROC curves.

In our system, the image classifier evolves as the model gets updated in every iteration of the self training process. In the image dataset collection process, the newly updated classifier categorizes the current group of images into foreground images and background images. The testing images are therefore different in each iteration of the self training process. Evaluating different classifiers on different test image sets respectively provide little useful information of the classifier quality. A good classifier could perform poorly on a challenging dataset while a poor classifier might perform satisfactorily on a simple dataset. Thus, we only compare our model at the end of the incremental learning process with a model that is learned in a batch mode by testing both models on the same set of test images. We use an ROC curve to illustrate the classification result for each model, shown in Fig. 1.9. Classifier quality is measured by the area under its ROC curve. As demonstrated in Fig. 1.9 (Right), while batch learning and incremental approaches are comparable to each other in classification, both of them show superior performance over the base models trained by seed images only. In addition, Fig. 1.9 (Left) and Fig. 1.9 (Right) show that the number of seed images has little influence on the performances of the iterative approaches. This can be easily explained by the property of self training which teaches the model automatically by using the predicted result. Once a decent initial model is learned, self training can use the correct detection to update the model. This is equivalent to feeding the model manually with more images.

Diversity analysis. In Fig. 1.10, we show the average image of each category collected by OPTIMOL comparing with those of *Caltech101*. We also illustrate images collected by OPTIMOL from the Caltech 101-web and Web-23 datasets online⁴. We observe that images collected by OPTIMOL exhibit a much larger degree of diversity than those in *Caltech101*.

Furthermore, we use “accordion” and “euphonium” categories as examples to

⁴<http://vision.stanford.edu/projects/OPTIMOL/main/main.html#Dataset>



Figure 1.10: Average image of each category, in comparison to the average images of *Caltech101*. The grayer the average image is, the more diverse the dataset is.

demonstrate the learned internal structure of our dataset in Fig. 1.11. Fig. 1.11 demonstrates how our model clusters the images. The average image at the top of each tree is very gray indicating that our collected dataset is highly diverse. The middle layer shows the average images of different clusters in this dataset. Attached to these average images are the example images within each cluster. Each of the clusters exhibits unique pattern whereas the root of each tree demonstrates a combination of these patterns. Here only the three clusters with most images are shown. Theoretically, the system can have unbounded number of clusters given the property of HDP.

Hyper-parameter analysis. The concentration parameter γ controls the number

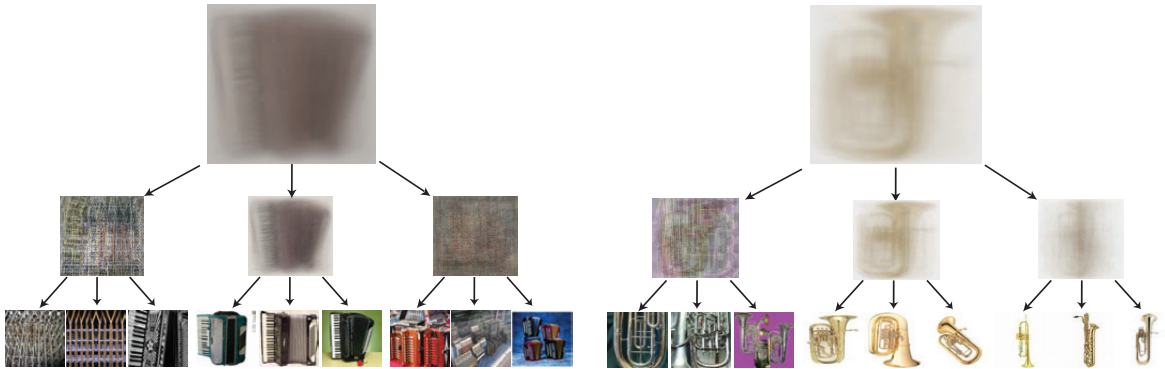


Figure 1.11: **Diversity of our collected dataset.** **Left:** Illustration of the diverse of clusters in the “accordion” dataset. Root image is the average image of all the images in “accordion” dataset collected by OPTIMOL. Middle layers are average images of the top 3 “accordion” clusters generated from the learned model. Leaf nodes of the tree structure are 3 example images attached to each of the cluster average image. **Right:** Illustration of the diverse clusters in the “euphonium” dataset.

of global clusters shared within each class. α influences the number of local clusters in each image. We demonstrate the influence of these hyper parameters on the number of global and local clusters in the “accordion” class in Fig. 1.12. In Fig. 1.12 (Left), we show that as the value of γ increases, the number of global clusters estimated by the model increases too. The average number of local clusters in each image increases when α increases (Fig. 1.12 (Right)).

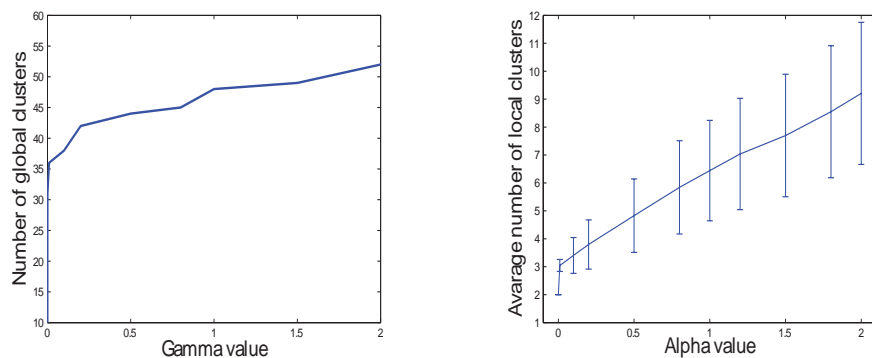


Figure 1.12: **Left:** Number of global clusters shared within each category as a function of the value of γ . **Right:** Average number of clusters in each image as a function of the value of α . The standard deviation of the average numbers are plotted as vertical bars centered at the data points.

Concept drift analysis. We have discussed in Sec. 1.3.4 that “concept drift” is the

phenomenon of object model drifting from one category to another. This will result in degraded recognition accuracy. To demonstrate the issue of “concept drift”, we compare the dataset collection performance of OPTIMOL by using different likelihood ratio threshold values. We present results of the “accordion” and “euphonium” datasets where likelihood ratio thresholds are set at 0, 30, 60, 90 percentile respectively in Fig. 1.13. Our experiment shows that a tight likelihood threshold allows fewer images to be classified as the foreground images with less false positives but more misses. A low likelihood threshold can help OPTIMOL to collect more images. But it introduces relatively more false positives hence leads to concept drift.

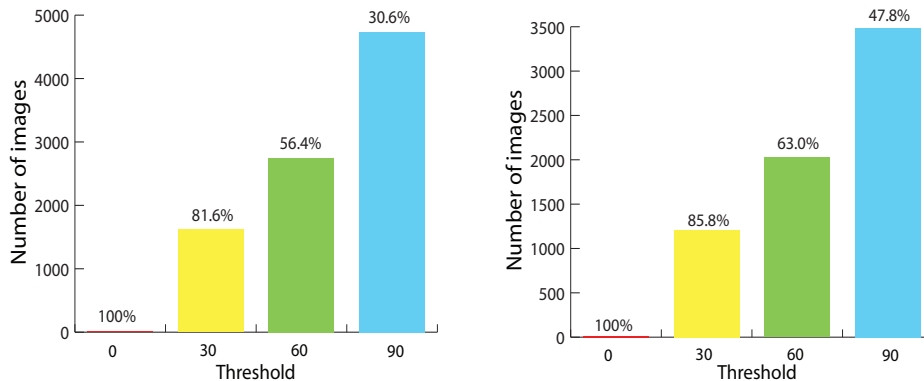


Figure 1.13: **Left:** Data collection results of OPTIMOL with different likelihood ratio threshold values for the “accordion” dataset. X-axis denotes the likelihood ratio threshold values. Y-axis represents the number of collected images. The number on the top of each bar represents the detection rate for OPTIMOL with that entropy threshold value. **Right:** Data collection results of OPTIMOL with different likelihood ratio threshold values for the “euphonium” dataset.

Illustration of images in the permanent dataset, the “cache set” and the “junk set”. We have discussed in Sec. 1.3.3 that learning with all the images accepted in classification lead to over-specialized dataset. To avoid this problem, we introduce the “cache set” to incrementally update the model. In this experiment, we compare the appearances of the images incorporated into the permanent dataset, the “cache set” as well as the discarded ones (in “junk set”). We show example images from the “accordion” class in Fig. 1.14(Left). We observe that those images to be appended to the dataset are very similar to the training images. Images kept in the “cache set” are more diverse ones among the training images whereas images

being discarded are unrelated to the image class. In Fig. 1.14(Right), we show more example images with highest and lowest entropy values from “accordion” and “inline-skate” classes.

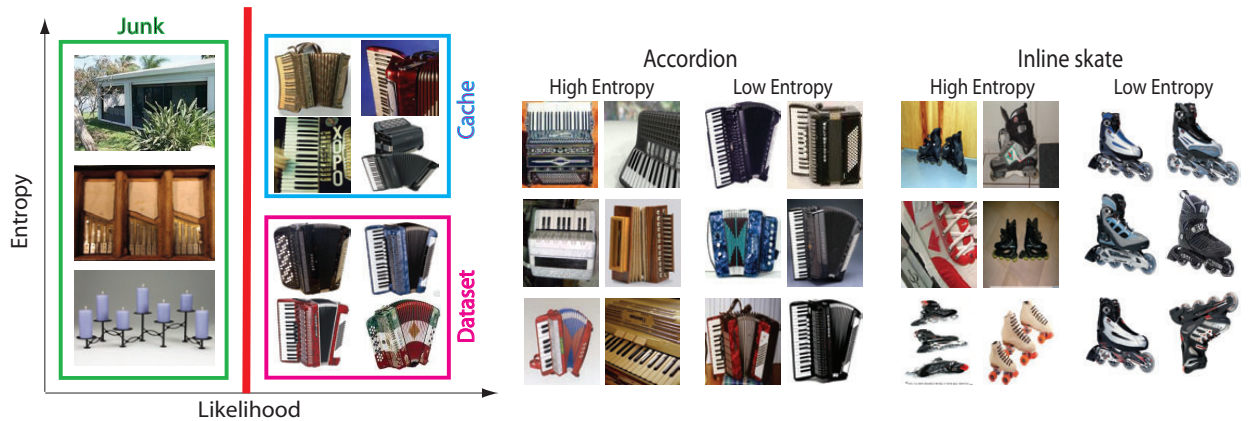


Figure 1.14: **Left:** Illustration of images in the permanent dataset, the “cache set” and the “junk set”. X-axis represents the likelihood while y-axis represents the entropy. If the likelihood ratio of an image is higher than some threshold, it is selected as a related image. This image will be further measured by its entropy. If the image has low entropy, it will be appended to the permanent dataset. If it has high entropy, it will stay in the “cache set” to be further used to train the model. **Right:** Examples of high and low entropy images in “accordion” and “inline-skate” classes.

Illustration of images collected using different entropy thresholds. In addition, we show example images from the “accordion” dataset collected by using two different entropy threshold values. Specifically, the two entropy threshold values selected are: top 30% of the related images with high entropy and all related images. Our experiment shows that a low entropy threshold allows a large proportion of the related images to be used to learn the model. Most of them have similar appearance compared to the seed images. Learning from these images makes the model susceptible to over-specialized. In other words, the updated model tends to collect even more similar images in the next iteration. Fig. 1.15(Left) shows that images collected by OPTIMOL with a low threshold are highly similar to each other. On the other hand, a high threshold provides more diverse images for the model learning, which leads to a more robust model capable of collecting more diverse images. We show these diverse images in Fig. 1.15(Right).

Detection performance comparison. In this experiment, we compare detection

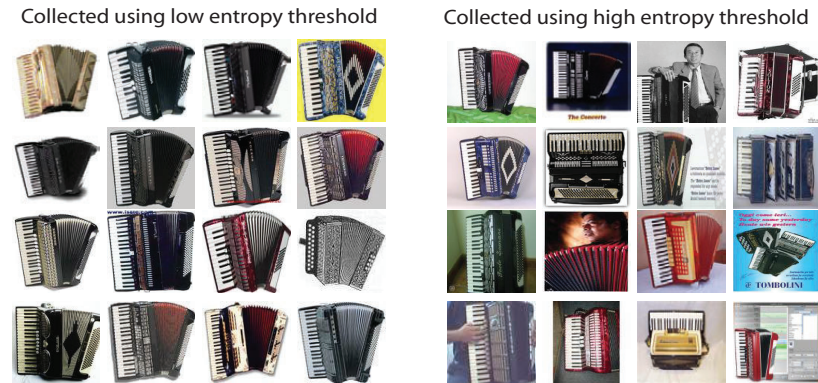


Figure 1.15: Sampled images from dataset collected by using different entropy threshold values. **Left:** Example images from dataset collected with entropy threshold set at top 100% (all images). **Right:** Example images from dataset collected with entropy threshold set at top 30%.

performance of OPTIMOL with different numbers of seed images. Using “accordion” dataset as an example, we show in Fig. 1.16 detection performance as a function of number of seed images. We use the detection rate as the criterion to measure the performance of detection. A higher detection rate indicates better performance. Our experiment shows that when the number of seed images is small, the detection rate increases significantly along with the number of seed images. When adequate initial training images are provided to train a good classifier, OPTIMOL acts robustly in selecting good examples to train itself automatically. From then on, adding seed images makes little difference in the self-training process.

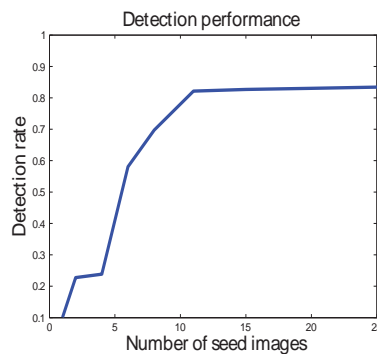


Figure 1.16: **Detection performance.** x-axis is the number of seed images. y-axis represents the detection rate.

Polysemous class analysis. We have discussed in the introduction that one challenge in collecting images from text queries is the issue of polysemy. A “mouse” could mean a computer mouse, or an animal mouse. In this experiment, we demonstrate that OPTIMOL is capable of discovering different clusters of images that reflect the polysemous nature of the query word(s). Fig. 1.17 illustrates the result. As Fig. 1.17 shows, an image search result of “mouse” gives us images of both the computer mouse and the animal mouse, in addition to other noisy images not necessarily related to either. Using a small number of seed images, OPTIMOL learns a model that captures the polysemous nature of the query. In Fig. 1.17, we show examples of images belonging to two main topic clusters estimated by OPTIMOL. It is clear that one topic contains images of the animal mouse, whereas the other contains images of the computer mouse. OPTIMOL achieves this discovery of multiple semantic clusters (i.e. polysemy) due to its ability to automatically assign meaningful topics to different images.

1.5.3 Exp.2: Image Collection

21 object categories are selected randomly from Caltech 101-Web for this experiment. The experiment is split into three parts: 1. Retrieval from Caltech 101-Web. The number of collected images in each category is compared with the manually collected images in Caltech 101-Human. 2. Retrieval from Web-23 using the same 21 categories as in part 1. 3. Retrieval from Princeton-23 using the same 21 categories as in part 1. Results of these three experiments are displayed in Fig. 1.20. We first observe that OPTIMOL is capable of automatically collecting very similar number of images from Caltech 101-Web as the humans have done by hand in Caltech 101-Human. Furthermore, by using images from Web-23, OPTIMOL collects on average 6 times as many images as Caltech 101-Human (some even 10 times higher). Princeton-23 provides a further jump on the number of collected images to approximately 20 times as that of Caltech 101-Human. In Fig. 1.20, we also compare our results with LabelMe [119] for each of the 22 categories. A “penguin” category is also included so that we can compare our results with the state-of-art dataset collecting approach [11].

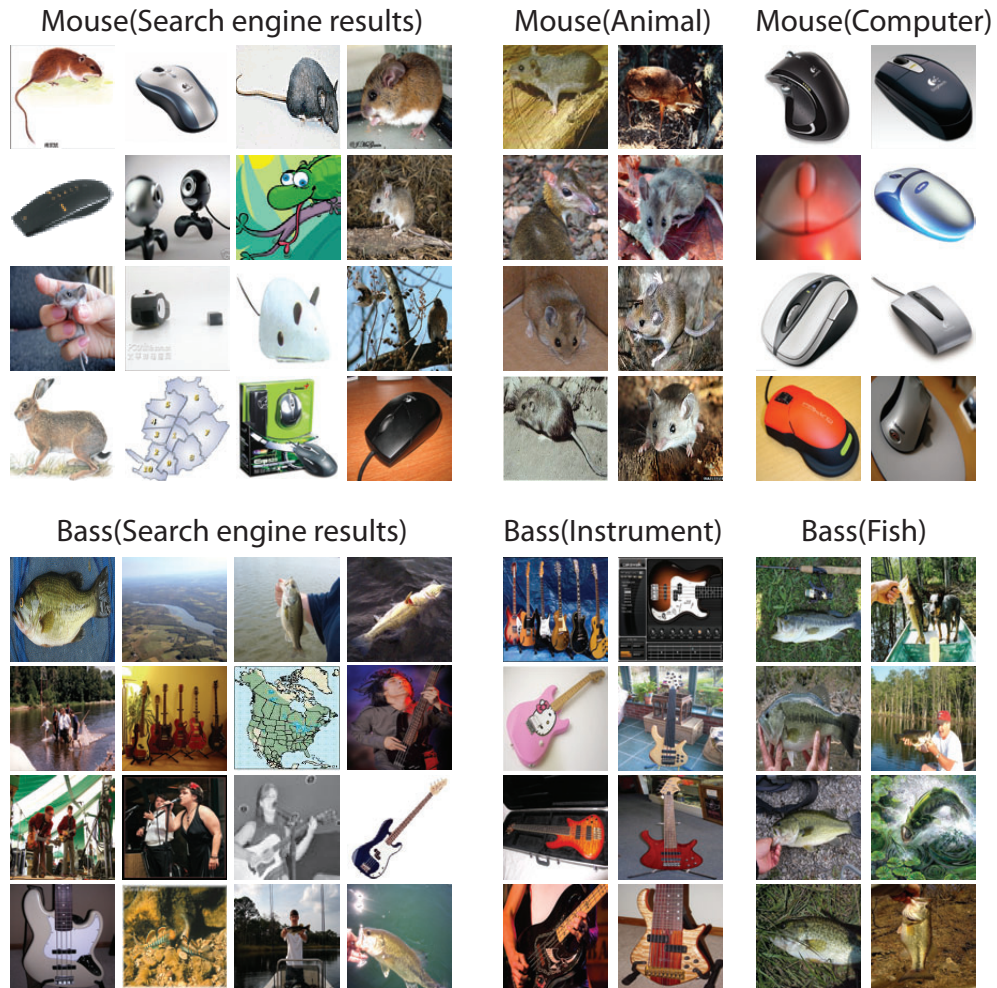


Figure 1.17: **Polysemy discovery using OPTIMOL.** Two polysemous query words are used as examples: “mouse” and “bass”. **Left:** Example images of “mouse” and “bass” from image search engines. Notice that in the “mouse” group, the images of animal mouse and computer mouse are intermixed with each other, as well as with other noisy images. The same is true for the “bass” group. For this experiment, 50 seed images are used for each class. **Right:** For each query word, example images of the two main topic clusters discovered by OPTIMOL are demonstrated. We observe that for the “mouse” query, one cluster mainly contains images of the animal mouse, whereas the other cluster contains images of the computer mouse.

In all cases, OPTIMOL collected more related images than the Caltech 101-Human, the LabelMe dataset and the approach in [11]. In addition, we conduct an additional experiment to demonstrate that OPTIMOL performs better than the base model by comparing their performance of dataset collection. The result is shown in Fig. 1.20,

where the number of images collected by the base model is represented by the gray bar. The likelihood ratio threshold is set at the same value for the full OPTIMOL model and the base model. The comparison indicates that the full OPTIMOL model collects significantly more images than the base model. This is attributed to the effectiveness of the iterative classification and model learning in OPTIMOL. Note that all of these results are achieved without any human intervention⁵, thus suggesting the viability of OPTIMOL as an alternative to costly human dataset collection. In Fig. 1.18, we demonstrate sample images from the OPTIMOL-collected datasets of “accordion” and “inline-skate” categories in the Princeton-23 data. We highlight the false positives among the images. These mistakes are most likely due to the similar appearance of the false positive images to those of the foreground images.



Figure 1.18: **Left:** Randomly selected images from the image collection result for “accordion” category. False positive images are highlighted by using red boxes. **Right:** Randomly selected images from the image collection result for “inline-skate” category.

1.5.4 Exp.3: Classification

To demonstrate that OPTIMOL not only collects large datasets of images, but also learns good models for object classification, we conduct experiment on Fergus IC-CV’05 dataset. In this experiment, we use the same experiment settings as in Fergus et al. [51] to test the multi-class classification ability of OPTIMOL. 7 object category

⁵We use 15 images from Caltech 101-Human as seed for the image collection experiment of Caltech 101-Raw since we do not have the order of the downloaded images for Caltech 101-Raw. The detection rates of Caltech 101-Raw and Web-23 in Fig. 1.20 are comparable indicating the equivalent effects of automatic and manual selection of seed set on image dataset collecting task.

models are learnt from the same training sets used by [51]. We use the same validation set in [51] to train a 7-way SVM classifier to perform object classification. The input of the SVM classifier is a vector of 7 entries, each denoting the image likelihood given each of the 7 class models. The results are shown in Fig. 1.19, where we achieve an average performance of 74.8%. This result is comparable to the 72.0% achieved by [51]. Our results show that OPTIMOL is capable of learning reliable object models.

	α	β	γ	δ	ε	ζ	ξ
airplane	76.0	14.0	0.3	5.3	0.3	0.3	4.8
car	1.0	94.5	0.3	4.5	0.3	0.3	0.3
face	0.5	1.4	82.9	3.7	0.5	0.5	11.5
guitar	2.2	4.9	5.6	60.4	13.3	0.2	13.3
leopard	1.0	2.0	1.0	5.0	89.0	1.0	2.0
motorbike	0.3	5.5	0.3	5.5	1.0	67.3	20.5
watch	1.7	5.5	17.7	11.0	5.5	5.0	53.6

Figure 1.19: **Confusion table for Exp.3.** We use the same training and testing datasets as in [51]. The average performance of OPTIMOL is 74.82%, whereas [51] reports 72.0%.

1.6 Discussion

We have developed a new approach (OPTIMOL) for image dataset collection and model learning. The self training framework makes our model more robust and generalized whereas the incremental learning algorithm boosts the speed. Our experiments show that as a fully automated system, OPTIMOL achieves accurate diverse dataset collection result nearly as good as those of humans. In addition, it provides a useful annotation of the objects in the images. Further experiments show that the models learnt by OPTIMOL are competitive with the current state-of-the-art model learned from internet images for object classification. Human labor is one of the most costly and valuable resources in research. We provide OPTIMOL as a promising alternative to collect larger diverse image datasets with high accuracy.

While the object recognition results are promising, semantic image understanding is much more comprehensive than object categorization and localization. In Chapter 2, we discuss our effort on developing visual models towards complete understanding of images.

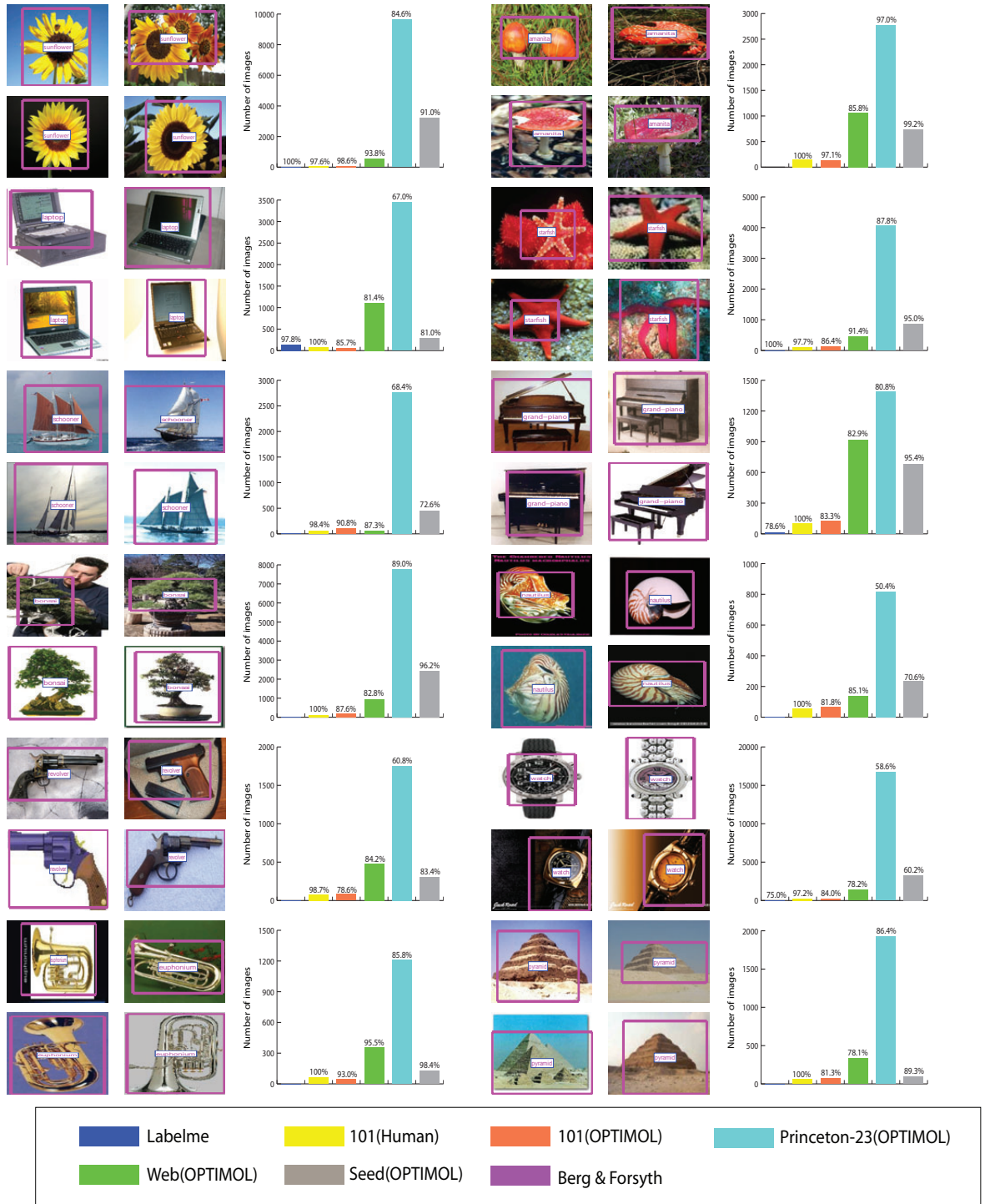


Figure 1.20: **Image collection and annotation results by OPTIMOL.** Each row in the figure contains two categories, where each category includes 4 sample annotation results and a bar plot. Let us use “Sunflower” as an example. The left sub-panel gives 4 sample annotation results (bounding box indicates the estimated locations and sizes of the “Sunflower”). The right sub-panel shows the comparison of the number of images in “Sunflower” category given different datasets. The blue bar indicates the number of “Sunflower” images in LabelMe dataset, the yellow bar the number of images in Caltech 101-Human. The OPTIMOL results are displayed using the red, green, and cyan bars, representing the numbers of images retrieved for the “Sunflower” category in Caltech 101-Web, Web-23 and Princeton-23 dataset respectively. The gray bar in each figure represents the number of images retrieved by the base model trained with only seed images. The number on top of each bar represents the detection rate for that dataset.

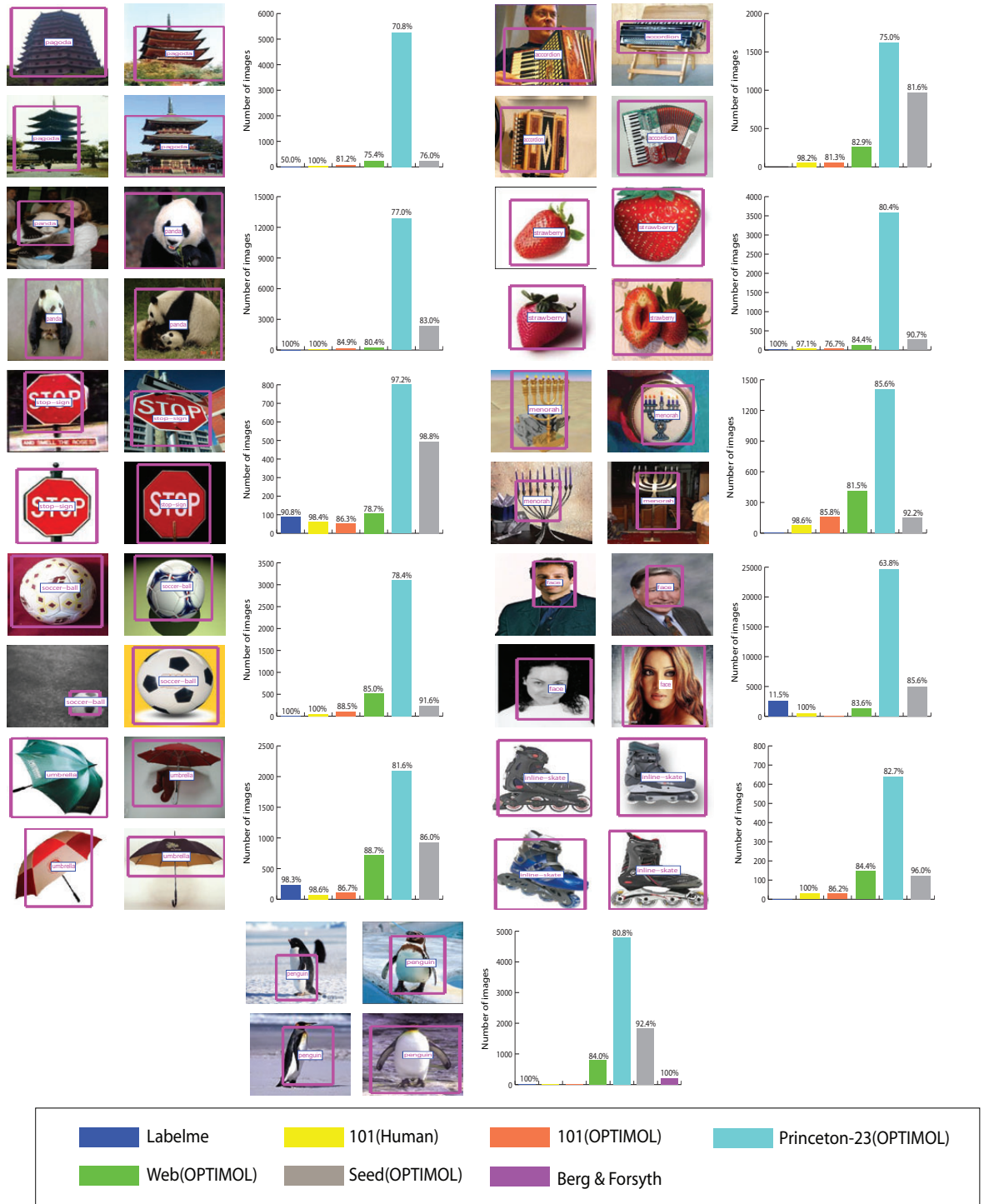


Figure 1.21: Image collection and annotation results by OPTIMOL. Notation is the same as Fig. 1.20. Since the pictures in the “face” category of Caltech 101-Human were taken by camera instead of downloading from the web, the raw Caltech images of the “face” category are not available. Hence, there is no result for “face” by 101 (OPTIMOL). All of our results have been put online at <http://vision.stanford.edu/projects/OPTIMOL.htm>

Chapter 2

Towards Total Scene Understanding

Images can provide rich information of our complex world. There is much more meaningful content beyond objects in an image, e.g. scenes, events, activities, emotions, and intentions. Psychologists have shown that the human visual system is particularly efficient and effective in perceiving high-level meanings in cluttered real-world scenes. While recognizing isolated objects and object classes is a critical component of visual recognition, a lot more is needed to be done to reach a complete understanding of visual scenes. Here, we discuss our effort on developing visual recognition algorithms towards complete understanding of visual scenes, which we call total scene understanding.

Chapter 2 is organized as follows. We first discuss the importance and motivation for total scene understanding Sec. 2.1. We then review related prior work in Sec. 2.2. In Sec. 2.3, we start with introducing a first attempt towards classifying events/complex scenes in static images by integrating scene and object categorizations. We provide a unified framework to classify an image by recognizing, annotating and segmenting the objects within the image in Sec. 2.4. Earlier versions of this work appeared in ICCV 2007 [91], a book chapter in ‘Studies in Computational Intelligence - Computer Vision’ [45] and CVPR 2009 [92].

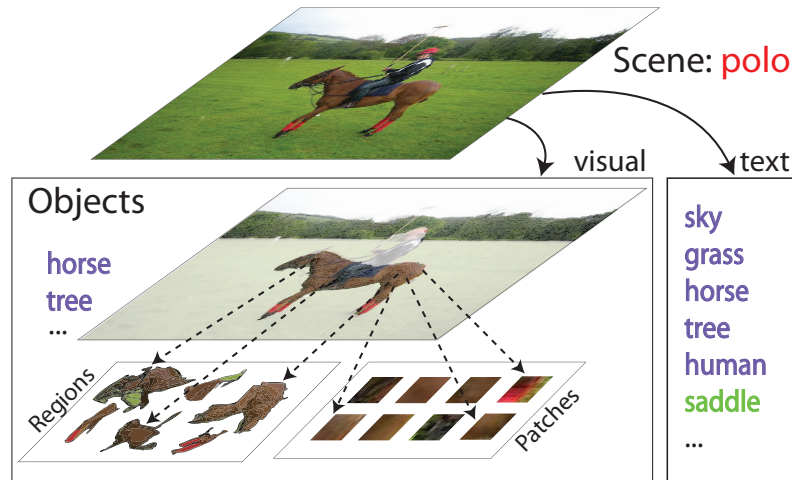


Figure 2.1: An example of what our total scene model can understand given an unknown image. At the scene level, the image is classified as a “polo” scene/event. A number of objects can be inferred and segmented by the visual information in the scene, hierarchically represented by object regions and feature patches. In addition, several tags can be inferred based on the scene class and the object correspondence.

2.1 Introduction and Motivation

When presented with a real-world image, such as the image at the top in Fig. 2.1, what do you see? For most of us, this picture contains a rich amount of semantically meaningful information. One can easily describe the image with the objects it contains (such as human, athlete, grass, trees, horse, etc.), the scene environment it depicts (such as outdoor, grassland, etc.), as well as the activity it implies (such as a polo game). Recently, a psychophysics study has shown that in a single glance of an image, humans can not only recognize or categorize many of the individual objects in the scene, tell apart the different environments of the scene, but also perceive complex activities and social interactions [44]. As humans, we can effortlessly classify this image as a rowing image (high-level scene/event classification), recognize different objects within the image (annotation), and localize and delineate where the objects are in the image (segmentation). In computer vision, a lot of progress has been made in object recognition and classification in recent years (see [43] for a review).

A number of algorithms have also provided effective models for scene environment categorization [128, 109, 135, 46]. But little has been done in holistic scene/event recognition of static images. Towards this goal, we propose principle probabilistic models to represent complex scene/event images in coherent frameworks. Three main motivations have guided our work. We highlight our contribution in achieving each of them in unified frameworks.

Total scene understanding. Most of the earlier object and scene recognition work offers a single label to an image, e.g. an image of a panda, a car or a beach. Some go further in assigning a list of annotations without localizing where in the image each annotation belongs (e.g. [89]). A few concurrent segmentation and recognition approaches have suggested more detailed decomposition of an image into foreground object and background clutter. But all of them only apply to a single object or a single type of object (e.g. [84]). A more related pioneer work is by Heitz et. al. [68], which performs segmentation on individual single type of foreground object together with a small group of stuff objects. Our proposed models captures the co-occurrences of multiple types of objects and high-level scene/event classes. Recognition becomes more accurate when different semantic components of an image are simultaneously recognized, allowing each component to provide contextual constraints to facilitate the recognition of the others. In addition, both object recognition within a scene as well as scene classification can benefit from understanding the spatial extents of each semantic concept. *Our total scene model can recognize and segment multiple objects as well as classify scenes in one coherent framework.*

Flexible and automatic learning. Learning scalability is a critical issue when considering practical applications of computer vision algorithms. For learning a single, isolated object, it is feasible to obtain labeled data. But as one wishes to understand complex scenes and their detailed object components, it becomes increasingly labor-intensive and impractical to obtain labeled data. Fortunately, the Internet offers a large amount of tagged images. *We propose a framework for automatic learning from Internet images and tags (i.e. flickr.com), hence offering a scalable approach with no additional human labor.*

Robust representation of the noisy, real-world data. While flickr images and tags provide a tremendous data resource, the caveat for exploiting such data is the large amount of noise in the user labels. The noisy nature of the labels is reflected in the highly uneven number and the quality of flickr tags: using a “polo” image as an example, many tags do not have obvious visual correspondences (e.g. “pakistan”, “adventure”); some tags can be incorrect (e.g. “snow”, “mountain”); and visually salient tags are often missing (e.g. “grass”, “human”). *Our total scene model offers, for the first time, a principled representation to account for noise related to either erroneous or missing correspondences between visual concepts and textual annotations.*

Total scene understanding is both an intriguing scientific question as well as a highly useful engineering one. From the scientific point of view, much needs to be done to understand how such complex and high level visual information can be represented in efficient yet accurate way. From an engineering point of view, total scene understanding is a useful task for numerous of applications. It is part of the ongoing effort of providing effective tools to retrieve and search semantically meaningful visual data. Such algorithms are at the core of the large scale search engines and digital library organizational tools. Total scene understanding is also particularly useful for helping us to build a safe world, as well as descriptive interpretation of the visual world for visually-impaired patients.

2.2 Related Work

A number of previous works have offered ways of recognizing scene categories [109, 135, 46]. Most of these algorithms learn global statistics of the scene categories through either frequency distributions or local patch distributions.

Object categorization is one of the most widely researched areas recently. One could grossly divide the literature into those that use generative models (e.g. [140, 52, 84]) and those that use discriminative models or methods (e.g. [47, 32, 134, 146]). Also related are algorithms about object recognition in context, either through geometric constraints [71] or through semantic relations [115]. But none of these approaches has

offered a rigorous probabilistic framework to perform detailed scene understanding tasks.

Several previous works have taken on a more holistic approach in scene interpretation [104, 72, 127, 132]. In all these works, global scene level information is incorporated in the model for improving better object recognition or detection. Mathematically, our approach is closest in spirit with Sudderth et al [127]. We both learn a generative model to label the images. Our event understanding model, however, differ fundamentally from the previous works by providing a set of integrative and hierarchical labels of an image, performing the *what*(event), *where*(scene) and *who*(object) recognition of an entire scene. In addition, our total scene understanding model extends this to more exhaustive image understanding including simultaneous object recognition, image annotation and segmentation.

2.3 Event Image Understanding by Scene and Object Recognition

Before we discuss our coherent framework for simultaneous classification, segmentation and annotation, we introduce our first attempt towards total scene understanding, an approach which classifies events/complex scenes in static images by integrating scene and object categorizations. In this work, we define an *event* to be a semantically meaningful human activity, taking place within a selected environment and containing a number of necessary objects. Take Fig. 2.2 as an example, the event is a rowing game. The scene environment is a lake, and the objects that are involved to define this event are athletes and the rowing boat. We present here a first attempt to mimic the human ability of recognizing an event and its encompassing objects and scenes. Fig. 2.2 best illustrates the goal of this work. We would like to achieve event categorization by producing as much semantic level image interpretation as possible. This is somewhat like what a school child does when learning to write a descriptive sentence of the event. It is taught that one should pay attention to the 5 W's: who, where, what, when and how. In our system, we try to answer 3 of the 5 W's: *what*



Figure 2.2: Telling the *what*, *where* and *who* story. Given an *event* (rowing) image such as the one on the left, our system can automatically interpret what is the event, where does this happen and who (or what kind of objects) are in the image. The result is represented in the figure on the right. A red name tag over the image represents the event category. The scene category label is given in the white tag below the image. A set of name tags are attached to the estimated centers of the objects to indicate their categorical labels. As an example, from the image on the right, we can tell from the name tags that this is a rowing sport event held on a lake (scene). In this event, there are rowing boat, athletes, water and trees (objects).

(the **event** label), *where* (the **scene** environment label) and *who* (a list of the **object categories**).

2.3.1 The Integrative Model

Given an image of an event, our model integrates scene and object level image interpretation in order to achieve the final event classification. Let's use the sport game polo as an example. In the foreground, a picture of the polo game usually consists of distinctive objects such as horses and players (in polo uniforms). The setting of the polo field is normally a grassland. Following this intuition, we model an event as a combination of scene and a group of representative objects. The goal of our approach is not only to classify the images into different event categories, but also to give meaningful, semantic labels to the scene and object components of the images.

To incorporate all these different levels of information, we choose a generative

model to represent our image. Fig. 2.3 illustrates the graphical model representation. We first define the variables of the model, and then show how an image of a particular event category can be generated based on this model. For each image of an event, our fundamental building blocks are densely sampled local image patches (sampling grid size is 10×10). In recent years, interest point detectors have demonstrated much success in object level recognition (e.g. [94, 36, 108]). But for a holistic scene interpretation task, we would like to assign semantic level labels to as many pixels as possible on the image. It has been observed that tasks such as scene classification benefit more from a dense uniform sampling of the image than using interest point detectors [135, 46]. Each of these local image patches then goes on to serve both the scene recognition part of the model, as well as the object recognition part. For scene recognition, we denote each patch by X in Fig. 2.3. X only encodes here appearance based information of the patch (e.g. a SIFT descriptor [94]). For the object recognition part, two types of information are obtained for each patch. We denote the appearance information by A , and the layout/geometry related information by G . A is similar to X in expression. G in theory, however, could be a very rich set of descriptions of the geometric or layout properties of the patch, such as 3D location in space, shape, and so on. For scenes subtending a reasonably large space (such as these event scenes), such geometric constraint should help recognition. In Sec. 2.3.4, we discuss the usage of three simple geometry/layout cues: verticalness, sky at infinity and the ground-plane.¹

We now go over the graphical model (Fig. 2.3) and show how we generate an event picture. Note that each node in Fig. 2.3 represents a random variable of the graphical model. An open node is a latent (or unobserved) variable whereas a darkened node is observed during training. The lighter gray nodes (event, scene and object labels) are only observed during training whereas the darker gray nodes (image patches) are

¹The theoretically minded machine learning readers might notice that the observed variables X , A and G occupy the same physical space on the image. This might cause the problem of “double counting”. We recognize this potential confound. But in practice, since our estimations are all taken placed on the same “double counted” space in both learning and testing, we do not observe a problem. One could also argue that even though these features occupy the same physical locations, they come from different “image feature space”. Therefore this problem does not apply. It is, however, a curious theoretical point to explore further.

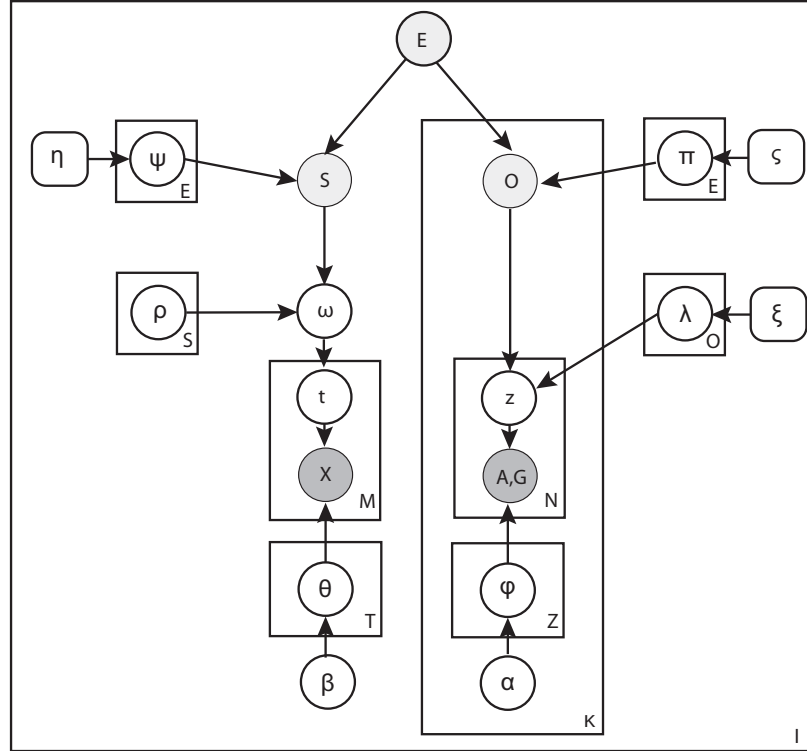


Figure 2.3: Graphical model of our approach. E , S , and O represent the event, scene and object labels respectively. X is the observed appearance patch for scene. A and G are the observed appearance and geometry/layout properties for the object patch. The rest of the nodes are parameters of the model. For details, please refer to Sec. 2.3.1

observed in both training and testing.

1. An **event** category is represented by the discrete random variable E . We assume a fixed uniform prior distribution of E , hence omitting showing the prior distribution in Fig. 2.3. We select $E \sim p(E)$. The images are indexed from 1 to I and one E is generated for each of them.
2. Given the event class, we generate the **scene** image of this event. There are in theory S classes of scenes for the whole event dataset. For each event image, we assume only one scene class can be drawn.
 - A scene category is first chosen according to $S \sim p(S|E, \psi)$. S is a discrete variable denoting the class label of the scene. ψ is the multinomial parameter

that governs the distribution of S given E . ψ is a matrix of size $E \times S$, whereas η is an S dimensional vector acting as a Dirichlet prior for ψ .

- Given S , we generate the mixing parameters ω that governs the distribution of scene patch topics $\omega \sim p(\omega|S, \rho)$. Elements of ω sum to 1 as it is the multinomial parameter of the latent topics t . ρ is the Dirichlet prior of ω , a matrix of size $S \times T$, where T is the total number of the latent topics.
 - For each of the M patches X in the scene image
 - Choose the latent topic $t \sim Mult(\omega)$. t is a discrete variable indicating which latent topic this patch will come from.
 - Choose patch $X \sim p(X|t, \theta)$, where θ is a matrix of size $T \times V_S$. V_S is the total number of vocabularies in the scene codebook for X . θ is the multinomial parameter for discrete variable X , whereas β is the Dirichlet prior for θ .
3. Similar to the scene image, we also generate an **object** image. Unlike the scene, there could be more than one objects in an image. We use K to denote the number of objects in a given image. There is a total of O classes of objects for the whole dataset. The following generative process is repeated for each of the K objects in an image.
- An object category is first chosen according to $O \sim p(O|E, \pi)$. O is a discrete variable denoting the class label of the object. A multinomial parameter π governs the distribution of O given E . π is a matrix of size $E \times O$, whereas ς is a O dimensional vector acting as a Dirichlet prior for π .
 - Given O , we are ready to generate each of the N patches A, G in the k th object of the object image
 - Choose the latent topic $z \sim Mult(\lambda|O)$. z is a discrete variable indicating which latent topic this patch will come from, whereas λ is the multinomial parameter for z , a matrix of size $O \times Z$. K is the total number of objects appear in one image, and Z is the total number of latent topics. ξ is the Dirichlet prior for λ .

- Choose patch $A, G \sim p(A, G|t, \varphi)$, where φ is a matrix of size $Z \times V_O$. V_O is the total number of vocabularies in the codebook for A, G . φ is the multinomial parameter for discrete variable A, G , whereas α is the Dirichlet prior for φ . Note that we explicitly denote the patch variable as A, G to emphasize on the fact it includes both appearance and geometry/layout property information.

Putting everything together in the graphical model, we arrive at the following joint distribution for the image patches, the event, scene, object labels and the latent topics associated with these labels.

$$p(E, S, \mathbf{O}, \mathbf{X}, \mathbf{A}, \mathbf{G}, \mathbf{t}, \mathbf{z}, \omega | \rho, \varphi, \lambda, \psi, \pi, \theta) = p(E)p(S|E, \psi)p(\omega|S, \rho) \cdot \prod_{m=1}^M p(X_m|t_m, \theta)p(t_m|w) \prod_{k=1}^K p(O_k|E, \pi) \prod_{n=1}^N p(A_n, G_n|z_n, \varphi)p(z_n|\lambda, O_k) \quad (2.1)$$

where $\mathbf{O}, \mathbf{X}, \mathbf{A}, \mathbf{G}, \mathbf{t}, \mathbf{z}$ represent the generated objects, appearance representation of patches in the scene part, appearance and geometry properties of patches in the object part, topics in the scene part, and topics in the object part respectively. Each component of Eq.2.1 can be broken into

$$p(S|E, \psi) = Mult(S|E, \psi) \quad (2.2)$$

$$p(\omega|S, \rho) = Dir(\omega|\rho_{j.}), S = j \quad (2.3)$$

$$p(t_m|\omega) = Mult(t_m|\omega) \quad (2.4)$$

$$p(X_m|t, \theta) = p(X_m|\theta_{j.}), t_m = j \quad (2.5)$$

$$p(O|E, \pi) = Mult(O|E, \pi) \quad (2.6)$$

$$p(z_n|\lambda, O) = Mult(z_n|\lambda, O) \quad (2.7)$$

$$p(A_n, G_n|z, \varphi) = p(A_n, G_n|\varphi_{j.}), z_n = j \quad (2.8)$$

where “.” in the equations represents components in the row of the corresponding matrix.

2.3.2 Labeling an Unknown Image

Given an unknown event image with unknown scene and object labels, our goal is: a) to classify it as one of the event classes (*what*); b) to recognize the scene environment class (*where*); and c) to recognize the object classes in the image (*who*). We realize this by calculating the maximum likelihood at the event level, the scene level and the object level of the graphical model (Fig. 2.3).

At the object level, the likelihood of the image given the object class is

$$p(I|O) = \prod_{n=1}^N \sum_{j=1} P(A_n, G_n | z_j, O) P(z_j | O) \quad (2.9)$$

The most possible objects appear in the image are based on the maximum likelihood of the image given the object classes, which is $O = \operatorname{argmax}_O p(I|O)$. Each object is labeled by showing the most possible patches given the object, represented as $O = \operatorname{argmax}_O p(A, G|O)$.

At the scene level, the likelihood of the image given the scene class is:

$$p(I|S, \rho, \theta) = \int p(\omega | \rho, S) \left(\prod_{m=1}^M \sum_{t_m} p(t_m | \omega) \cdot p(X_m | t_m, \theta) \right) d\omega \quad (2.10)$$

Similarly, the decision of the scene class label can be made based on the maximum likelihood estimation of the image given the scene classes, which is $S = \operatorname{argmax}_S p(I|S, \rho, \theta)$. However, due to the coupling of θ and ω , the maximum likelihood estimation is not tractable computationally [17]. Here, we use the variational method based on Variational Message Passing [142] provided in [46] for an approximation.

Finally, the image likelihood for a given event class is estimated based on the object and scene level likelihoods:

$$p(I|E) \propto \sum_j P(I|O_j) P(O_j|E) \sum_i P(I|S_i) P(S_i|E) \quad (2.11)$$

The most likely event label is then given according to $E = \operatorname{argmax}_{Ep}(I|E)$.



Figure 2.4: Our dataset contains 8 sports event classes: rowing (250 images), badminton (200 images), polo (182 images), bocce (137 images), snowboarding (190 images), croquet (236 images), sailing (190 images), and rock climbing (194 images). Our examples here demonstrate the complexity and diversity of this highly challenging dataset.

2.3.3 Learning the Model

The goal of learning is to update the parameters $\{\psi, \rho, \pi, \lambda, \theta, \beta\}$ in the hierarchical model (Fig. 2.3). Given the event E , the scene and object images are assumed independent of each other. We can therefore learn the scene-related and object-related parameters separately.

We use Variational Message Passing method to update parameters $\{\psi, \rho, \theta\}$. Detailed explanation and update equations can be found in [46]. For the object branch of the model, we learn the parameters $\{\pi, \lambda, \beta\}$ via Gibbs sampling [83] of the latent topics. In such a way, the topic sampling and model learning are conducted iteratively. In each round of the Gibbs sampling procedure, the object topic will be sampled based on $p(z_i | \mathbf{z}_{\setminus i}, A, G, O)$, where $\mathbf{z}_{\setminus i}$ denotes all topic assignment except the current one. Given the Dirichlet hyperparameters ξ and α , the distribution of topic given object $p(z|O)$ and the distribution of appearance and geometry words given topic $p(A, G|z)$ can be derived by using the standard Dirichlet integral formulas:

$$p(z = i | \mathbf{z}_{\setminus i}, O = j) = \frac{c_{ij} + \xi}{\sum_i c_{ij} + \xi \times H} \quad (2.12)$$

$$p((A, G) = k | \mathbf{z}_{\setminus i}, z = i) = \frac{n_{ki} + \varphi}{\sum_k n_{ki} + \varphi \times V_O} \quad (2.13)$$

where c_{ij} the total number of patches assigned to object j and object topic i , while n_{ki} is the number of patch k assigned to object topic i . H is the number of object topics, which is set to some known, constant value. V_O is the object codebook size. And a patch is a combination of appearance (A) and geometry (G) features. By combining Eq.2.12 and 2.13, we can derive the posterior of topic assignment as

$$p(z_i | \mathbf{z}_{\setminus i}, A, G, O) = p(z = i | \mathbf{z}_{\setminus i}, O) p((A, G) = k | \mathbf{z}_{\setminus i}, z = i) \quad (2.14)$$

Current topic will be sampled from this distribution.

2.3.4 System Implementation

Our goal is to extract as much information as possible out of the event images, most of which are cluttered, filled with objects of variable sizes and multiple categories. At the feature level, we use a grid sampling technique similar to [46]. In our experiments, the grid size is 10×10 . A patch of size 12×12 is extracted from each of the grid centers. A 128-dim SIFT vector is used to represent each patch [94]. The poses of the objects from the same object class change significantly in these events. Thus, we use rotation invariant SIFT vector to better capture the visual similarity within each object class. A codebook is necessary in order to represent an image as a sequence of appearance words. We build a codebook of 300 visual words by applying K-means for the 200000 SIFT vectors extracted from 30 randomly chosen training images per event class. To represent the geometry/layout information, each pixel in an image is given a geometry label using the codes provided by [72]. In our framework, only three simple geometry/layout properties are used. They are: ground plane, vertical structure and sky at infinity. Each patch is assign a geometry membership by the

major vote of the pixels within.

2.3.5 Dataset and Experimental Setup

Dataset

As the first attempt to tackle the problem of static event recognition, we have no existing dataset to use and compare with. Instead we have compiled a new dataset containing 8 sports event categories collected from the Internet: bocce, croquet, polo, rowing, snowboarding, badminton, sailing, and rock climbing. The number of images in each category varies from 137 (bocce) to 250 (rowing). As shown in Fig. 2.4, this event dataset is a very challenging one. Here we highlight some of the difficulties.

- The background of each image is highly cluttered and diverse;
- Object classes are diverse;
- Within the same category, sizes of instances from the same object are very different;
- The pose of the objects can be very different in each image;
- Number of instances of the same object category change diversely even within the same event category;
- Some of the foreground objects are too small to be detected.

We have also obtained a thorough groundtruth annotation for every image in the dataset. This annotation provides information for: event class, background scene class(es), most discernable object classes, and detailed segmentation of each objects.

Experimental Setup

We set out to learn to classify these 8 events as well as labeling the semantic contents (scene and objects) of these images. For each event class, 70 randomly selected images are used for training and 60 are used for testing. We do not have any previous work to compare to. But we test our algorithm and the effectiveness of each components

of the model. Specifically, we compare the performance of our full integrative model with the following baselines.

- A *scene only* model. We use the LDA model of [46] to do event classification based on scene categorization only. We “turn off” the influence of the object part by setting the likelihood of O in Eq.2.11 to a uniform distribution. This is effectively a standard “bag of words” model for event classification.
- An *object only* model. In this model we learn and recognize an event class based on the distribution of foreground objects estimated in Eq.2.9. No geometry/layout information is included. We “turn off” the influence of the scene part by setting the likelihood of S in Eq.2.11 to a uniform distribution.
- A *object + geometry* model. Similar to the object-only model, here we include the feature representations of both appearance (A) and geometry/layout (G).

Except for the LDA model, training is supervised by having the object identities labeled. We use exactly the same training and testing images in all of these different model conditions.

2.3.6 Results

We report an overall 8-class event discrimination of 73.4% by using the full integrative model. Left panel of Fig. 2.5 shows the confusion table results of this experiment. In the confusion table, the rows represent the models for each event category while the columns represent the ground truth categories of events. It is interesting to observe that the system tends to confuse bocce and croquet, where the images tend to share similar foreground objects. On the other hand, polo is also more easily confused with bocce and croquet because all of these events often take places in grassland type of environments. These two facts again with our intuition that an event image could be represented as a combination of the foreground objects and the scene environment.

In the control experiment with different model conditions, our integrative model consistently outperforms the other three models (see Fig. 2.5 Right). A curious observation is that the *object + geometry* model performs worse than the *object only* model.

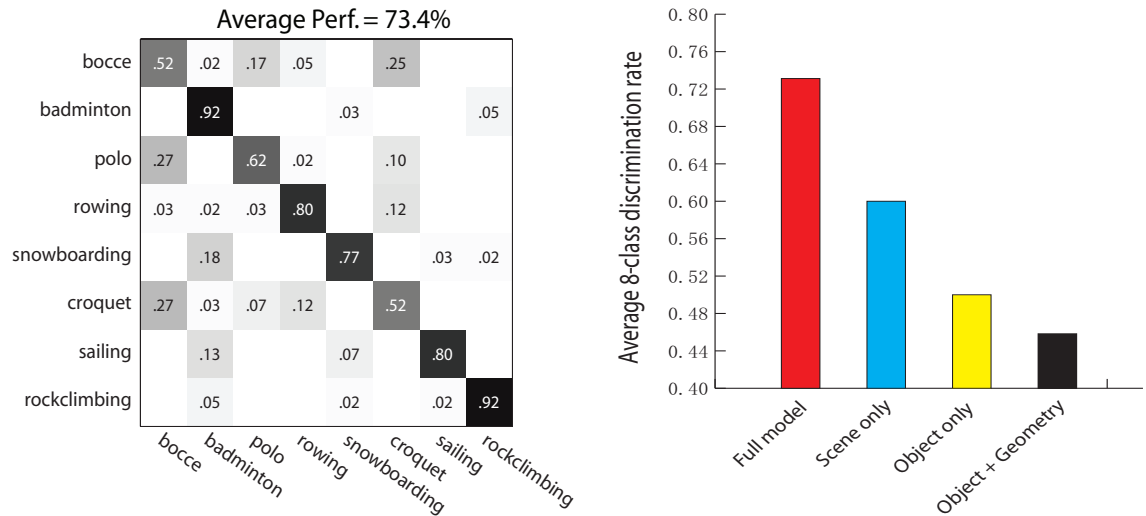


Figure 2.5: **Left:** Confusion table for the 8-class event recognition experiment. The average performance is 73.4%. Random chance would be 12.5%. **Right:** Performance comparison between the full model and the three control models. The x-axis denotes the name of the model used in each experiment, where “full model” indicates the proposed integrative model (see Fig. 2.3). The y-axis represents the average 8-class discrimination rate, which is the average score of the diagonal entries of the confusion table of each model.

We believe that this is largely due to the simplicity of the geometry/layout properties. While these properties help to differentiate sky, ground from vertical structures, they also introduce noise. As an example, water and snow are always incorrectly classified as sky or ground by the geometry labeling process, which deteriorates the result of object classification. However, the scene recognition alleviates the confusion among water, snow, sky and ground by encoding explicitly their different appearance properties. Thus, when the scene pathway is added to the integrated model, the overall results become much better.

Finally, we present more details of our image interpretation results in Fig. 2.6. We set out to build an algorithm that can tell a *what*, *where* and *who* story of the sport event pictures. We show here how each of these W’s is answered by our algorithm. Note all the labels provided in this figure are automatically generated by the algorithm, no human annotations are involved.

2.3.7 Discussion

In this work, we propose an integrative model that learns to classify static images into complicated social events such as sport games. This is achieved by interpreting the semantic components of the image as detailed as possible. Namely, the event classification is a result of scene environment classification and object categorization. Our goal is to offer a rich description of the images. It is not hard to imagine such algorithm would have many applications, especially in semantic understanding of images. Commercial search engines, large digital image libraries, personal albums and other domains can all benefit from more human-like labeling of images. Our model is, of course, just the first attempt for such an ambitious goal. Much needs to be improved. In the next section, we introduce a further step towards complete understanding of images, resulting in a hierarchical generative model that unified framework to classify an image by recognizing, annotating and segmenting the objects within the image.



Figure 2.6: (This figure is best viewed in color and with PDF magnification.) Image interpretation via event, scene, and object recognition. Each row shows results of an event class. **Column 1** shows the event class label. **Column 2** shows the object classes recognized by the system. Masks with different colors indicate different object classes. The name of each object class appears at the estimated centroid of the object. **Column 3** is the scene class label assigned to this image by our system. Finally **Column 4** shows the sorted object distribution given the event. Names on the x-axis represents the object class, the order of which varies across the categories. y-axis represents the distribution.

2.4 A Probabilistic Model Towards Total Scene Understanding: Simultaneous Classification, Annotation and Segmentation

In this section, we propose an algorithm that classifies the overall scene, recognizes and segments each object component, as well as annotates the image with a list of tags. The result of our algorithm is a generative model that encodes a hierarchy of semantic information contained in the scene (Fig. 2.1). To our knowledge, this is the first model that performs all three tasks in one coherent framework. For instance, a scene of a polo game consists of several visual objects such as human, horse, grass, etc. In addition, it can be further annotated with a list of more abstract (e.g. dusk) or visually less salient (e.g. saddle) tags. Our generative model jointly explains images through a visual model and a textual model.

2.4.1 The Hierarchical Generative Model

We propose a hierarchical generative model which aims to interpret scene images, their objects and the associated noisy tags. The model shown in Fig. 2.7 describes the scene of an image through two major components. In the visual component, a scene consists of objects that are in turn characterized by a collection of patches and several region features. The second component deals with noisy tags of the image by introducing a binary switch variable. This variable enables the model to decide whether a tag is visually represented by objects in the scene or whether it represents more visually irrelevant information of the scene. Therefore, the switch variable enables a principled joint modeling of images and text and a coherent prediction of what tags are visually relevant. The hierarchical representation of image features, object regions, visually relevant and irrelevant tags, and overall scene provides both top-down and bottom-up contextual information to components of the model.

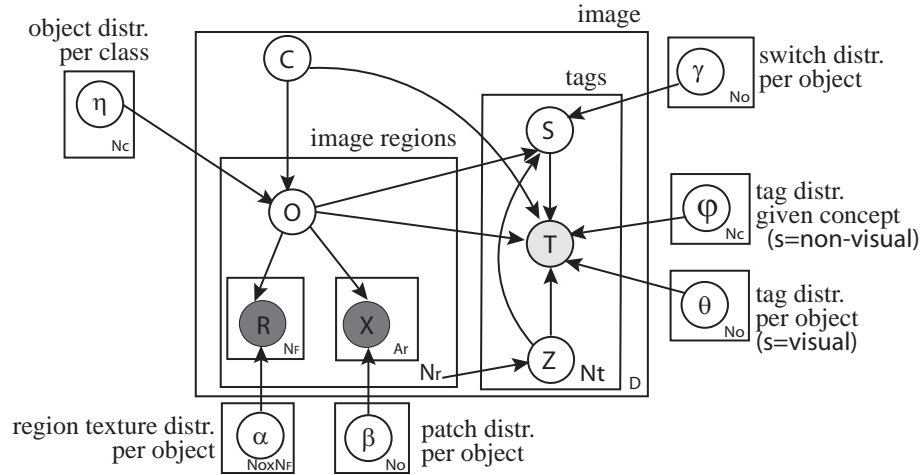


Figure 2.7: A graphical model representation of our generative model. Nodes represent random variables and edges indicate dependencies. The variable at the right lower corner of each box denotes the number of replications. The box indexed by N_r represents the visual information of the image, whereas the one indexed by N_t represents the textual information (i.e. tags). $N_c, N_o, N_x, N_{f_i}, i \in 1, 2, 3, 4$ denote the numbers of different scenes, objects, patches and regions for region feature type i respectively. Hyperparameters of the distributions are omitted for clarity.

The Generative Process

In order to explain the generative process of our model mathematically, we first introduce the observable variables. Each image $d \in D$ is over-segmented into small coherent regions by using Felzenszwalb et al [49]. For each region, we extract $N_F = 4$ types of features, where $F = \{\text{shape, color, location, texture}\}^2$. We further vector quantize region features into region codewords, denoted by the variable R in the model (see example of the representative regions for “horse” in Fig. 2.1). We use 100, 30, 50, 120 codewords for each feature type, respectively. Additionally, the set of patches X is obtained by dividing the image into blocks. Similarly, patches are represented as 500 codewords obtained by vector quantizing the SIFT [94] features extracted from them (see example of the representative patches for “horse” in Fig.

²We use the shape and location features described in [96]. Color features are simple histograms. Texture features are the average responses of filterbanks in each region.

2.1). Noisy tags are represented by the variable T , which is observed in training. To generate an image and its corresponding annotations, a scene class C is sampled from a fixed uniform prior distribution. Given a scene, we are now ready to generate both the visual and textual components of the scene.

Generating the visual component. Given the scene class C , the probability of objects in such scenes is governed by a multinomial distribution. For each of the N_r image regions denoted by the left internal box in Fig. 2.7, we first sample an object $O \sim \text{Mult}(\eta_c)$. Given the object O , we sample the image appearance:

1. For each $i \in F$, sample global appearance features: $R_i \sim \text{Mult}(\alpha_i|O)$, where there is a unique α_i for each object and each type of region feature.
2. Sample A_r many patches: $X \sim \text{Mult}(\beta|O)$.

Generating the tag component. At the same time, a region index Z is sampled from a uniform distribution. Z is used to account for the different numbers of tags and regions in this image, as suggested by [16]. As mentioned above, the switch variable S allows tags T to correspond to either visually relevant (i.e. the objects) or visually irrelevant (i.e. more abstract information) parts of the scene. This is formulated by allowing tags T to be drawn from either the distribution governed by object O , or the one controlled by scene class C . These ideas are summarized in the following generative procedure. For each of the N_t image tags:

1. Sample an index variable: $Z \sim \text{Unif}(N_r)$. Z is responsible for connecting an image region with a tag.
2. Sample the switch variable $S \sim \text{Binomial}(\gamma_{O_Z})$. S decides whether this tag is generated from the visually relevant object O or more visually irrelevant information related to the scene C . Fig. 2.8 shows examples of switch probabilities for different objects.
 - (a) If $S = \text{non-visual}$: sample a tag $T \sim \text{Mult}(\varphi_c)$.
 - (b) If $S = \text{visual}$: sample a tag $T \sim \text{Mult}(\theta_{O_Z})$.

Putting the generative process together, the resulting joint distribution of scene class C , objects O , regions R , image patches X , annotation tags T , as well as all the latent variables becomes:

$$\begin{aligned}
p(C, \mathbf{O}, \mathbf{R}, \mathbf{X}, \mathbf{S}, \mathbf{T}, \mathbf{Z} | \eta, \alpha, \beta, \gamma, \theta, \varphi) &= p(C) \cdot \left(\prod_{n=1}^{N_r} p(O_n | \eta, C) \right) \\
&\times \prod_{n=1}^{N_r} \left(\left(\prod_{i=1}^{N_F} p(R_{ni} | O_n, \alpha_i) \right) \cdot \prod_{r=1}^{A_r} p(X_{nr} | O_n, \beta) \right) \\
&\times \prod_{m=1}^{N_i} p(Z_m | N_r) p(S_m | O_{Z_m}, \gamma) p(T_m | O_{Z_m}, S_m, \theta, C, \varphi)
\end{aligned} \tag{2.15}$$

2.4.2 Properties of the Model

Our model is designed to perform three visual recognition tasks in one coherent framework: classification, annotation and segmentation. Eq.2.15 shows the joint probability of variables governing these three tasks. Later, in Eq.2.23, it will be clear how scene classification can directly influence the annotation and segmentation tasks.

Through the coupling of a scene C , its objects and their regions the model creates a hierarchical representation of an image. By modeling three layers jointly, they each improve the overall recognition accuracy. Each scene C defines a unique distribution $p(O|C)$ over objects. Additionally, the scene class C influences the distribution $p(T|C)$ over tags. This scene class influence serves as a top-down contextual facilitation of the object recognition and annotation tasks.

One unique feature of our algorithm is the concurrent segmentation, annotation and recognition, achieved by combining a textual and a visual model. Furthermore, our visual model goes beyond the “bag-of-words” model by including global region features and patches inspired by [21].

Lastly, the model presents a principled approach to dealing with noisy tags. Fig. 2.8 shows probability values of switch variable “S” for different objects. When a tag is likely to be generated from a visually irrelevant, abstract source (e.g. “wind”), its $p(S = \text{visual})$ is low; whereas tags such as “grass”, “horse” show high probability of

being visually relevant.

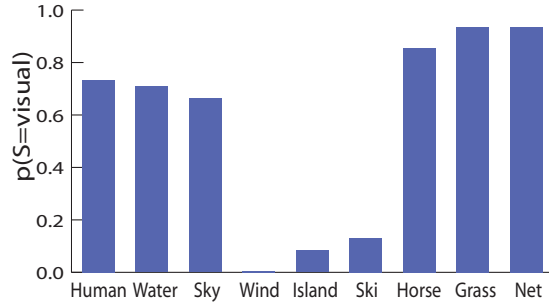


Figure 2.8: Probabilities of different objects. Words such as “horse” or “net” have higher probability because users tend to only tag them when the objects are really present, largely due to their clear visual relevance. On the contrary, words such as “island” and “wind” are usually related to the location or some other visually irrelevant concept, and usually not observable in a normal photograph.

2.4.3 Learning via Collapsed Gibbs Sampling

We have described the model in detail and can now turn to learning its parameters. To this end, we derive a collapsed Gibbs sampling algorithm [106]. For each image and its tags, we sample the following latent variables: object O , switch variable S and index variable Z . Let O_{dn} denote the object for the n th region in the d th image, \mathbf{R}_{dn} and \mathbf{X}_{dn} represent the sets of its region features and patches. We define set $A_{dn} = \{j : Z_{dj} = n\}$ and $B_{dn} = \{j : Z_{dj} = n, S_{dj} = \text{visual}\}$. The switch variables related to A_{dn} is denoted as S_A , i.e., $S_A = \{S_{dj} : j \in A_{dn}\}$. The tags related to B_{dn} are represented as $T_B = \{T_{dj} : j \in B_{dn}\}$. \bar{O}_{dn} represents all object assignments excluding O_{dn} . Similarly, we define the switch, index and tag variables S_{dm} , Z_{dm} , T_{dm} , \bar{S}_{dm} , \bar{Z}_{dm} and \bar{T}_{dm} for the m th tag in the d th image. \bar{S}_A and \bar{T}_B represent the corresponding assignments excluding A_{dn} and B_{dn} respectively. Following the Markov property of variable O , we analytically integrate out parameters $\eta, \alpha, \beta, \gamma, \varphi, \theta$. Then, the posterior over the object O_{dn} can be described as:

$$\begin{aligned}
 p(O_{dn} = o | \bar{O}_{dn}, C_d, \mathbf{R}, \mathbf{X}, \mathbf{S}, \mathbf{T}, \mathbf{Z}) &\propto p(O_{dn} = o | \bar{O}_{dn}, C_d) \cdot p(\mathbf{R}_{dn} | \bar{\mathbf{R}}_{dn}, \mathbf{O}) \cdot \\
 p(\mathbf{X}_{dn} | \bar{\mathbf{X}}_{dn}, \mathbf{O}) &\cdot p(\mathbf{Z}_A | N_r) \cdot p(S_A | \mathbf{O}, \mathbf{Z}, \bar{S}_A) \cdot p(T_B | \mathbf{O}, \mathbf{Z}, \mathbf{S}, \bar{T}_B)
 \end{aligned} \tag{2.16}$$

Using standard Dirichlet integral formulation, we obtain the first element of this product:

$$p(O_{dn} = o | \bar{O}_{dn}, C_d = c) = \frac{n_{co,-dn} + \pi_o}{\sum_{o'} n_{co',-dn} + N_o \pi_o} \quad (2.17)$$

where π_o is the symmetric Dirichlet hyperparameter governing η . N_o is the total numbers of different objects. $n_{co,-dn}$ denotes the number of assignments of the object class o to scene class c , not including the current instance.

Similarly, the other counting variables $n_{of_i,-dn}, n_{ox,-dn}, n_{os,-dm}$ and $n_{ot,-dm}$ are also defined as the number of occurrences for f_i, x, s, t with o excluding the instances related to dn or dm . Given $n_{os} = \#(Z = z, O_z = o, S = s)$, $n_{os,-A_{dn}}$ indicates the frequency of s co-occurring with o excluding instances related to set A_{dn} . Given $n_{ot} = \#(Z = z, O_z = o, S = \text{visual}, T = t)$, $n_{ot,-B_{dn}}$ is the frequency of t co-occurring with o excluding instances related to set B_{dn} . $n_{ct,-dm}$ denotes the number of times tag t co-occurring with scene type c , excluding the current instance. Furthermore, N_{f_i}, N_x, N_t are the total numbers of different region features, patches and tags. The hyperparameters $\pi_o, \pi_{f_i}, \pi_x, \pi_s, \pi_{ct}, \pi_{ot}$ are symmetric Dirichlet distributions governing $\eta, \alpha_i, \beta, \gamma, \varphi, \theta$.

S only has two possible values: $S = \text{visual}$ indicates a visually relevant object and $S = \text{non-visual}$ indicates a visually irrelevant object or scene information. Hence, $N_s = 2$.

The second and third part of Eq.2.16 become:

$$\begin{aligned} p(\mathbf{R}_{dn} | \bar{\mathbf{R}}_{dn}, \mathbf{O}) &= \prod_{i=1}^{N_F} p(R_{dni} = f_i | \bar{\mathbf{R}}_{dni}, \mathbf{O}_{dn} = o, \bar{O}_{dn}) \\ &= \prod_{i=1}^{N_F} \frac{n_{of_i,-dn} + \pi_{f_i}}{\sum_{f'_i} n_{of'_i,-dn} + N_{f_i} \pi_{f_i}} \end{aligned} \quad (2.18)$$

$$p(\mathbf{X}_{dn} | \bar{\mathbf{X}}_{dn}, \mathbf{O}) = \frac{\Gamma(\sum_{x'} n_{ox',-dn} + N_x \pi_x)}{\prod_{x'} \Gamma(n_{ox',-dn} + \pi_x)} \times \frac{\prod_{x'} \Gamma(n_{ox'} + \pi_x)}{\Gamma(\sum_{x'} n_{ox'} + N_x \pi_x)} \quad (2.19)$$

$p(Z = z | N_r) = \frac{1}{N_r}$, hence $p(\mathbf{Z}_A | N_r)$ is constant. The part related to S is:

$$p(S_A | \mathbf{O}, \bar{S}_A, \mathbf{Z}) = \frac{\Gamma(\sum_{s'} n_{os',-A_{dn}} + N_s \pi_s)}{\prod_{s'} \Gamma(n_{os',-A_{dn}} + \pi_s)} \frac{\prod_{s'} \Gamma(n_{os'} + \pi_s)}{\Gamma(\sum_{s'} n_{os'} + N_s \pi_s)}. \quad (2.20)$$

The contribution of tags to object concept O_{dn} is:

$$p(T_B | \mathbf{O}, \bar{T}_B, \mathbf{Z}, \mathbf{S}) = \frac{\Gamma(\sum_{t'} n_{ot', -B_{dn}} + N_t \pi_{ot})}{\prod_{t'} \Gamma(n_{ot', -B_{dn}} + \pi_{ot})} \frac{\prod_{t'} \Gamma(n_{ot'} + \pi_{ot})}{\Gamma(\sum_{t'} n_{ot'} + N_t \pi_{ot})}. \quad (2.21)$$

Similarly, we derive the posterior over the switch variable S and index Z .

Up to this point, the update equations only need tags and images. However, there is no information of which tag T corresponds to which object O inside the image. Without such information it is possible to confuse tag-object relations, if both only occur together, e.g. “water” and “sailboat”. To prevent such a case, we introduce an automatic initialization system which provides a few labeled regions.

2.4.4 Automatic Initialization Scheme

In this section, we propose an initialization scheme which enables us to learn the model parameters with no human effort of labeling. The goal of the initialization stage is to provide a handful of relatively clean images in which some object regions are marked with their corresponding tags. During the learning process, these regions and tags provide seed information to the update equations.

In the preprocessing step, we use a lexicon to remove all tags that do not belong to the “physical entity” group. Any lexicon dataset may be used for this purpose, we choose WordNet [103]. We also group all words in one WordNet synset (a group of synonyms) to one unique word, e.g. “sailing boat” and “catboat” are both transformed to “sailboat”.

In the next step, we query flickr.com with the object names collected from the previous step to obtain initial training sets for each of the object classes. We then train the object model described in [21] and apply it to all scene images. A few object regions are collected from a handful of images for each object class. We now have a small number of partially annotated scene images and their still noisy tags, we select the best K of such images to seed the learning process described in Sec. 2.4.3. This is done by ranking the images by favoring larger overlaps between the tags and

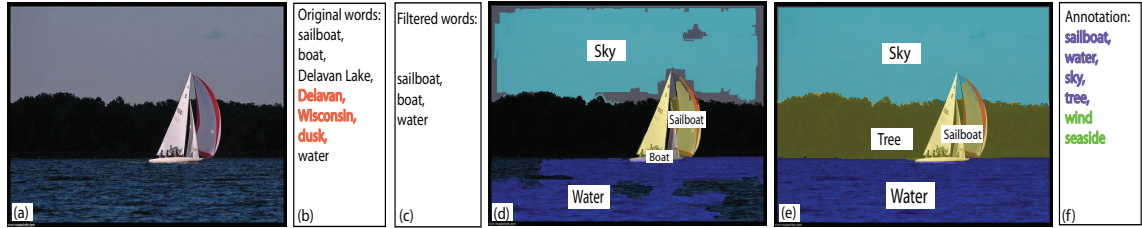


Figure 2.9: Walk-through of the learning process. **(a)**: The original image. **(b)**: The original tags from Flickr. Visually irrelevant tags are colored in red. **(c)**: Output of Step 1 of Algorithm 1: Tags after the WordNet pruning. **(d)**: Output of Step 2 of Algorithm 1: The image is partly annotated using the initialization scheme. Different object concepts are colored differently. Note that there is a background class in our initialization scheme, which is denoted in black in this figure. Since the criterion for being selected as an initial image is very conservative, the image annotations are clean but many regions are not annotated (missing tags). **(e)**: Output of Step 3 of Algorithm 1: After training the hierarchical model, the image is completely and more precisely segmented. **(f)**: Final annotation proposed by our approach. Blue tags are predicted by the visual component ($S = \text{visual}$). Green tags are generated from the top down scene information learned by the model ($S = \text{non-visual}$).

annotated objects ³.

2.4.5 Learning Summary

Algorithm 1 summarizes the learning process. Furthermore, Fig. 2.9 provides an example walk-through of one training image. After the pre-processing stage, some of the noisy tags are pruned out, but some visually relevant tags are still missing (e.g. tree, sky). The automatic initialization scheme then provides some segmented and annotated regions as seeds for the learning of the generative model. Using these seed images and additional unannotated images, the visual and textual components of the model are jointly trained. The effect of this joint learning is a robust model that can segment images more accurately (Fig. 2.9(e)). Note that some visually irrelevant tags could also be recovered at the end of training (Fig. 2.9(f), e.g. wind). This is attributed to the top-down influence of the scene class on the tags. Having learned

³ $rank(O_d, T_d, P(T_d|C)) = \frac{\sum_{T_d \in O_d \cap T_d} P(T_d|C)}{\sum_{T_d \in O_d \cup T_d} P(T_d|C)}$, where T_d are the flickr tags of image d and $P(T_d|C)$ is the observed probability of tag T_d given the scene class C

Algorithm 1 Automatic training framework

Step 1: Obtaining Candidate Tags Reduce the number of tags by keeping words that belong to the “physical entity” group in WordNet. Group synonyms using WordNet synsets.

Step 2: Initialize Object Regions

Obtain initial object models. Apply the automatic learning method of [21] to learn an initial object model.

Annotate scene images. Apply the learned object model to annotate candidate object regions in each scene image.

Select initialization images. Select a small number of initialized images by a ranking metric described by Footnote3

Step 3: Automatic Learning. Treat the automatically selected top ranked images as “supervised” data, add more flickr images and their tags to jointly train the model described in section 2.4.1.

this model, we can now turn to inference.

2.4.6 Inference: Classification, Annotation and Segmentation

Classification. The goal of classification is to estimate the most likely scene class for an image given an unknown image without any annotation tags. We use the visual component of the model (i.e. the region and patch appearances) to compute the probability of each scene class, by integrating out the latent object variable O :

$$\begin{aligned}
 p(C|\mathbf{R}_d, \mathbf{X}_d) &= \frac{p(C, \mathbf{R}_d, \mathbf{X}_d)}{p(\mathbf{R}_d, \mathbf{X}_d)} \\
 &\propto \prod_{N_r} \sum_O p(\mathbf{R}|O)p(\mathbf{X}|O)p(O|C)
 \end{aligned}
 \tag{2.22}$$

Finally, we choose $c = \operatorname{argmax}_C p(C|\mathbf{R}_d, \mathbf{X}_d)$.

Annotation. Given an unknown image, annotation tags are extracted from the segmentation results derived below.

Segmentation. Segmentation infers the exact pixel locations of each of the objects in the scene. By integrating out all the scene classes, we obtain:

$$\begin{aligned} p(O|\mathbf{R}, \mathbf{X}) &= \sum_C p(O, C|\mathbf{R}, \mathbf{X}) \propto \sum_C p(O, C, \mathbf{R}, \mathbf{X}) \\ &= \sum_C p(O|C)p(\mathbf{R}|O)p(\mathbf{X}|O)p(C) \end{aligned} \quad (2.23)$$

We observe that object segmentation is influenced both by the top-down force of scene class (first term in Eq.2.23) as well as the bottom-up force generated by the visual features (second and third terms in Eq.2.23).

2.4.7 Experiments and Results

We test our approach on 8 scene categories suggested in [91]: *badminton, bocce, croquet, polo, rock climbing, rowing, sailing, snowboarding*. By using these category names as keywords, we first automatically crawl the Flickr website to obtain 800 images and their tags for each category. 200 randomly selected images from each class are set aside as the testing images. After Step 1 of Algorithm 1, we obtain a vocabulary of 1256 unique tags. For segmentation experiments we consider the 30 most frequent words from this list. Note, however, that top down influence from the scene information still enables our model to be able to annotate images with tags from the full list of 1256 words. We offer more details about this dataset in the technical report.

Our hierarchical model can perform three tasks: image level classification, individual object annotation as well as pixel level segmentation. We now investigate performance of these tasks as well as the influence of parts of the model such as the switch variable on overall accuracy.

Scene Classification

The goal in this experiment is to classify an unknown image as one of the eight learned scene classes. We perform three experiments to analyze the different aspects of our model and learning approach. All evaluations are done based on the 8-way classification results⁴.

A. Comparison with different models. We compare the results of our model with three other approaches: (i) a baseline bag of words image classification model [46]; (ii) the region-based model used to initialize our initial object class models [21]; (iii) a modified Corr-LDA model based on [16] by adding a class variable on top of the mixing proportion parameter θ in the original model.

We provide the same list of tags generated by our system to our model and the modified model of [16]. Fig. 2.10 summarizes the results. Our model consistently outperforms the other three approaches, whereas the region-based model [21] and the modified Corr-LDA model outperform a simple bag of words model. A comparison of the modified Corr-LDA model and our model underlines the effectiveness of our selective learning of visually relevant and irrelevant tags of the real-world data.

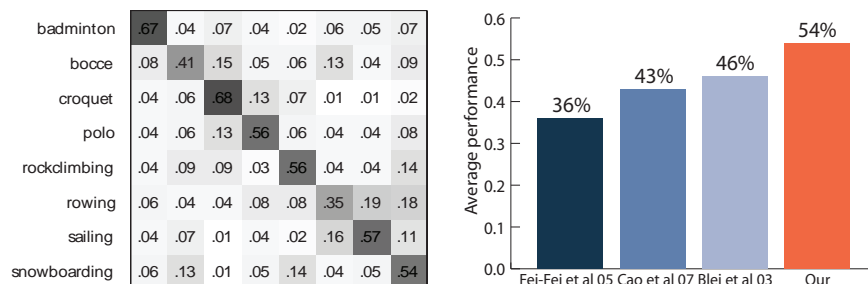


Figure 2.10: Comparison of classification results. **Left: Overall performance.** Confusion table for the 8-way scene classification. Rows represent the models for each scene while the columns represent the ground truth classes. The overall classification performance is 54%. **Right: Comparison with different models (Experiment A).** Performance of four methods. Percentage on each bar represents the average scene classification performance. 3rd bar is the modified Corr-LDA model [16].

⁴An 8-way classification result can be depicted by an 8×8 confusion table. By convention, we use the average of the diagonal entries of the table as the overall classification results of a particular model.

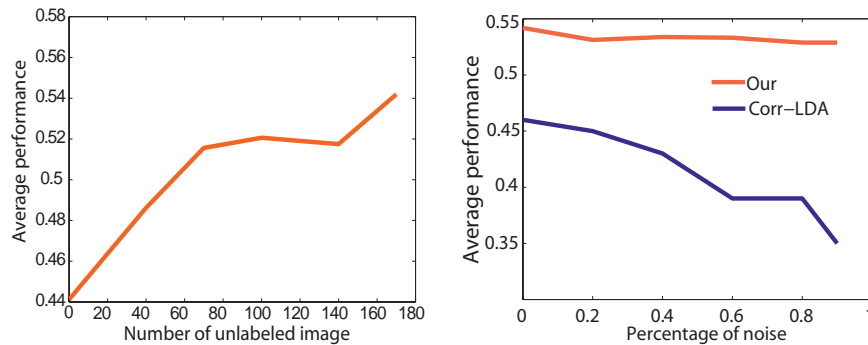


Figure 2.11: **Left: Influence of unannotated data (Experiment B).** Classification performance as a function of number of unannotated images. The y axis represents the average classification performance. The x axis represents the number of unlabeled images. It shows the unannotated images also contribute to the learning process of our model. **Right: Effect of noise in tags (Experiment C).** Performance of different models as a function of noise percentage in the tags. The y axis is average classification performance. The x axis represents the percentage of noisy tags. While the performance of corr-LDA decreases with the increase of percentage of noise, our model performs robustly by selectively learning the related tags.

B. Influence of unannotated data. To provide some insight into the learning process, we show in Fig. 2.11(Left) the classification performance curve as a function of the number of unlabeled images given to the model. In this experiment, the number of initialized images are fixed to 30. Performance gradually increases when more unlabeled images are included. This proves the effectiveness of unlabeled data in our learning framework.

C. Effect of noise in tags. In order to underline the robustness of our model to noisy training data, we present a set of experiments in which we dilute the original flickr tags with different percentages of noise by adding arbitrary words from the list of 1256 words during the training process. Fig. 2.11(Right) shows that while the algorithm of [16] decreases in accuracy when noise increases, our model is oblivious to even large percentages of noise. The robustness to noise is mostly attributed to the switch variable “S”, which correctly identifies most of the noisy tags, hence keeping the visual part of the model working properly even amidst a large amount of tagging noise.

Image annotation

Annotation tags are given through the results of segmentation. If there is a region of a certain object, we treat the name of this object as a tag.

D. Comparison to other annotation methods. In this experiment, we compare annotation results with two other state-of-the-art annotation methods – Alipr [89] and Corr-LDA [16]. We use precision-recall and F-measures to demonstrate the annotation results. Table 2.1 lists detailed annotation results for seven objects, as well as the overall scores from all object classes. Our annotation consistently and significantly outperforms the other two methods. This can be largely attributed to the selective learning of useful tags that can find a balance between bottom-up visual appearance cues and top-down scene class information.

Object	Alipr			Corr LDA			Our Model		
	Prec	Rec	F	Prec	Rec	F	Prec	Rec	F
human	.83	.97	.89	.83	1.00	.91	.85	.98	.91
horse	–	–	–	.17	.91	.28	.17	.91	.29
grass	.42	.86	.56	.22	1.00	.35	.33	.86	.48
sky	.59	.33	.43	.55	.17	.26	.44	.92	.59
tree	.45	.38	.41	.25	.01	.03	.38	.93	.54
net	–	–	–	–	–	–	.27	.85	.41
sand	–	–	–	–	–	–	.24	.46	.32
Mean	.15	.22	.16	.16	.40	.15	.28	.73	.34

Table 2.1: Comparison of precision and recall values for annotation with Alipr, corr-LDA and our model. Detailed results are given for seven objects, but means are computed for all 30 object categories (**Experiment D**).

Image segmentation

Our model not only classifies an image as a scene class, but also provides pixel level segmentation of the objects in the image without any such information given during training. We first compare quantitative results with another approach and then show a qualitative difference in example segmentations with and without the top down contextual influence provided by the scene class C .

Object	Cao & Fei-Fei, 2007			Our Model		
	Prec	Rec	F	Prec	Rec	F
human	.35	.23	.28	.43	.47	.45
horse	.13	.49	.20	.27	.53	.36
grass	.62	.38	.47	.59	.50	.54
sky	.79	.44	.56	.74	.73	.73
tree	.40	.48	.44	.41	.59	.48
net	.04	.09	.05	.45	.26	.33
sand	.11	.32	.16	.29	.35	.32
Mean	.22	.34	.28	.42	.46	.43

Table 2.2: Results of segmentation on seven object categories and mean values for all 30 categories (**Experiment E**).

E. Comparison to another segmentation method. In the image segmentation and annotation experiments, we train our model on 30 initialized images plus 170 unlabeled images. We test on 240 images where groundtruth is provided by human segmentation. Precision is computed by dividing the total area of correctly segmented pixels by the total area of detected pixels for each object. Recall is calculated by dividing the total area of correctly segmented pixels by the total area of true pixels of each object. We compare our segmentation results with the region-based model in [21]. [21] is used in the training of our initial object models. It is also one of the state-of-the-art concurrent object segmentation and recognition methods. Table 2.2 shows that our model significantly outperforms [21] in every object classes.

F. Influence of the scene class on annotation and segmentation. In this experiment, we examine the top-down, contextual influence of a scene in our model (Fig. 2.12). We compare our full model to a damaged model in which the top down influence of the scene class is ignored. Our results underscore the effectiveness of the contextual facilitation by the top-down classification on the annotation and segmentation tasks.

2.5 Discussion

Holistic understanding of complex visual scenes is an indispensable functionality of the future generations of artificial intelligence system. One of the most important

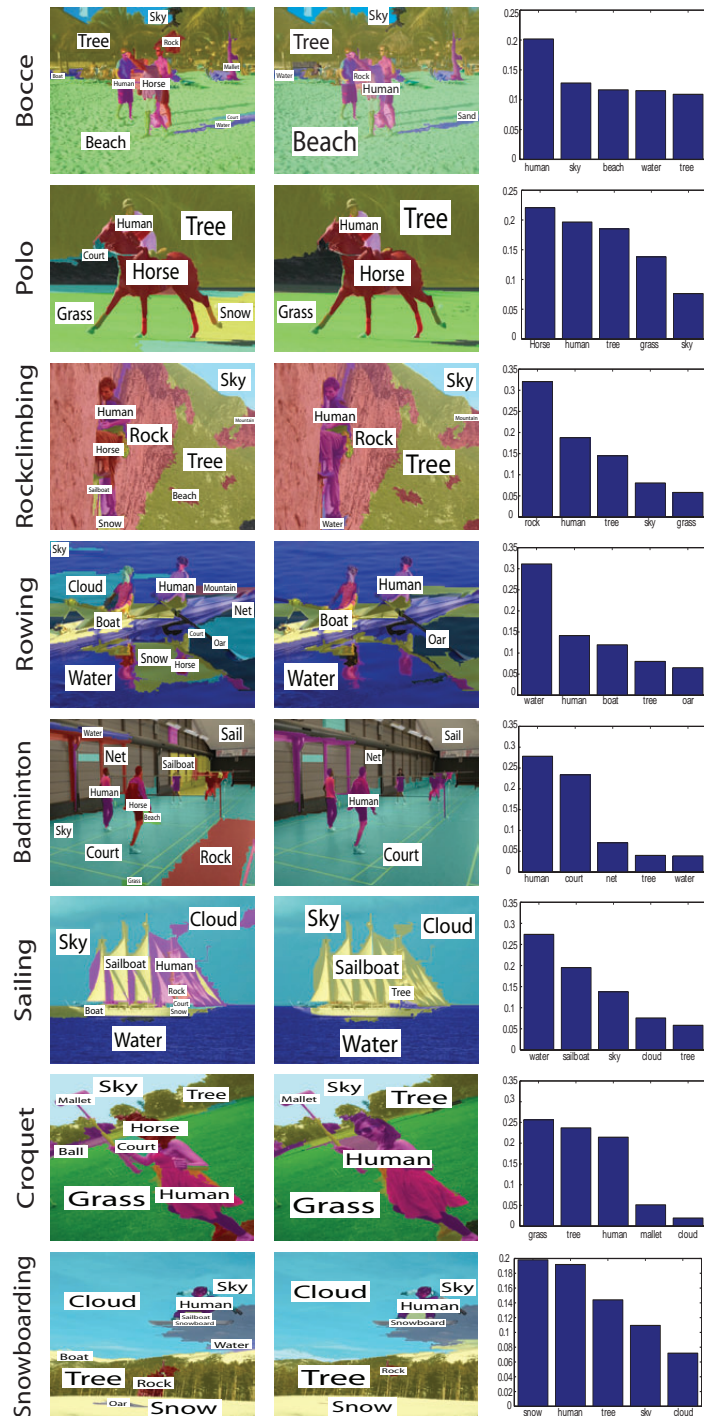


Figure 2.12: Comparison of object segmentation results with or without the top down scene class influence. Each triplet of images show results of one scene class (**Experiment F**). The **left** image shows object segmentation result without the top down contextual information, i.e. by setting the probability distribution of object given scene class to a fixed uniform distribution. The **center** image shows object segmentation result by using the full model. We observe objects are more accurately recognized and delineated. The **right** image shows the probability of the 5 most likely objects per scene class. This probability encodes the top down contextual information.

ultimate goal is to provide personal assistance to visually-impaired or blind people. Currently, other than specific domain applications such as texts and faces, little technology is available to assist them to interpret and analyze the visual environment in a comprehensive and meaningful way. Our project aims to contribute to both the scientific questions of image modeling and the technological advancement of visual intelligence. Towards this goal, we have proposed principled probabilistic models towards complete understanding of complex scene images. Hierarchical models are developed to unify the patch-level, object-level, and scene-level information. We have introduced our first attempt to recognize event/complex scene images by combining objects and scene environment in the event images. We then extend this model to a probabilistic model for simultaneous classifying, annotating and segmenting challenging complex scene images, providing a principled probabilistic treatment of noisy tags often seen in real-world data. Through an automatic training framework, we show that our model outperforms state-of-the-art methods in classifying, labeling and segmenting complex scene images.

Chapter 3

Probabilistic Model for Automatic Image Organization

Semantic image understanding of individual images is helpful for inferring the relationship among images based on visual content. The growing popularity of digital cameras allows us to easily capture and share meaningful moments in our lives, resulting in giga-bytes of digital images stored in our hard-drives or uploaded onto the Internet. While it is enjoyable to take, view and share pictures, it is tedious to organize them. Hierarchies are a natural way to organize concepts and data [15]. A meaningful image hierarchy can ease the human effort in organizing thousands and millions of pictures, making image organization, browsing and searching more convenient and effective (Fig. 3.1). In this chapter, we introduce a non-parametric hierarchical model for automatically constructing a semantically and visually meaningful hierarchy of texts and images on the Internet.

Two types of hierarchies have recently been explored in computer vision: language-based hierarchy and low-level visual feature based hierarchy. Pure language-based lexicon taxonomies, such as WordNet [103, 125], have been used in vision and multimedia communities for tasks such as image retrieval [80, 81, 33] and object recognition [99, 130]. While these hierarchies are useful to guide the semantically meaningful organization of images, they ignore important visual information that connects images together. For example, concepts such as snowy mountains and a skiing activity

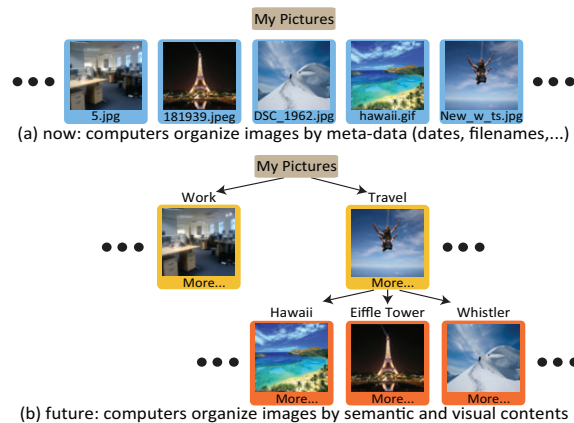


Figure 3.1: Traditional ways of organizing and browsing digital images include using dates or filenames, which can be a problem for large sets of images. Images organized by semantically meaningful hierarchy could be more useful.

are far from each other on the WordNet hierarchy, while visually they are close. On the other hand, a number of purely visual feature based hierarchies have also been explored recently [65, 98, 123, 3, 7]. They are motivated by the observation that the organization of the image world does not necessarily follow a language hierarchy. Instead, visually similar objects and concepts (e.g. shark and whale) should be close neighbors on an image hierarchy, a useful property for tasks such as image classification. But visual hierarchies are difficult to interpret – none of the work has a quantitatively evaluated of the effectiveness of the hierarchies directly. It is also not clear how useful a purely visual hierarchy is.

Motivated by having a more meaningful image hierarchy useful for end-tasks such as image annotation and classification, we propose a method to construct a *semantivisual* hierarchy, which is built upon both semantic and visual information related to images. Specifically, we make the following contributions:

1. Given a set of images and their tags, our algorithm automatically constructs a hierarchy that organizes images in a general-to-specific structure.
2. Our quantitative evaluation by human subjects shows that our semantivisual image hierarchy is more meaningful and accurate than other hierarchies.

3. Serving as a knowledge ontology, our image hierarchy performs better on image classification and annotation.

3.1 Related Work

Building the Image Hierarchy. Several methods [3, 7, 65, 98, 123] have been developed for building image hierarchies from image features. Most of them assess the quality of the hierarchies by using end tasks such as classification, and there is little discussion on how to interpret the hierarchies¹. We emphasize an automatic construction of semantically meaningful image hierarchies and quantitative evaluations of the constructed hierarchy.

Using the Image Hierarchy. A semantically meaningful image hierarchy can be useful for several end-tasks such as classification, annotation, searching and indexing. In object recognition, using WordNet [103] has led to promising results [99, 130]. Here, we focus on exploiting the image hierarchy for three image related tasks: classification (e.g., “Is this a wedding picture?”), annotation (e.g., a picture of water, sky, boat, and sun) and hierarchical annotation (e.g., a picture described by photo → event → wedding → gown).

Most relevant to our work are those methods matching pictures with words [5, 37, 23, 16, 138, 92]. These models build upon the idea of associating latent topics [69, 17, 15] related to both the visual features and words. Drawing inspiration from this work, our approach differs by exploiting a image hierarchy as a knowledge ontology to perform image annotation and classification. We are able to offer hierarchical annotations of images that previous work cannot, making our algorithm more useful for real world applications like album organization.

Album Organization in Multi-media Research. Some previous work has been done in the multimedia community for album organization [22, 77, 29]. These algorithms treat album organization as an annotation problem. Here, we build a general semantivisual image hierarchy. Image annotation is just one application of

¹[123] has provided some interesting insights of their hierarchy by labeling the nodes with class names of human segmented image regions [141] but without any quantitative evaluations.

our work.

3.2 Building the Semantivisual Image Hierarchy

Our research in building such a semantically meaningful image hierarchy considers the following issues:

- Images should cluster meaningfully at all levels of the hierarchy. Tags related to the images should be correctly assigned to each node of the hierarchy;
- Our algorithm organizes the images in a general-to-specific relationship which is deliberately less strict compared to formal linguistic relations;
- It is unclear what a semantically meaningful image hierarchy should look like in either cognitive research [117] or computer vision. Indeed formalizing such relations would be a study of its own. We follow a common wisdom - the effectiveness of the constructed image hierarchy is quantitatively evaluated by both human subjects and end tasks.

Sec. 3.2.1 details the model. Sec. 3.2.2 sketches out the learning algorithm. Sec. 3.2.3 visualizes our image hierarchy and presents the quantitative evaluations by human subjects.

3.2.1 A Hierarchical Model for both Image and Text

We use a multi-modal model to represent images and textual tags on the semantivisual hierarchy (Fig. 3.2). Each image is decomposed into a set of over-segmented regions $\mathbf{R} = [R_1, \dots, R_r, \dots, R_N]$, and each of the N regions is characterized by four appearance features – color, texture, location and quantized SIFT [94] histogram of the small patches within each region. An image and its tags $\mathbf{W} = [W_1, \dots, W_w, \dots, W_M]$ form an image-text pair. M is the number of distinct tags for this image. Each image is associated with a path of the hierarchy, where the image regions can be assigned to different nodes of the path, depending on which visual concept the region depicts. For

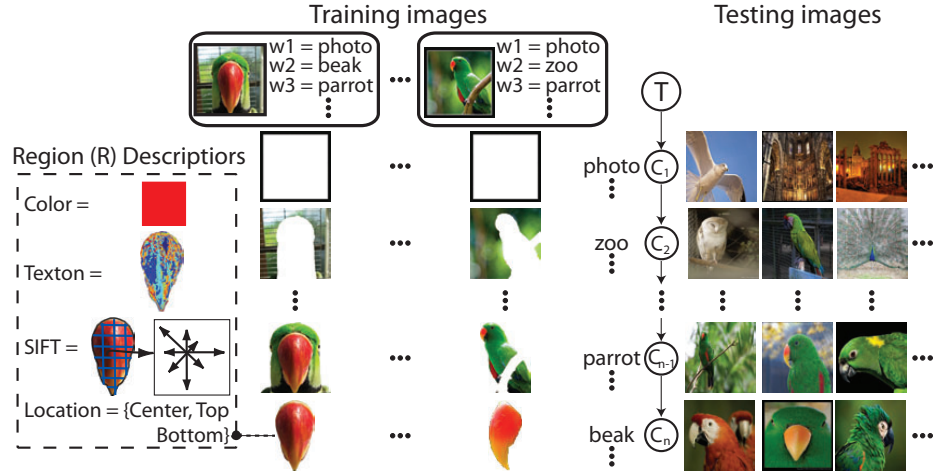
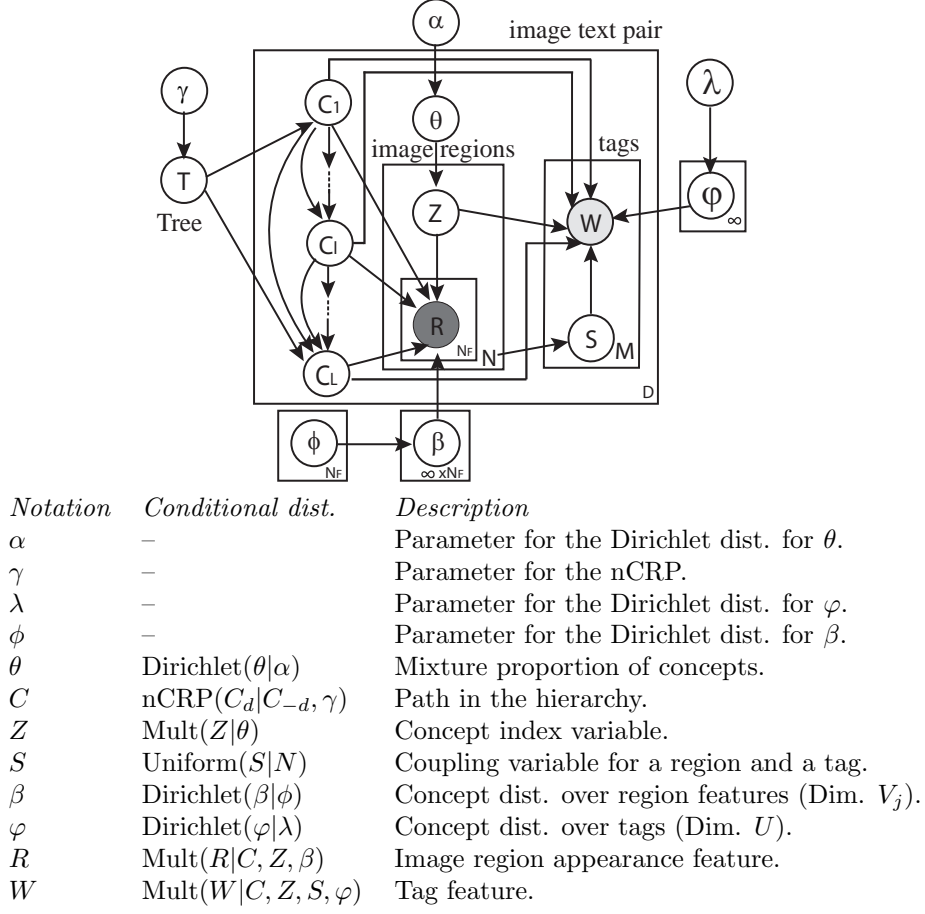


Figure 3.2: Schematic illustration of associating a training image in the semantivisual hierarchical model (**left**) and assigning a test image to a node on a given path of the hierarchy (**right**). The hierarchical model is summarized in variable T , where only one path is explicitly drawn from C_1 to C_n . **Left of the model:** Two training images and their Flickr tags are shown. Each image is further decomposed into regions. Each region is characterized by the features demonstrated in the bounding box on the left. A region is assigned to a node that best depicts its semantic meaning. **Right of the model:** A query image is assigned to a path based on the distribution of the concepts it contains. To further visualize the image on a particular node of the path, we choose the node that corresponds to the dominating region concepts in the image.

example, in the “photo→zoo→parrot→beak” path, a foliage region of a bird photo is likely to be associated with the more general “zoo” node, whereas a region containing the bird beak is likely to be assigned to the leaf node “beak”.

Fig. 3.3 (Top) shows the graphical model. Each image-text pair (\mathbf{R}, \mathbf{W}) is assigned to a path $\mathbf{C}_c = [C_{c_1}, \dots, C_{c_l}, \dots, C_{c_L}]$ in the infinite image and text hierarchy $T = [C_1, \dots, C_c, \dots, C_\infty]$. Here l indicates the level in the path, with L the maximum. The path is sampled from an nCRP(γ) (nested Chinese Restaurant Process)[15], where γ is a parameter controlling the branching probability.

Let d be the index of an image-text pair, with N_d regions, M_d tags in this image-text pair, and $N_F = 4$ types of region descriptors indexed by j . The joint distribution


 Figure 3.3: The graphical model (**Top**) and the notations of the variables(**Bottom**).

of all random variables (hidden and observed) is

$$\begin{aligned}
 p(\mathbf{C}, \boldsymbol{\theta}, \mathbf{Z}, \mathbf{R}, \mathbf{S}, \mathbf{W}, \boldsymbol{\beta}, \boldsymbol{\varphi} | \alpha, \phi, \lambda) = & \\
 \prod_{c \in T} \prod_{j=1}^4 p(\beta_{j,c} | \phi_j) p(\varphi_c | \lambda) \prod_{d=1}^D p(\mathbf{C}_d | \mathbf{C}_{1:d-1}) p(\boldsymbol{\theta}_d | \alpha) & \\
 \prod_{r=1}^{N_d} p(Z_{d,r} | \boldsymbol{\theta}_d) \prod_{j=1}^4 p(R_{d,r,j} | \mathbf{C}_d, Z_{d,r}, \boldsymbol{\beta}) & \\
 \prod_{w=1}^{M_d} p(S_{d,w} | N_d) p(W_w | \mathbf{C}_d, \mathbf{Z}_d, S_{d,w}, \boldsymbol{\varphi}), & \quad (3.1)
 \end{aligned}$$

where λ, ϕ_j, α are Dirichlet priors for the mixture proportion of concepts θ , region appearances given concept β , and words given concept φ . The conditional distribution of \mathbf{C}_d given $\mathbf{C}_{1:d-1}$, $p(\mathbf{C}_d | \mathbf{C}_{1:d-1})$, follows the nested Chinese restaurant process (nCRP).

Remarks. The image part of our model is adapted from the nCRP [15], which was later applied in vision in [7, 123]. We improve this representation by coupling images and their tags through a correspondence model. Inspired by [16], we use the coupling variable S to associate the image regions and the tags. In our model, the correspondence of tags and image regions occurs at the nodes in the hierarchy. This differentiates our work from [3, 7, 65, 98, 123]. Both textual and visual information serve as bottom up information to estimation of the hierarchy. As shown in Fig. 3.5, by combining the tag information, the constructed image hierarchy becomes semantically meaningful, since textual information are often more descriptive. A comparison between our model and [15] demonstrates that our visual-textual representation is more effective than the language-based representation (Fig. 3.6).

3.2.2 Learning the Semantivisual Image Hierarchy

Given a set of unorganized images and user tags associated with them (e.g. Flickr images and user tags), the goal of learning is to estimate an image hierarchy in which images and tags of the same concept can be associated with each other via learning of the concept index Z and the coupling variable S . In addition, their location in the hierarchy is estimated by learning the concept index Z and the path C . This involves computing the posterior distribution of the hidden variables given the observations. However, this posterior is intractable to compute in our model. We use an approximation algorithm i.e. Gibbs sampling [62]. Gibbs sampling defines a Markov chain whose stationary distribution is the posterior of interest. The chain is defined by iteratively drawing each hidden variable from its conditional distribution given the other hidden variables and observations. We use a collapsed version of Gibbs sampling algorithm by integrating out β , φ and θ . It samples the concept index Z , the coupling variable S and the path C .

Sampling concept index Z . The conditional distribution of a concept index of a particular region depends on 1) the likelihood of the region appearance, 2) the likelihood of tags associated with this region and 3) the concept indices of the other regions in the same image-text pair. Since the path assignment for the image-text pair is fixed at this step, the resampled concept index is *restricted* to this path. For

the r th region of d th image-text pair, let $S_r = \{w : S_{d,w} = r\}$ be the set of tags associated with this region r ,

$$\begin{aligned}
 & p(Z_{d,r} = l | \text{rest}) \propto \\
 & p(Z_{d,r} = l | Z_d^{-r}, \alpha) \prod_{j=1}^4 p(R_{d,r,j} | \mathbf{R}^{-dr}, \mathbf{C}_d, \mathbf{Z}, \phi_j) \\
 & p(\{W_{d,w} : w \in S_r\} | \mathbf{W}^{-dw:w \in S_r}, \mathbf{C}_d, Z_{d,r}, \lambda) = \\
 & \prod_{j=1}^4 \frac{n_{C_{d,l},j,R_{d,r,j}}^{-dr} + \phi_j}{n_{C_{d,l},j,\cdot}^{-dr} + V_j \phi_j} \times \prod_{w \in S_r} \frac{n_{C_{d,l},W_{d,w}}^{-dw} + \lambda}{n_{C_{d,l},\cdot}^{-dw} + U\lambda} \times \frac{n_{d,\cdot}^{-r} + \alpha}{n_{d,\cdot}^{-r} + L\alpha},
 \end{aligned}$$

where $n_{d,l}^{-r}$ is the number of regions in the current image assigned to level l except the r th region, $n_{C_{d,l},j,R_{d,r,j}}^{-dr}$ is the number of regions of type j , index $R_{d,r,j}$ assigned to node $C_{d,l}$ except the r th region in image-text pair d , and $n_{C_{d,l},W_{d,w}}^{-dw}$ is the number of tags of index $W_{d,w}$ assigned to node $C_{d,l}$ except the w th region in image-text pair d . Marginal counts are represented with dots.

Sampling coupling variable S . Coupling variable S couples the image regions with the tags. Since it has a uniform prior over the number of regions, its conditional distribution solely depends on the likelihood of the tag, i.e. how frequently one specific tag is assigned to a node through an image region. Note that the path assignment is still fixed at this step. The conditional probability is

$$\begin{aligned}
 p(S_{d,w} = r | \text{rest}) & \propto p(W_{d,w} | S_{d,w} = r, \mathbf{S}^{-dw}, \mathbf{W}^{-dw}, \mathbf{Z}_d, \mathbf{C}_d, \lambda) \\
 & = \frac{n_{C_{d,Z_{d,r},W_{d,w}}^{-dw} + \lambda}{n_{C_{d,Z_{d,r},\cdot}^{-dw} + U\lambda}.
 \end{aligned}$$

Sampling path C . The path assignment of a new image-text pair is influenced by the previous arrangement of the hierarchy and the likelihood of the image-text pair:

$$p(\mathbf{C}_d | \text{rest}) \propto p(\mathbf{R}_d, \mathbf{W}_d | \mathbf{R}_{-d}, \mathbf{W}_{-d}, \mathbf{Z}, \mathbf{C}, \mathbf{S}) p(\mathbf{C}_d | \mathbf{C}_{-d}),$$

where $p(\mathbf{C}_d | \mathbf{C}_{-d})$ is the prior probability induced by nCRP and $p(\mathbf{R}_d, \mathbf{W}_d | \mathbf{R}_{-d}, \mathbf{W}_{-d}, \mathbf{Z}, \mathbf{C}, \mathbf{S})$ is the likelihood,



Figure 3.4: Example images from each of the 40 Flickr classes.

$$p(\mathbf{R}_d, \mathbf{W}_d | \mathbf{R}_{-d}, \mathbf{W}_{-d}, \mathbf{Z}, \mathbf{C}, \mathbf{S}) \propto \prod_{w=1}^{M_d} \frac{n_{C_d, Z_d, S_d, w}^{-d} W_{d, w}^{+ \lambda}}{n_{C_d, Z_d, S_d, w}^{-d} + U \lambda} \times \prod_{l=1}^L \prod_{j=1}^4 \left(\frac{\Gamma(n_{C_{d, l}, j, \cdot}^{-d} + V_j \phi_j)}{\prod_v \Gamma(n_{C_{d, l}, j, v}^{-d} + \phi_j)} \times \frac{\prod_v \Gamma(n_{C_{d, l}, j, v}^{-d} + n_{C_{d, l}, j, v}^d + \phi_j)}{\Gamma(n_{C_{d, l}, j, \cdot}^{-d} + n_{C_{d, l}, j, \cdot}^d + V_j \phi_j)} \right).$$

The Gibbs sampling algorithm samples the hidden variables iteratively given the conditional distributions. Samples are collected after the burn in.

3.2.3 A Semantivisual Image Hierarchy

We use a set of 4,000 user uploaded images and 538 unique user tags² across 40 image classes from Flickr³ to construct a semantivisual image hierarchy. The average number of tags for each image is 4. As Fig. 3.4 shows, photos from real-world sources are very challenging to average, even for human. We detail in this section how we conduct our evaluation and how effective our image hierarchy is compared to other hierarchies.

Implementation. Each image is divided into small patches of 10×10 pixels, as well as a collection of over-segmented regions based on color, brightness and texture homogeneity [4]. Each patch is assigned to a codeword in a codebook of 500 visual

²Incorrectly spelled words and adjectives are omitted.

³The image classes are: animal, bride, building, cake, child, christmas, church, city, clouds, dessert, dinner, flower, spring, friends, fruit, green, high-school, calcio, italy, europe, london, love, nature, landscape, macro, paris, party, present, sea, sun, sky, seagull, soccer, reflection, sushi, vacation, trip, water, silhouette, and wife.

words obtained by applying K-means clustering to the 128-dim SIFT features extracted from 30 randomly chosen images per class. Similarly, we obtain our 4 region codebooks of size 100, 50, 100 and 100 for color (HSV histogram), location (vector quantization of the region center, top and bottom position), texture (normalized textron [88] histogram) and normalized SIFT histogram respectively. To speed up learning, we initialize the levels in a path by assigning the regions with high tf-idf (term frequency-inverse document frequency) scores in one of the visual feature to the leaf node and those with low tf-idf scores to the root node. It takes about 2 hours to learn the hierarchy from 4,000 images and 30 minutes for test on 4,000 images on a PC with an Intel 2.66GHz CPU.

Visualizing the semantivisual hierarchy. For all 4,000 images and 538 tags, we obtain a hierarchy of 121 nodes, 4 levels and 53 paths. Fig. 3.5 visualizes in more details different parts the hierarchy. Our observations are as follows.

- The general-to-specific relationship is observed in most parts of the hierarchy. The root node contains images that are difficult to be named, but fall under the general category of “photo”. Directly under the root, images are organized into “architecture”, “garden”, “event”, “food”, etc. Examine the leftmost path of the “event” subtree. This path is about photos taken at wedding events. The leaf node of this path is “wedding gown”, a child of “wedding” and a sister of “wedding flower”. This organization can be useful for browsing large photo libraries. The users no longer have to remember different dates of various wedding events. Instead, they can quickly access the wedding concept and its related classes.
- We have argued that purely visual information sometimes cannot provide semantically meaningful image hierarchy. As demonstrated by the “event” subtree, it is difficult to imagine that pictures of “dancing at a birthday party” can be a sister node to “birthday cake” based only on low-level image features. Our semantivisual hierarchy offers a connection between these two groups via the parent of “birthday.”

- Similarly, a purely language-based hierarchy would be likely to miss close connections such as “tower” and “business district” (in the “architecture” subtree). In WordNet, “tower” and “business district” have to traverse 15 inherited parent nodes to reach each other.
- Our hierarchy illustrates that images assigned to each node are diverse. It is easy to predict that for nodes at the higher levels, the visual appearance of images are diverse because the semantic meaning of the nodes is general. For example, “food” can be interpreted as “sushi”, “cake”, or “dinner”. As one traverses down along a path, concepts represented in the nodes become more specific. However even at the bottom levels such as “sugar” and “cheese”, the images are diverse. This is because of the tightly coupled clustering of images using both the visual and textual information. A purely visual-feature based algorithm would not be able to achieve this.

A quantitative evaluation of image hierarchies. Evaluating the effectiveness of an image hierarchy is not an easy task. What makes a meaningful image hierarchy? We consider two criteria for evaluation: 1) good clustering of images that share similar concepts, i.e., images along the same path, should be more or less annotated with similar tags; 2) and good hierarchical structure given a path, i.e., images and their associated tags at different levels of the path, should demonstrate good general-to-specific relationships. To measure if an image on a path associates well with the set of concepts depicted by this path, we present human subjects trials in which each image and six word concepts are presented (Fig. 3.6). Inspired by [25], we present five of the six tag concepts associated with the path of the image (learned by the model) and one randomly chosen tag concept that is unlikely to be in the path. The subject is asked to select which set of tags are unrelated to the image (Fig. 3.6 (top left)). In the ideal case, if the image path is effective, then it is more likely that the randomly chosen word would be the only irrelevant concept to the image. An Amazon Mechanical Turk (AMT) experiment is set up for this evaluation. We compare our hierarchy with one that is obtained by using only text clustering [15]. Fig. 3.6 (top right) shows that our hierarchy is more effective than the purely text-based method.



Figure 3.5: Visualization of the learned image hierarchy. Each node on the hierarchy is represented by a colored plate, which contains four randomly sampled images associated with this node. The color of the plate indicates the level on the hierarchy. A node is also depicted by a set of tags, where only the first tag is explicitly spelled out. The **top** subtree shows the root node “photo” and some of its children. The rest of this figure shows six representative sub-trees of the hierarchy: “event”, “architecture”, “food”, “garden”, “holiday” and “football”.

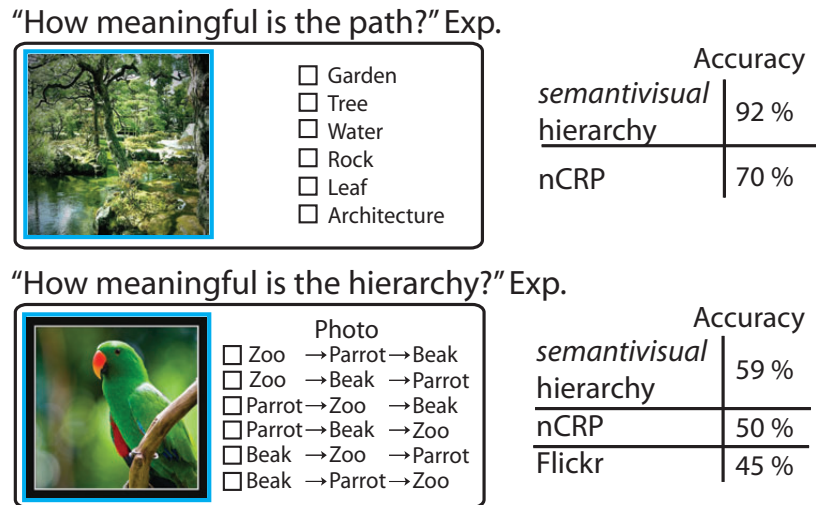


Figure 3.6: Evaluation of the hierarchy. **Top:** “How meaningful is the path” experiment. **Top Left:** The AMT users are provided with a list of words. The users need to identify the words that are not related to the image to the left. **Top Right:** Quantitative results of our hierarchy and nCRP [15]. Our hierarchy performs the best by incorporating the visual information associated to the tags. **Bottom:** “How meaningful is the hierarchy” experiment. **Bottom Left:** The AMT users are provided with all permutations of candidate words from the path corresponding to the image that correctly represents the hierarchical structure. **Bottom Right:** Quantitative results of our hierarchy, nCRP [15] and Flickr. All three algorithms use exactly the same tag input to construct the hierarchy.

The second metric measures how good the hierarchical relations are in our image hierarchy. Again we use AMT. We break down the evaluation by path. For each trial, a path of L levels is selected from the hierarchy. The $(L-1)!$ permutations of the nodes in the path⁴ are presented to a human subject, depicted by the text concepts (see Fig. 3.6 (bottom left)). Subjects are instructed to select the path that best illustrates a general-to-specific hierarchical relation. We compare the human selected path (as ground-truth) with the model generated path using modified Damerau-Levenshtein distance. We compare our hierarchy with two purely text-based hierarchies including one obtained by [15] and the default by Flickr. Fig. 3.6 (bottom right) shows that our hierarchy agrees more with human ground-truth than the others.

⁴The root node is kept intact because “photo” is always assigned to it.

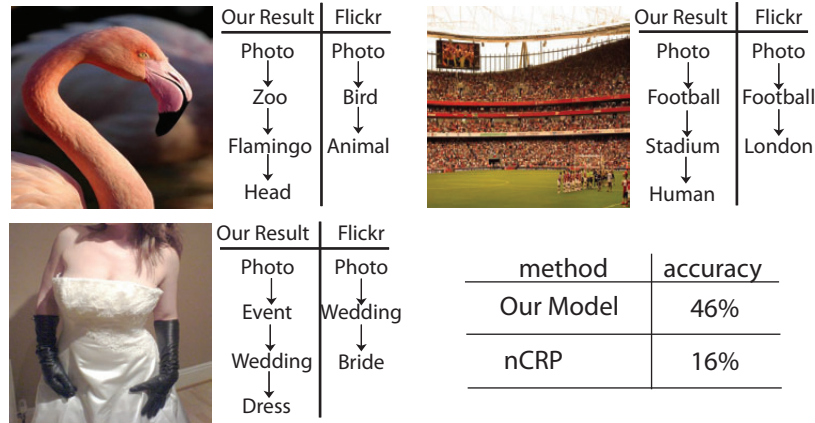


Figure 3.7: Results of the hierarchical annotation experiment. Three sample images and their hierarchical annotations by our algorithm and the original Flickr tags are shown. The table presents quantitative comparison on our hierarchy and nCRP[15]. The performance is measured by the modified Damerau-Levenshtein distance between the proposed hierarchical annotation by each algorithm and the human subjects’ result.

3.3 Using the Semantivisual Image Hierarchy

A good image hierarchy can serve as a knowledge ontology for difficult recognition tasks such as image classification and annotation. In the following experiments, we choose three tasks to show the applications of the hierarchy.

3.3.1 Hierarchical Annotation of Images

Given our learned image ontology, we can propose a hierarchical annotation of an unlabeled query image, such as photo \rightarrow zoo \rightarrow bird \rightarrow flamingo. For an unannotated image I with the posterior path assignments represented as $\mathbb{S}_C = \{\mathbf{C}_I^1, \mathbf{C}_I^2, \dots, \mathbf{C}_I^{|\mathbb{S}_C|}\}$, the probability of tag W for level l is estimated by averaging the paths in \mathbb{S}_C ,

$$p(W|I, \text{level} = l) \approx (1/|\mathbb{S}_C|) \sum_{i=1}^{|\mathbb{S}_C|} p(W|\tilde{\varphi}_i, C_I^i(l)),$$

where $\tilde{\varphi}_i$ is the MAP (maximum a posterior) estimate of tag concept distributions from the training data given the i th sample, $C_I^i(l)$ specifies the node in path \mathbf{C}_I^i at

level l and $p(W|\tilde{\varphi}, C_I^i(l))$ indicates the probability of tag W given node $C_I^i(l)$ and $\tilde{\varphi}$, i.e. $\tilde{\varphi}_{C_I^i(l), W}$.

We show in Fig. 3.7 examples of the hierarchical image annotation results and the accuracy for 4000 testing images evaluated by using our image hierarchy and the nCRP algorithm [15]⁵. Our experiment shows that our semantivisual hierarchical model outperforms the text-only model [15]. There are two reasons. First, [15] cannot perform well on sparse tag words (about 4 tags per image in our dataset). Its proposed hierarchy has many words assigned to the root node, resulting in very few paths. This hierarchy cannot demonstrate the real structure of the image-text data. Second, a simple clustering algorithm such as KNN cannot find a good association between the test images and the training images in our challenging dataset with large visual diversity. In contrast, our model learns an accurate association of visual and text data simultaneously.

3.3.2 Image Labeling

Serving as an image and text knowledge ontology, our semantivisual hierarchy and model can be used for image labeling without a hierarchical relation. This is the image annotation task. For a test image I and its posterior samples $\mathbb{S}_C = \{C_I^1, C_I^2, \dots, C_I^{|\mathbb{S}_C|}\}$ and $\mathbb{S}_Z = \{Z_I^1, Z_I^2, \dots, Z_I^{|\mathbb{S}_Z|}\}$ ($|\mathbb{S}_C| = |\mathbb{S}_Z|$). We estimate the probability of tag W given the image I as,

$$p(W|I) \approx (1/|\mathbb{S}_C|) \sum_{i=1}^{|\mathbb{S}_C|} \sum_{l=1}^L p(W|\tilde{\varphi}_i, C_I^i(l))p(l|Z_I^i),$$

which sums over all the region assignments over all levels. Here $p(l|Z_I^i)$ is the empirical distribution over the levels for image I . In this setting, the most related words will be proposed regardless of which level they are associated to.

Quantitatively, we compare our method with two other image annotation methods: the Corr-LDA [16] and a widely known CBIR method Alipr [89]. We collect the top 5

⁵Note that the original form of [15] is only designed to handle textual data. For comparison purposes, we allow it to annotate images by applying the KNN algorithm to associate the testing images with the training images and represent the hierarchical annotation of the test image by using the tag path of the top 100 training images.




				
Alipr	building photo landscape sky people	card people female fashion cloth	people ocean water landscape snow	38%
Corr-LDA	cake dress garden architecture flower	photo birthday bird architecture portrait	light cloud photo city human	44%
Ours	photo wedding gown bride flower	photo birthday kid cake human	photo cloud sky architecture building	74%

Figure 3.8: Results of the image labeling experiment. We show example images and annotations by using our hierarchy, the Corr-LDA model [16] and the Alipr algorithm [89]. The numbers on the right are quantitative evaluations of these three methods by using an AMT evaluation task.

predicted words of each image by each algorithm and present them to the AMT users. The users then identify if the words are related to the images in a similar fashion as Fig. 3.6(top). Fig. 3.8 shows that our model outperforms Alipr and Corr-LDA according to the AMT user evaluation. As shown in Fig. 3.8(first image column), Alipr tries to propose words such as “landscape” and “photo” which are generally applicable for all images. Corr-LDA provides relatively more related annotation such as “flower” and “garden” based on the co-occurrence of the image appearance and the tags among the training images. Our algorithm provides both general and specific descriptions, e.g. “wedding”, “flower” and “gown”. This is largely because our model captures the hierarchical structure of images and tags.

3.3.3 Image Classification

Finally, we evaluate our model on a highly challenging image classification task. Another 4,000 images are held out as test images from the 40 classes. Each image is represented by the estimated concept distribution over the entire hierarchy. If there are K nodes in the learned hierarchy, the dimension of the distribution is K . Only nodes that are associated to the image have nonzero values in the distribution. We calculate the χ^2 -distances between the concept distribution of the test images and those of the training images. The KNN algorithm is then applied to obtain the class

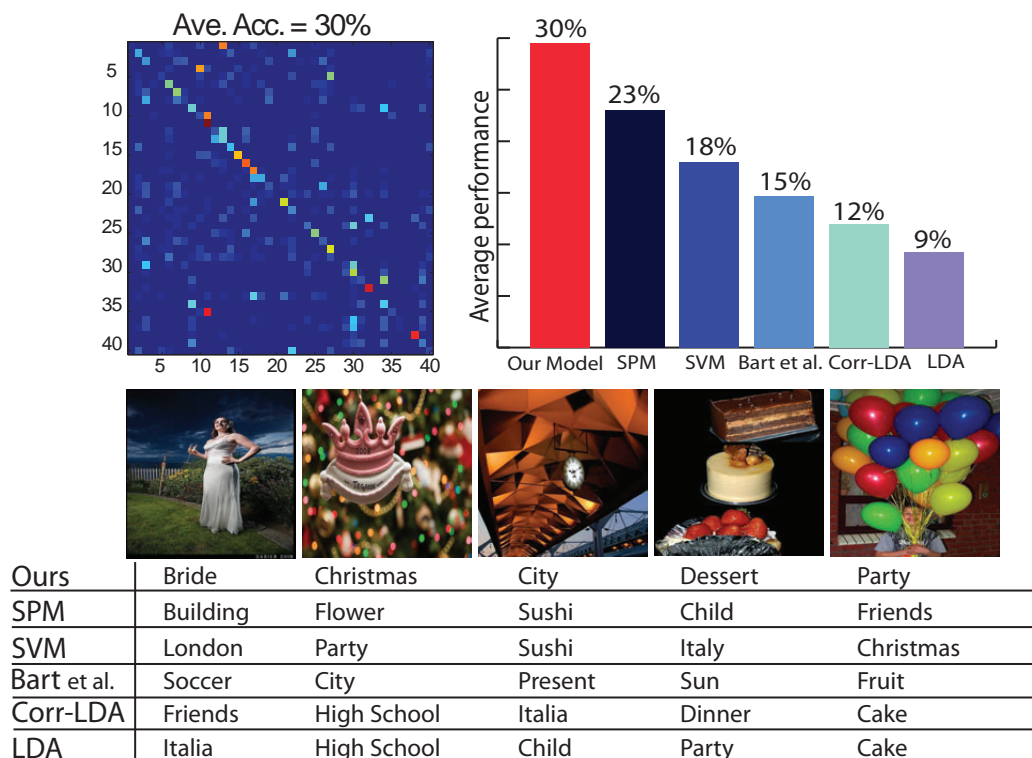


Figure 3.9: Comparison of classification results. **Top Left: Overall performance.** Confusion table for the 40-way Flickr images classification. Rows represent the models for each class while the columns represent the ground truth classes. **Top Right: Comparison with different models.** Percentage on each bar represents the average scene classification performance. Corr-LDA also has the same tag input as ours. **Bottom: classification example.** Example images that our algorithm correctly classified but all other algorithms misclassified.

label. Fig. 3.9 shows the confusion table of classification achieved by our algorithm. In the bar plot in Fig. 3.9, we compare our result to spatial pyramid matching (SPM) [85], SVM [30], Bart et. al. [7], Corr-LDA [16] and LDA [17].

From Fig. 3.9 (top right), we observe that LDA [17] gives the lowest classification performance. This shows that a flat single modality model cannot capture the complex structure of our challenging dataset. The classification performance improves by incorporating semantic meaning of the images in training (Corr-LDA [16]) or a more descriptive hierarchical representation [7]. By encoding semantic meaning to the hierarchy, our semantivisual hierarchy delivers a more descriptive structure, which could

be helpful for classification. Finally, comparison among our algorithm, SPM and SVM demonstrates the importance of semantic meaning in interpreting complicated and noisy real world images such as “Christmas” or “party” photos.

3.4 Discussion

In this chapter, we propose a principle probabilistic model to discover the “semantivisual” image hierarchy by incorporating both image and tag information. We emphasize the importance of quantifying the effectiveness of the learned hierarchy, as well as comparing our method with others in the end-task applications. Our experiments show that humans find our semantivisual image hierarchy more effective than those solely based on texts or low-level visual features. We then use several end tasks to illustrate its wide applications.

While demonstrating the effectiveness in modeling visual recognition tasks related to semantic image understanding, our previous work leaves ample room for improvement especially in another fundamental problem in computer vision, i.e. visual feature representation. In Part III, we demonstrate our novel high level image representation called Object Bank.

Part III

High Level Image Representation for Semantic Image Understanding

Chapter 4

Introduction

High-level image recognition is one of the the most challenging domains in the field of computer vision. Any high-level image recognition task using computer vision algorithms starts with image representation, the process of turning pixels into a vector of numbers for further computation and inference. Of all the modules for a robust high-level image understanding system, the design of robust image representation is of fundamental importance and has been attracting many vision researchers. In the past decade, a great amount of research has been conducted on image representation. Among the image representations widely adopted so far, most of them are low level image representations focusing on describing images by using some variant of image gradients, textures and/or colors (e.g. SIFT [94], filterbanks [58, 112], GIST [109], etc.). However, there exists a large discrepancy between these low level image representations and the ultimate high level image recognition goals, which is the so called “Semantic gap”. One way to close the semantic gap is by deploying increasingly sophisticated models, such as the probabilistic grammar model [149], compositional random fields [79], and probabilistic topic models [46, 127]. While these approaches are based on rigorous statistical formulation, good learning and inference are still extremely difficult. In addition, most of the papers have shown promising results only on small scale datasets. It still remains a very challenging task for the models to bridge the low level representations and the high level visual recognition tasks.

Attribute-based methods have made significant progress in object recognition in

recent few years. Its success in recognition is largely accredited to the introduction of “attributes”, which effectively summarize the low-level image properties [56, 39, 131]. In attribute based recognition, a polar bear can be described as white, fluffy object with paws. Such visual attributes summarize the low-level features into object parts and other properties, and then are used as the building blocks for recognizing the object. Attribute based methods have demonstrated great potential for high-level vision tasks. For example, in [131], the authors build an image representation from a collection of classifier predictions and have achieved promising results in object image classification.

On the other hand, using global/local structure information has proved to be useful to increase the descriptive power of a representation. For example, by applying spatial pyramid structure to bag of words (BoW) representation, [85] proposed the Spatial Pyramid Model that gives superior performance compared with the original BoW features.

Therefore, we hypothesize that object appearance and their spatial locations could be very useful for representing and recognizing images. In this part, we introduce Object Bank, a novel high level image feature to represent complex real-world image by collecting the responses of many object detectors at different spatial locations in the image. Drawing an analogy to low-level image representation, instead of using image filters to represent local texture, we introduce *object filters* to characterize local image properties related to the presence/absence of objects. By using a large number of such object filters, our *object filter bank* representation of the image can provide rich information of the image that captures much of the high-level meaning. Object Bank is a novel high level image representation, which encodes knowledge of objects for challenging high level visual tasks in real world problems such as image classification.

4.1 Background and Related Work

A plethora of image feature detectors and descriptors have been developed for object recognition and image classification [109, 9, 82, 102, 94]. We particularly draw the analogy between our *object bank* and the *texture filter banks* [112, 58]. Instead of

using gradients or colors in an image to represent it, Object Bank characterizes local image properties by using object filters related to the presence/absence of objects, adding more high level information into the representation to bridge the semantic gap.

Object detection and recognition also entail a large body of literature [43, 20]. In this work, we mainly use the current state-of-the-art object detectors of Felzenszwalb et. al. [47], as well as the geometric context classifiers (“stuff” detectors) of Hoem et. al. [70] for pre-training the object detectors.

A few recent works have begun to explore visual attributes based image recognition [131, 39, 56]. These approaches focus on single object classification based on visual attributes. The visual attributes in these approaches does not necessarily directly related to visual pattern in the images, e.g. “carnivore” and “can bite”. Different than these approaches, Object Bank representation encodes semantic and spatial information of objects universally applicable for high level visual recognition tasks.

In [135], a handful number of concepts are learned for describing an image. For each location only the most probable concept is used to form the representation based on binary classification result. Significant amount of information is lost during their feature extraction process. Our approach, on the other hand, encodes the probabilities of all objects candidates appearing in all locations in the image resulting in much richer image representation.

The idea of using object detectors as the basic representation of images is related to work in multimedia by applying a large number of “semantic concepts” to video and image annotation [67]. However, in [67], each semantic concept is trained by using the entire images or frames of video. Understanding cluttered images composed of many objects will be challenging since there is no localization of object concepts in images in this approach. To our knowledge, Object Bank is the first high level image representation that provides probability of objects appearing in images and their spatial locations as the signature of images.

In Chapter 5, we first introduce our new way of representing complex visual-world, Object Bank. As a proof of concept, we apply Object Bank to high level image classification tasks by using simple, off-the-shelf classifiers. It delivers superior image

recognition results to all reported state-of-the-art performance on various benchmark datasets. We further analyze the effectiveness of each ingredient in Object Bank and demonstrate that Object Bank can not only achieve state-of-the-art performance in high level visual recognition tasks but also discover meaningful aspects of objects in an image. In Chapter 6, we explore a supervised feature selection method to make our representation more efficient and reveal semantically meaningful feature patterns. Lastly, in Chapter 7, we propose an unsupervised multi-level structured sparse image coding approach to compress Object Bank to a lower-dimensional and more compact encoding of the image features while preserving and accentuating the rich semantic and spatial information of Object Bank.

Chapter 5

The Object Bank Representation of Images

In this chapter, we introduce the concept of high-level Object Bank (OB) representation. The ultimate goal of Object Bank representation is to encode as much objects information including their semantic meaning, spatial locations, sizes and view points etc. as possible for representing complex scene images. We achieve this by constructing Object Bank, a collection of object filters trained on multiple objects with different view points.

5.1 Construction of Object Bank

Fig. 5.1 illustrates our Object Bank representation construction process. Given an image, an *object filter* response can be viewed as the response of a “generalized object convolution.” We obtain object responses by running a bunch of object filters (each is an object detector trained from images with similar view point) across an image at various locations and scales by using the sliding window approach. For each scale and each detector, we obtain an initial response map, whose value at each location indicates the possibility of the occurrence of that object. To capture the spatial location property of objects, we build a spatial pyramid for the response map. At each layer of the spatial pyramid structure, we extract the signal from all grids.

Finally, we build the Object Bank representation by concatenating all the extracted responses. We summarize the representations generated by using different pooling methods below:

Max response representation (OB-Max, Fig. 5.1). As shown in Fig. 5.1, OB-Max encodes the strongest object filter response at each grid and each level of the spatial pyramid and detector scale. In our case, we have 177 objects, 12 scales and 21 spatial pyramid grid($L=2$), which is in 44604 dimension in total. In the example image shown in Fig. 5.1, given the OB-Max representation, the possibility of sailboat and water appearing in that specific grid is higher than other objects. If not specified, OB-Max is the default Object Bank pooling method in this chapter.

Average response representation (OB-Avg, Fig. 5.1). OB-Avg encodes the average object filter response in each grid at each level of the spatial pyramid and detector scale. From the responses to different object filters, we form the Object Bank feature by representing the (scene/event) image as a vector of average values from each spatial pyramid grid.

Histogram response representation (OB-Hist, Fig. 5.1). OB-Hist captures more detailed information of the object filters than OB-Max. Instead of using maximum response value of each object detector in each grid of the spatial pyramid representation of each of the response map, we construct a histogram of the responses in each grid. The histogram has a vector length of the number of objects, and the value of each bin is indicated by the number of pixels with the response value within that bin. Four histogram bins in each grid are used in the experiment. As the example in Fig. 5.1 illustrates, within that specific grid, most response values of sailboat detector are high whereas most response values of bear detector are low.

5.2 Implementation Details of Object Bank

So what are the “objects” to use in the Object Bank? And how many? An obvious answer to this question is to use all objects. As the detectors become more robust, especially with the emergence of large-scale datasets such as LabelMe [119] and ImageNet [34], this goal becomes more reachable.

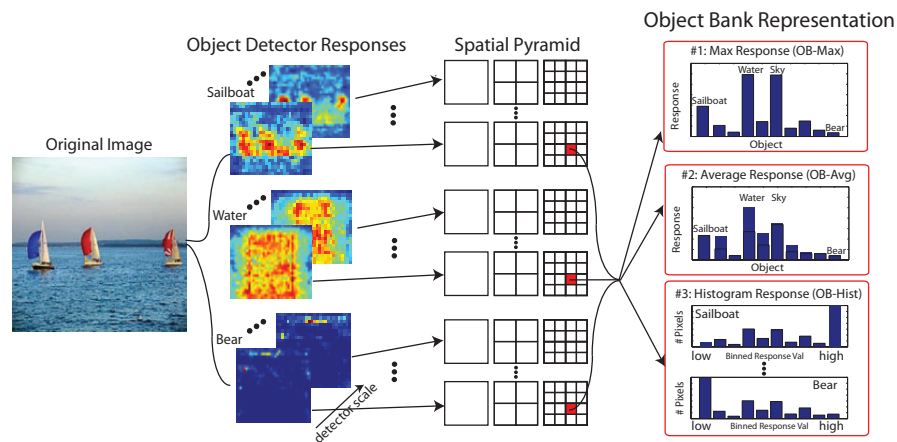


Figure 5.1: (Best viewed in colors and magnification.) Illustration of the object filter representations. Given an input image, we first run a large number of object detectors at multiple scales to obtain the object responses, the probability of objects appearing at each pixel. For each object at each scale, we apply a three-level spatial pyramid representation of the resulting object filter map, resulting in $\text{No. Objects} \times \text{No. Scales} \times (1^2 + 2^2 + 4^2)$ grids. An Object Bank representation of an image is a concatenation of statistics of object responses in each of these grids. We consider three ways of encoding the information. The first is the *max response representation (OB-Max)*, where we compute the maximum response value of each object, resulting in a feature vector of No. Objects length for each grid. The second is the *average response representation (OB-Avg)*, where we extract the average response value in each grid. The resulting feature vector has the same length as the maximum response. The third is the *histogram representation (OB-Hist)*. Here for each of the object detectors, we keep track of the percent of pixels on a discretized number of response values, resulting in a vector of $\text{No. Binned Response Values} \times \text{No. Objects}$ length for each grid.

But time is not fully ripe yet to consider using all objects in, say, the LabelMe dataset. Not enough research has yet gone into building robust object detector for tens of thousands of generic objects. And even more importantly, not all objects are of equal importance and prominence in natural images. As Fig. 5.2(Left) shows, the distribution of objects follows Zipf’s Law, which implies that a small proportion of object classes account for the majority of object instances.

We choose a few hundred most useful (or popular) objects in images¹. An important practical consideration for our study is to ensure the availability of enough

¹This criterion prevents us from using the Caltech101/256 datasets to train our object detectors [42, 64] where the objects are chosen without any particular considerations of their relevance to daily life pictures.

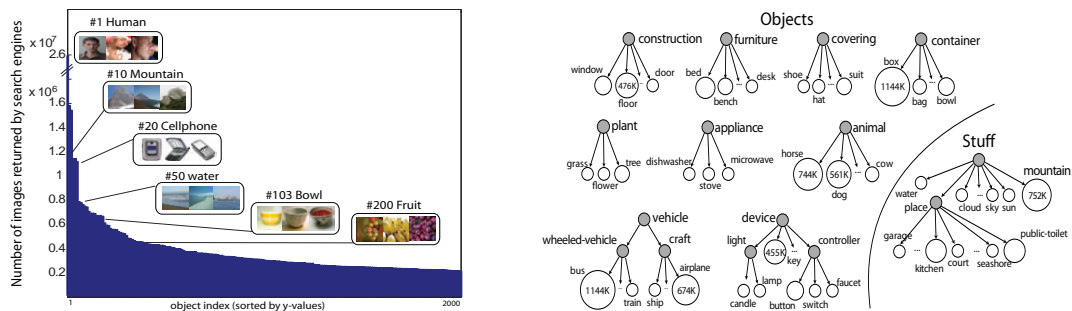


Figure 5.2: (Best viewed in colors and magnification.) **Left:** The frequency (or popularity) of objects in the world follows Zipf’s law trend: a small proportion of objects occurs much more frequently than the majority. While there are many ways of measuring this, e.g., by ranking object names in popular corpora such as the American National Corpora [73] and British National Corpus [38], we have taken a web-based approach by counting the number of downloadable images corresponding to object classes in WordNet on popular search engines such as Google, Ask.com and Bing. We show here the distribution of the top 2000 objects. **Right:** Rough grouping of the chosen object filters based loosely on the WordNet hierarchy [103]. The size of each unshaded node corresponds to the number of images returned by the search.

training images for each object detectors. We therefore focus our attention on obtaining the objects from popular image datasets such as ESP [136], LabelMe [119], ImageNet [34] and the Flickr online photo sharing community. After ranking the objects according to their frequencies in each of these datasets, we take the intersection set of the most frequent 1000 objects, resulting in 177 objects, where the identities and semantic relations of some of them are illustrated in Fig. 5.2(Right). To train each of the 177 object detectors, we use 100~200 images and their object bounding box information from the LabelMe [119] (86 objects) and ImageNet [34] datasets (177 objects). We use a subset of LabelMe scene dataset to evaluate the object detector performance. Final object detectors are selected based on their performance on the validation set from LabelMe.

Before we apply Object Bank representation for visual recognition tasks, we first ask whether this representation encodes discriminative information of images. In Fig. 5.3, we compare the Object Bank image representation to two popular low-level image representations: GIST [109] and the Spatial Pyramid (SPM) representation of SIFT [85]. The low-level feature responses of the two images belonging to different

semantic classes are shown to be very similar to each other, whereas the object filter bank features can easily distinguish such scenes due to the semantic information provided by the object filter responses. In Sec. 5.3, the discriminability of our Object Bank representation is further supported by a series of comparison to the state-of-the-art algorithms based upon low-level image representation on high level visual recognition tasks.

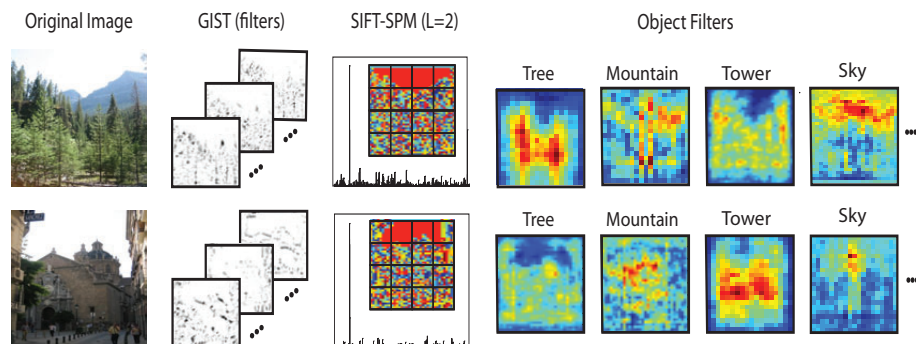


Figure 5.3: (Best viewed in colors and magnification.) Comparison of Object Bank representation with two low-level feature representations, GIST and SIFT-SPM of two types of images, mountain vs. city street. For each input image, we first show the selected filter responses in the GIST representation [109]. Then we show a histogram of the SPM representation of SIFT patches [85] at level 2 of the SPM representation where the codeword map is also shown as a histogram. Finally, we show a selected number of object filter responses.

5.3 High Level Visual Recognition by Using Different Visual Representations

As we try to tackle the high level visual recognition tasks, the semantic gap between low-level features and high-level meanings becomes big. One solution to this is to use complex models to pool in information [127, 91, 132, 92]. But the drawbacks are clear. Researchers have to put large amount of effort to design such complex models. Due to the complexity, they might not scale well with large scale data or different datasets. In addition, some models [127, 91] require extra amount of supervision, which causes such models to be impractical. Can we leverage on relatively simple

statistical models and classifiers, but try to develop descriptive image representation to close the semantic gap better? We hypothesize that by introducing features that are “higher level”, such as Object Bank, we could do this.

While it is good to see a clear advantage of discriminative power of Object Bank over the low level image representations visually, we want to further examine its potential in high level visual recognition tasks on multiple benchmark datasets.

In our experiments, we use simple off-the-shelf classifiers to dissect the contribution of the representations in classification. We compare to related image representations as well as the state-of-the-art approaches with more complex models. Scene classification performance is evaluated by average multi-way classification accuracy over all scene classes in each dataset.

5.3.1 Object Bank on Scene Classification

Before we describe the experiment details, we first introduce the four benchmark scene datasets used in our scene classification experiment, ranging from generic natural scene images (15-Scene [85], LabelMe 9-class scene dataset ²), to cluttered indoor images (MIT Indoor Scene [114]), and to complex event and activity images (UIUC-Sports [91]). We list below the experiment setting for each dataset:

- 15-Scene: we use 100 images in each class for training and rest for testing following [85].
- LabelMe: we randomly draw 50 images from each scene classes for training and 50 for testing.
- MIT Indoor: This is a dataset of 15620 images over 67 indoor scenes assembled by [114]. We follow their experimental setting in [114] by using 80 images from each class for training and 20 for testing.
- UIUC-Sports: This is a dataset of 8 complex event classes. 70 randomly drawn images from each classes are used for training and 60 for testing following [91].

For each dataset, we follow the setting in the paper introduced it and train a multi-class linear SVM.

Fig. 5.4 summarizes the results on scene classification based on Object Bank and a set of well known low-level feature representations: GIST [109], Bag of Words

²From 100 popular scene names, we obtained 9 classes from the LabelMe dataset in which there are more than 100 images: beach, mountain, bathroom, church, garage, office, sail, street, forest. The maximum number of images in those classes is 1000.

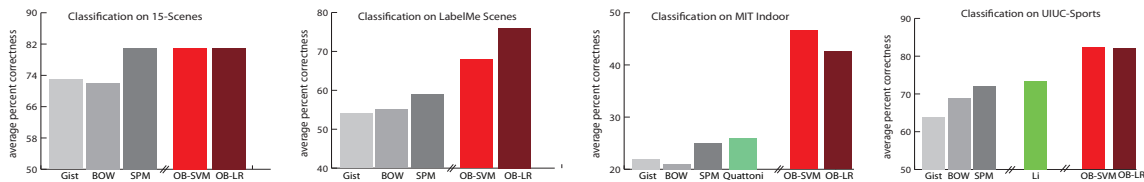


Figure 5.4: (Best viewed in colors and magnification.) Comparison of classification performance of different features (GIST vs. BOW vs. SPM vs. Object Bank) and classifiers (SVM vs. LR) on (top to down) 15 scene, LabelMe, UIUC-Sports and MIT-Indoor datasets. In the LabelMe dataset, the “ideal” classification accuracy is 90%, where we use the human ground-truth object identities to predict the labels of the scene classes. Previous state-of-the-art performance are displayed by using the green bars.

(BOW) [31] and Spatial Pyramid Matching (SPM) [85] on four challenging scene datasets. We also compare the performance of a simple linear SVM model and plain logistic regression built upon Object Bank representation to the existing state-of-the-art algorithms on each benchmark datasets, demonstrating that a semantically meaningful representation can help to reduce the burden of sophisticated models for bridging the “semantic gap” between high level visual recognition tasks and low level image representation³. We achieve substantially superior performances on three out of four datasets, and are on par with the 15-Scene dataset.

The advantage of Object Bank is especially obvious when the images are highly cluttered by objects. Its substantial performance gain on the UIUC-Sports and the MIT-Indoor scene datasets illustrates the importance of using a semantically meaningful representation for complex scenes cluttered with objects. For example, the difference between a living room and a bedroom is less obvious in the overall texture (easily captured by BoW or GIST), but more significant in the different objects and their arrangements. This result underscores the effectiveness of OB, highlighting the fact that in high-level visual tasks such as complex scene recognition, a higher level image representation can be very useful.

³We also evaluate the classification performance of using the detected object location and its detection score of each object detector as the image representation. The classification performance of this representation is 62.0%, 48.3%, 25.1% and 54% on the 15 scene, LabelMe, UIUC-Sports and MIT-Indoor datasets respectively.

Object Bank	Classemes
39%	36%

Table 5.1: Object classification performance by using different high level representations.

5.3.2 Object Bank on Object Recognition

A fundamental task in high level visual recognition is object recognition, in particular, generic object categorization. Generic object categorization is a challenging task owing to the various appearance and locations of objects in the images. Object Bank is constructed from the responses of many objects, which encodes the semantic and spatial information of objects within images. It can be naturally applied to object recognition task. We compare to classemes [131], an attribute based representation obtained as the output of a large number of weakly trained concept classifiers on the image without considering the spatial location and semantic meaning of objects. By encoding the spatial locations of the objects within an image, Object Bank significantly outperforms [131] on the 256-way classification task, where performance is measured as the average of the diagonal values of a 256×256 confusion matrix. The improvement of Object Bank over classemes underscores the importance of rich spatial information of objects and semantic meaning of objects encoded in Object Bank.

In Sec. 5.4, we analyze the effectiveness of both components in detail and demonstrate the advantage of using rich spatial information and semantic meaning of objects. We examine the most effective objects in each of the scene types. We further illustrate interesting patterns of object relationships discovered from our Object Bank representation, which can serve as potential contextual information for advanced models for high level visual tasks such as object detection, segmentation and scene classification.

5.4 Analysis: Role of Each Ingredient

In this section, we thoroughly analyze the role of each important component of our Object Bank representation in a systematical fashion. We demonstrate the designing rationale of Object Bank representation, the influence of detector quality and different components of Object Bank, and eventually provide a good understanding of Object Bank and how to construct a good Object Bank representation.

The ideal object filters is capable of capturing object appearance accurately without losing significantly useful information during the process of construction. Here, we investigate the effectiveness of different designing choices.

1. An robust object detector is able to produce accurate responses of object indicating the probability of the object appearing at each pixel in an image. We first examine the quality of different object detection algorithms [32, 47] as accurate appearance object filters.
2. Object view points vary over a wide range in different images. Multiple view points are essential for training our object filters to capture the multiple views of objects. We evaluate the effectiveness of the filters trained from different view points.
3. Object sizes could differ significantly in different images. We run object filters at different scales to incorporate responses of multiple object sizes. We examine the effectiveness of the different scales and the accumulated scales with an emphasis on the importance of using multiple scales.
4. To capture the various locations of objects in images, we apply a spatial pyramid structure over the responses generated by object filters. We analyze the necessity of constructing the spatial pyramid structure.
5. Our Object Bank representation is a collection of statistics based upon responses to the object detectors. We further examine the influence of different pooling methods on extracting the statistics from the response map.

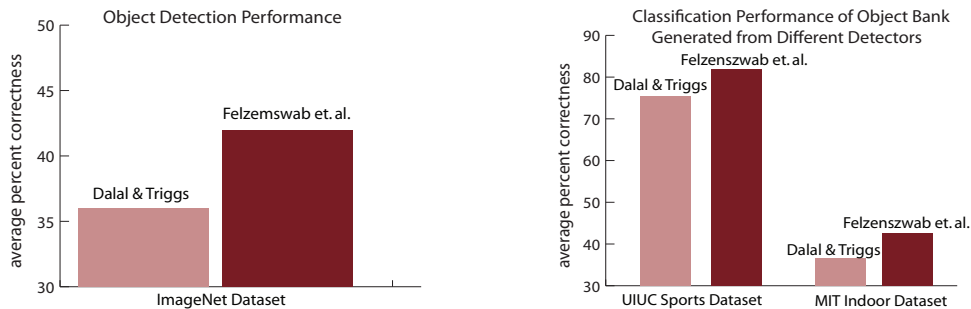


Figure 5.5: **Left:** Detection performance comparison of different detection methods on ImageNet objects. **Right:** Classification performance of different detection methods on UIUC sports dataset and MIT Indoor dataset.

- Objects are the most critical designing component in our Object Bank representation. Finally, we analyze different schemes for selecting objects for Object Bank construction.

In the following experiments, we measure the importance of each component of Object Bank based on its contribution to recognition of scene images in different benchmark datasets. If not specified, we employ simple plain logistic regression as the classifier. Classification performance is obtained from 5-fold random sampling of the training and testing examples.

5.4.1 Comparison of Different Types of Detectors

The first question is what type of object detectors/filters we should use in Object Bank. We are interested in examining the difference between a more sophisticated object detector LSVM [47] and a simple object detector Dalal & Triggs [32]. We first compare how well the detectors capture the object appearance based upon detection performance of LSVM and Dalal & Triggs on object categories from the ImageNet dataset.

As demonstrated in left panel of Fig. 5.5, the performance of Object Bank based on stronger object detectors (LSVM) is better than that of Object Bank representation based on Dalal & Triggs. This reflects that a strong object detector captures the object identity and location in an image more accurately, hence provides better description of an image. We envisage the Object Bank will become better as more



Figure 5.6: Diverse views of rowing boats in different images. Images are randomly selected from the UIUC sports dataset.

accurate object detection algorithms are developed. In the ideal case, if we use a perfect object detector, we can achieve close to perfect classification performance in semantically separable dataset. An interesting observation is that if we use the names of objects appear in each image from the UIUC sports dataset as the feature, we can achieve 100% in classification accuracy by using a simple linear SVM classifier.

5.4.2 Role of View Points

The view points of objects in different images could vary dramatically. For example, in Fig. 5.6, rowing boats appear in different view points depending on the sceneries the photographers want to snap.

In order to capture this property, we train the object detectors by using object images with different view points. To show our design rationale on view points, we apply Object Bank representation generated from objects with front view, side view and the combination of both to a scene classification task.

Demonstrated in the comparison (Left panel of Fig. 5.7), front view contributes more in classification experiments on both datasets than side view of objects. Combining the two views further boosts classification performance on MIT Indoor dataset⁴, which shows that combining multiple view points is useful.

To verify our assumption and further investigate the effectiveness of combining multiple view points, we conduct a control experiment by training and testing on different views of one specific object. We select “rowing boat” as an example since it

⁴The difference is not significant (1%). One possible reason for this is that there is not much view variance in most of the object detector training data from ImageNet. Majority of the training images are front shots of the objects.

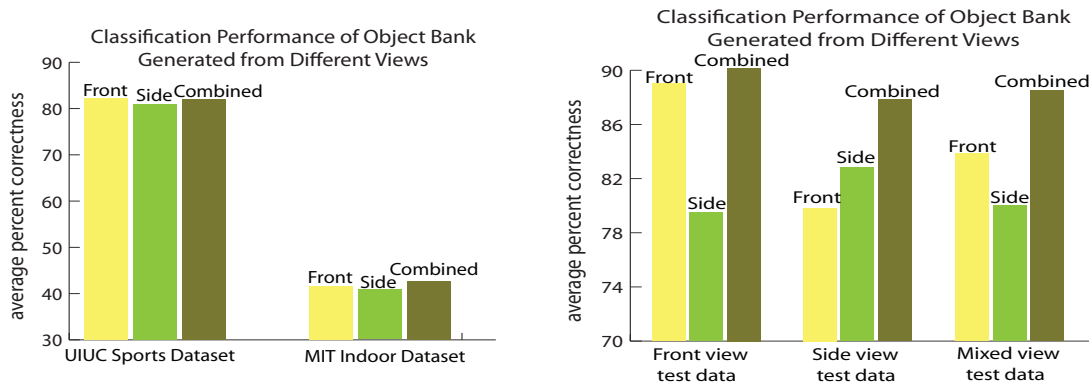


Figure 5.7: **Left:** Classification performance of Object Bank generated from detectors trained on images with different view points on the UIUC sports dataset and the MIT Indoor dataset. **Right:** Classification performance of Object Bank generated from different views on images with different view points.

has diverse view points.

As Fig. 5.7(Right) demonstrates, representation generated from only front view or side view performs reasonably well on test images with similar view points. Object Bank representation, by incorporating both views, significantly outperforms these two baselines on all three types of testing images with different view points.

5.4.3 Role of Scales

Object size in different images could vary dramatically, we therefore run object filter on different image scales to accurately capture this. In this experiment, we evaluate the importance of generating responses of objects at multiple scales in Object Bank. We compare classification performance by using Object Bank representation corresponding to each individual scale and the combination of multiple scales.

We observe that individual scales perform similarly to each other with the medium size scale consistently delivers the best result on both the UIUC sports and the MIT Indoor datasets. Our observation reflects that our object detector captures the medium size objects within these two datasets the best. This observation aligns well with the fact that the majority of objects in the ImageNet images have medium size. Each individual scale cannot capture all the variances but can already perform relatively good on this dataset. Same applies to the MIT Indoor dataset. We further

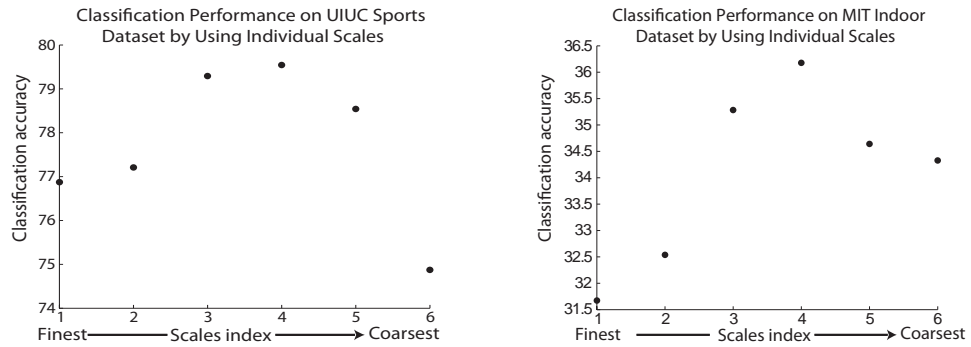


Figure 5.8: **Left:** Classification performance on the UIUC sports event dataset by using Object Bank representation corresponding to each single scale. **Right:** Classification performance on the MIT Indoor dataset by using Object Bank representation corresponding to each single scale. X axis is the index of the scale from fine to coarse. Y axis represents the average precision of a 8-way classification.

show an accumulative concatenation of scales.

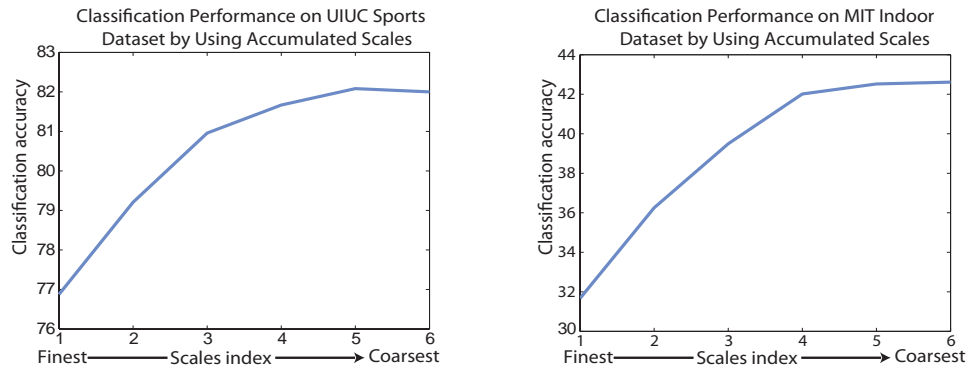


Figure 5.9: **Left:** Classification performance on the UIUC sports event dataset by using Object Bank representation corresponding to each single scale. **Right:** Classification performance on the MIT Indoor dataset by using Object Bank representation corresponding to each single scale. X axis is the index of the scale from fine to coarse. Y axis represents the average precision of a 8-way classification.

Fig. 5.9 shows that incorporating multiple scales is helpful since it captures the variance of object sizes in the datasets.

Objects could have significantly different sizes even they belong to the same object class. Here, we demonstrate that Object Bank is able to capture the scale variance. In this control experiment, we divide the images in the “ball” class into six scale groups based on the object size within the image. Same number of background images

are randomly selected from the ImageNet object images in the binary classification experiments. Representation generated from each individual scales and the combined ones are tested on a held out set of testing images with subgroups separated in a similar fashion. In addition, we simulate the real-world scenario by collecting a mixed test set of multiple object size images. In a similar manner, we generate the Object Bank representation based upon responses to combination of all scales.

We generate a mask whose transparency is proportional to the score in each grid in the confusion matrix. Our experiment shows that the diagonal of Fig. 5.10 is much brighter than the off diagonal ones, which indicates that Object Bank representation generated from each individual scale recognizes objects with similar size significantly better than the one generated from different size. In addition, the last row is much brighter than all the other grids, reflecting combination of all scales performs the best on different types of images. This again supports our design choice of incorporating multiple scales in Object Bank representation.

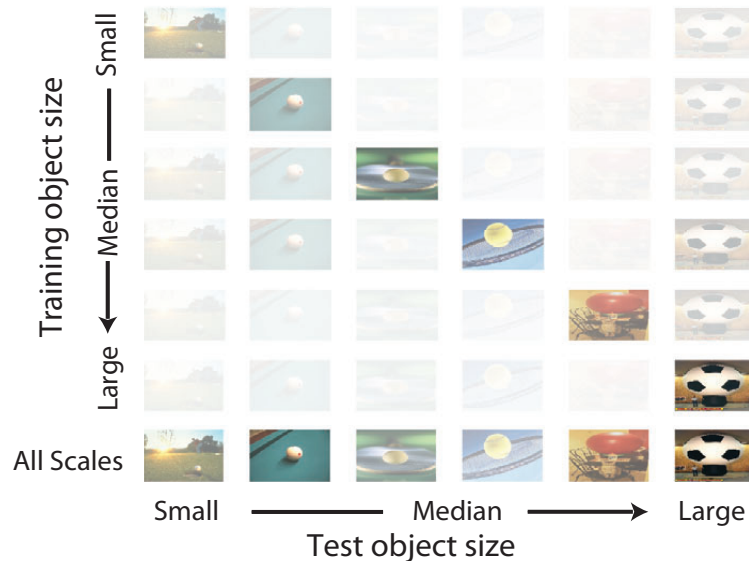


Figure 5.10: Binary classification experiment for each individual scale and the combination of them. Each grid is filled in with “ball” in different size. Each row represents a model trained on images with relatively similar scale (from small to large. Last row is the combination of all scales). Each column represents a test set with relatively similar scale. The more transparent the mask is, the better the classification accuracy is.

Index	Single UIUC	Pyramid UIUC	Single Indoor	Pyramid Indoor
Level 0	81.1%	-	36.2%	-
Level 1	81.0%	81.8%	42.0%	42.9%
Level 2	81.4%	82.0%	42.2%	42.6%

Table 5.2: Classification performance by using different spatial location structure.

5.4.4 Role of Spatial Location

Besides object semantic meaning, spatial locations of objects are critical for describing an image too. For example, “sky” always appears in the upper part of an image whereas “water” is always at the bottom. If the response of “sky” has higher values in the upper part of an image, it adds more evidence there is “sky” in the image. To capture the spatial properties of objects in an image, we apply the spatial pyramid structure on the response map. In this experiment, we analyze the effectiveness of the spatial pyramid structure.

From the experiment, we observe similar pattern as observed in [85] except for the good performance by using level 0 on the UIUC sports dataset. By using the maximum response from only level 0 as Object Bank representation (1/21 of the original dimension), we can achieve 81.1%. This reflects that in the UIUC sports semantic meaning alone is very discriminative. As long as we see a horse, no matter where it appears, it is able to differentiate polo scene from other scene types. On the other hand, adding spatial information does improve the classification performance with a small margin indicating the effectiveness of spatial location information of object.

Spatial location is critical in separating different indoor images. For example, computer room and office could both have computers, desks and chairs. But the number of instances and the spatial location arrangement of them could be quite different. Object Bank representation encoding spatial location is able to capture such difference and hence generates better performance in classification. Our classification experiment on the MIT Indoor dataset shows that by encoding spatial location, Object Bank representation significantly outperforms the one only contains semantic

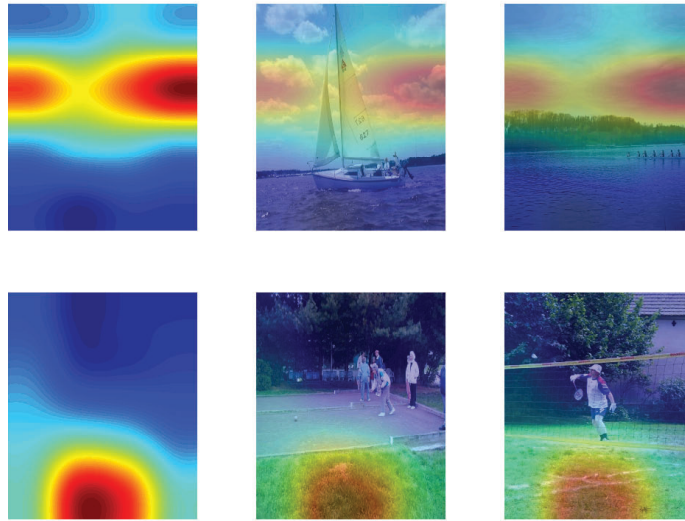


Figure 5.11: **Left:** Heat map of possible locations estimated from classification performance of Object Bank representation generated from different spatial locations. **Right:** Example images with the possible location map overlaid on the original image.

information.

To further demonstrate the effectiveness of the spatial location component in the Object Bank representation, we conduct a spatial location control experiment. In this experiment, we select an object that always appear at the top of the image, e.g. “cloud” and an object that always appear at the bottom of the image, e.g. “grass”. We use level 2 in the spatial pyramid structure as an example, each time we preserve one of the spatial location as the representation and perform classification based on the Object Bank feature extracted from it.

We observe that the Object Bank representation generated from regions with high performance are also the locations where the object frequently appears. For example, cloud usually appears in the upper half of a scene in the beach class whereas grass appear at the bottom.

5.4.5 Comparison of Different Pooling Methods

As described earlier, Object Bank representation is summarized from the responses of the image to different object filters. In order to obtain the geometric locations and

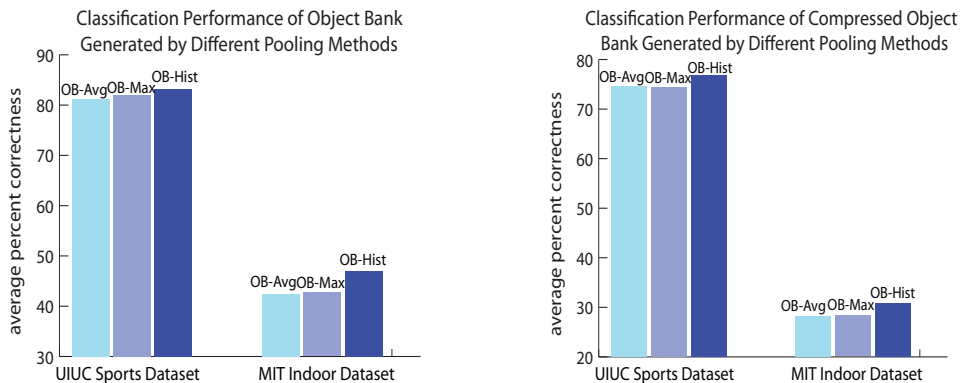


Figure 5.12: **Left:** Classification performance of different pooling methods. **Right:** Classification performance of PCA projected representations by using different pooling methods. Dimension is fixed to the minimum number (~ 100 dimensions) of principal components to preserve 99% of the unlabeled data variance of the three representations. Average and maximum response values within each spatial pyramid grid are extracted as the Object Bank feature in average pooling and max pooling respectively. We discretize values within each spatial pyramid to construct the histogram pooling representation.

semantic meaning of objects, we extract statistics of responses to object filters from different spatial locations by using pooling methods. The quality of pooling method influence the information that the Object Bank representation carries. In a similar vain, we analyze the effectiveness of different pooling methods. We fix other designing choices in our Object Bank and use the same classifier for different pooling methods.

One concern is that the richness of the representation could be attributed to the high dimension of features. To investigate this possibility, we compress the three representations to the same dimension by using PCA and perform classification on the compressed representation.

Fig. 5.12 shows that Object Bank representation generated from histogram pooling performs the best in classification even when it is compressed to the same dimension as the other two methods. It indeed carries more information that is more descriptive of the images.

5.4.6 Role of Objects

Earlier in this chapter, we have introduced that Object Bank is built upon image responses to a group of pre-trained object detectors. Object candidates are very

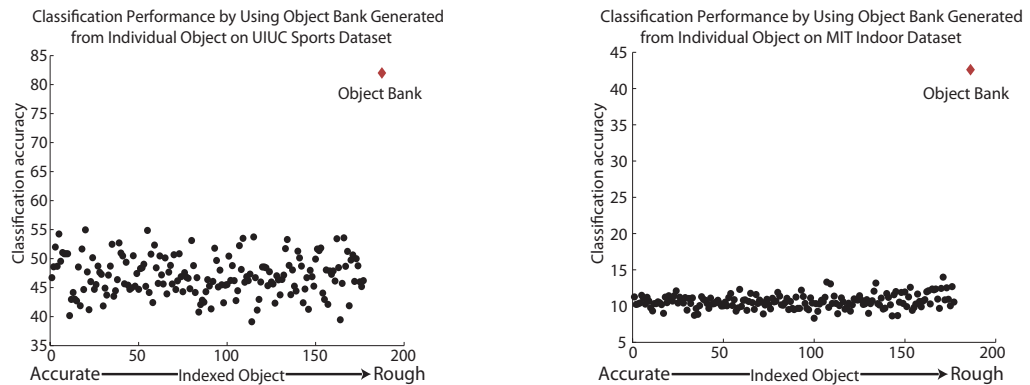


Figure 5.13: **Left:** Classification performance on the UIUC sports event dataset by using Object Bank representation corresponding to each single object. **Right:** Classification performance on the MIT Indoor dataset by using Object Bank representation corresponding to each single object. X axis is the index of the object sorted by using the detection performance on object datasets from ImageNet. Y axis represents the average precision of a 8-way classification.

critical component in designing the Object Bank representation. In this subsection, we analyze the effectiveness of different types of objects. Specifically, we are interested in Object Bank generated from two sources respectively: a generic pool of objects and a small group customized objects which are directly related to scene images.

Generic Object Bank

We first examine the generic Object Bank representation constructed from 177 most popular objects from ImageNet⁵. We begin with investigate how well each individual object is able to capture the essential information within images, evaluated by the classification performance of Object Bank representation generated from each individual object. In Fig. 5.13, each dot represents the classification performance of a specific object. The first observation is that the classification precisions across single objects are not necessarily correlated with the detection performance (objects are sorted by their detection performance from high to low). This can be attributed to the fact that the objects with good detections might not have semantic relationship with the scene types we test on. In general, the performance over a single object

⁵We evaluate the popularity of the objects based on the number of images available for downloading by using the object name as the query word in major search engines.

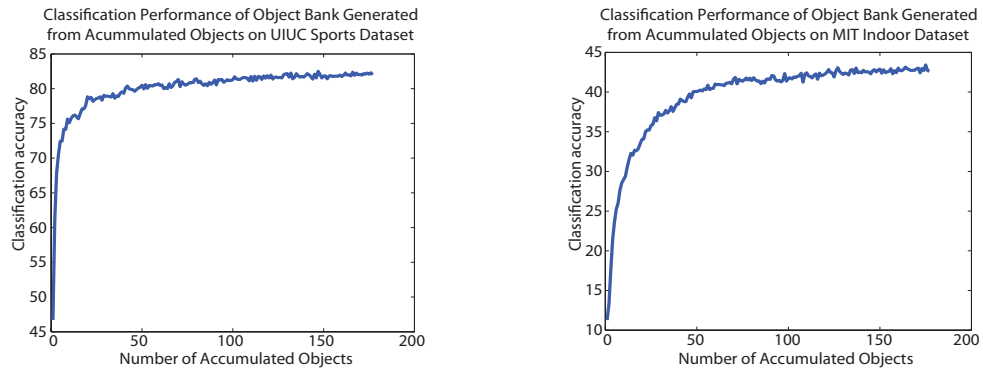


Figure 5.14: Classification performance on the UIUC sports event dataset and the MIT Indoor dataset by using Object Bank representation corresponding to accumulative object. X axis is the number of objects. Y axis represents the average precision of a 8-way classification.

falls in the 40%-60% range for the UIUC sports dataset, which indicates that the information captured by a single object is quite significant. However, it is still far from explaining away the information provided by combination of information from all objects, i.e., the full-dimensional Object Bank representation.

To investigate the effectiveness of using multiple object candidates in Object Bank, we vary the number of object candidates and test the resulting representation on scene classification. By plotting the average precision when a feature corresponding to a subsequent object is added one at a time in Fig. 5.17, we observe that the classification accuracy increases along with the increase of number of objects. We believe that future growth of Object Bank will lead to stronger representation power and more discriminative images models build on Object Bank.

How Much Semantic Meaning and Appearance Helps: Customized Object Bank

In the ideal case, if we know the identity of objects in each image, the classification performance on the UIUC sports event dataset is 100%. Models that accurately predict the semantic meaning of objects can serve as critical prior knowledge for describing an image. An important characteristics of Object Bank representation is that it encodes prior knowledge of objects. Here, we analyze the influence of prior knowledge especially the semantic meaning and appearance knowledge of objects

encoded in Object Bank representation generated by using a group of customized objects.

We first investigate the appearance models, i.e. our object filters, trained from both the ImageNet and UIUC training images. We show below comparison of object filters trained on two candidate objects, “sail boat” and “human”. We illustrate the models visualization comparison of these two objects.

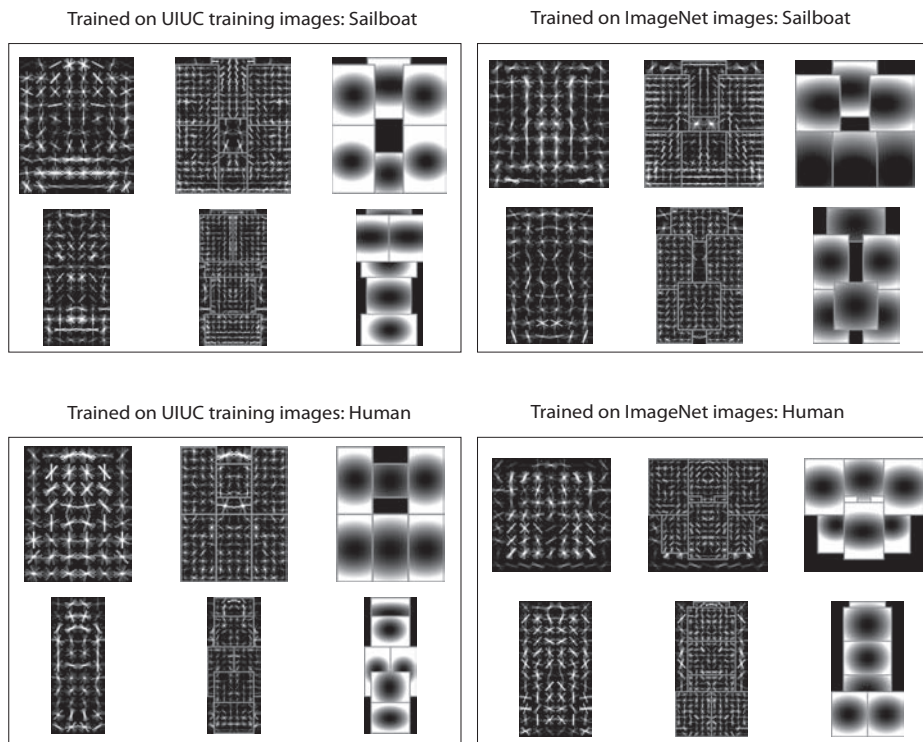


Figure 5.15: Model comparison of “sail boat” and “human” models trained on the UIUC training images (Left) and the ImageNet images (Right).

As shown in Fig. 5.15, models trained on the UIUC training images capture a clearer shape of the objects. The model quality is also reflected by their object detection performance on objects within the UIUC scene test images.

By clearly depicting the object appearance, models trained on the UIUC training images detects the objects in the object images in a held-out set accurately (Fig. 5.16).

In addition, we compare the image classification performance by using only these

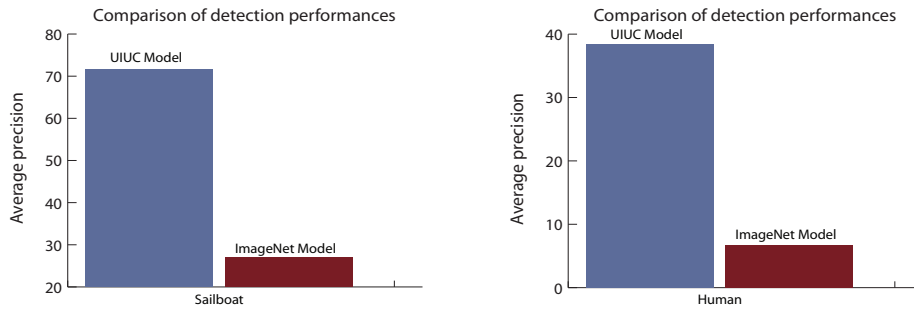


Figure 5.16: Detection performance comparison of models trained on the UIUC training images and the ImageNet images.

models each at a time.

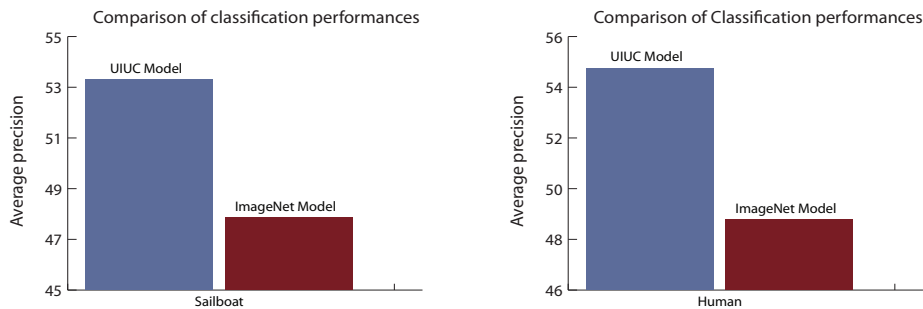


Figure 5.17: Classification performance on the UIUC sports event dataset by using Object Bank representation generated from sailboat and human models trained on the UIUC training images and the ImageNet images respectively. Y axis represents the average precision of a 8-way classification.

The customized Object Bank models captures the object property better, which leads to better detection accuracy and better classification performance. Individual customized object models exhibit lots of potential in generating more descriptive image representation, we further verify the potential of customized object models of all semantically related objects in the UIUC sports dataset.

We compare the overall classification performance by using all semantically related models, where we train the customized Object Bank filters by using 25 object candidates from the UIUC training images. We construct the customized Object Bank representation based on these filters, called UIUC-25. UIUC-25 carries knowledge of object appearance from the UIUC training images whereas the generic Object Bank representation (ImageNet-177) encodes prior knowledge of object appearance from

the ImageNet object training images. We compare UIUC-25 to ImageNet-177, Object Bank representation constructed from a subset of randomly selected 25 objects (ImageNet-25) as well as the ‘pseudo’ Object Bank representation generated from a set of synthesized models neglecting the semantic meaning of objects.

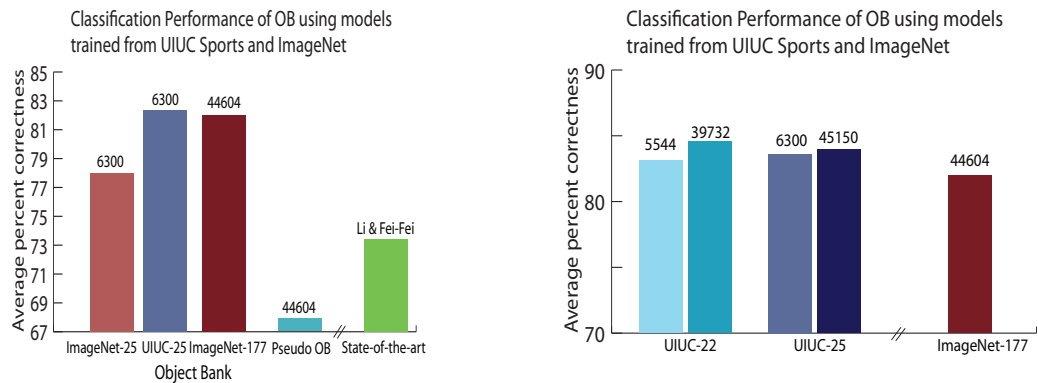


Figure 5.18: **Left:** Classification performance on the UIUC sports event dataset by using UIUC-25 (customized Object Bank), ImageNet-177 (generic Object Bank), ImageNet-25 (25 objects randomly selected from the ImageNet object candidates), randomly generated filters (Pseudo Object Bank) and state-of-the-art algorithm on the UIUC sports dataset. The blue bar in the last panel is the performance of “pseudo” Object Bank representation extracted from the same number of “pseudo” object detectors. The values of the parameters in these “pseudo” detectors are generated without altering the original detector structures. In the case of linear classifier, the weights of the classifier are randomly generated from a Gaussian distribution instead of learned. “Pseudo” Object Bank is then extracted with exactly the same setting as Object Bank. **Right:** Classification performance on the UIUC sports event dataset by using different appearance models: UIUC-22, UIUC-25, and ImageNet-177. Numbers at the top of each bar indicates the corresponding feature dimension.

In Fig. 5.18(Left), while the generic Object Bank (ImageNet-177 and ImageNet-25) has very good generalizability, the customized Object Bank consistently delivers much better result. It not only outperforms Object Bank generated from equivalent number of object candidates in ImageNet, but also outperforms full dimensional Object Bank. It is worth noticing that the dimension of full dimensional Object Bank is over 7 times that of the customized Object Bank representation. Comparing to [91], which requires labels of each pixel within an image in training, customized Object Bank outperforms it significantly without additional information required. In fact, obtaining bounding box costs less labor than obtaining object contour required in [91]. We further decompose the spatial structure and semantic meaning encoded in Object Bank by using a “pseudo” Object Bank without semantic meaning. The significant

improvement of Object Bank in classification performance over the “pseudo OB” is largely attributed to the effectiveness of using object detectors trained from image. On the other hand, “pseudo” Object Bank performs reasonably well indicating that it does capture consistent structures in the images. To demonstrate the capabilities of different models in encoding the structural information in images, we show the models and their corresponding response maps in Fig.5.19. As we can observe from Fig.5.19,

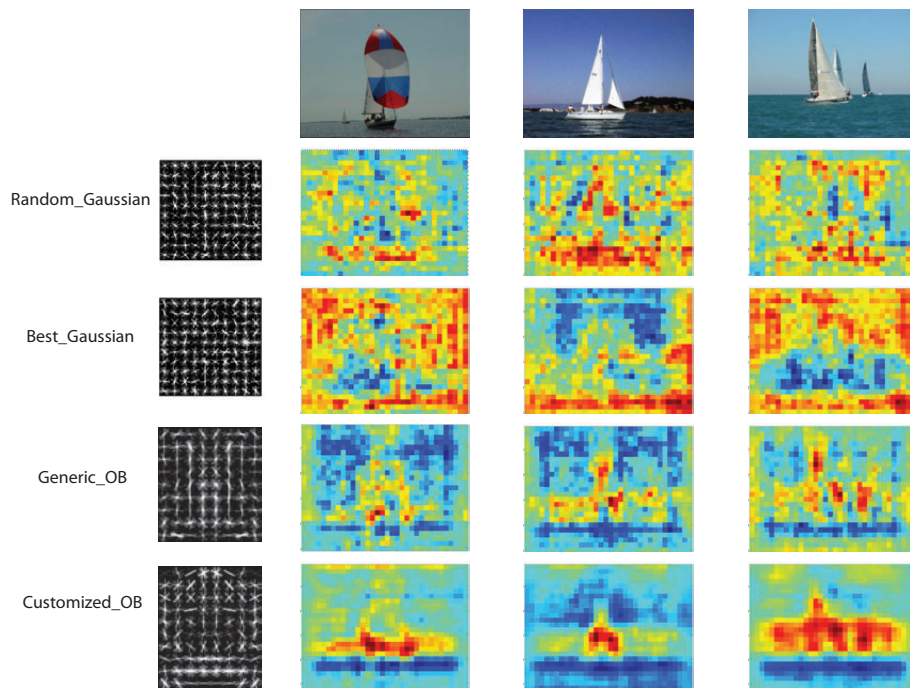


Figure 5.19: Comparison of different models and response maps generated. **Column 1:** Models visualized by using learned weights of histogram of oriented gradients. Here, “Random Gaussian” represents a randomly generated model by using random numbers sampled from a Gaussian distribution. “Best Gaussian” refers to the randomly generated model which performs best in classifying images containing “sailboat” from other images in the UIUC sports dataset. **Column 2-4:** Original images and the corresponding response maps. Each row corresponds to the response maps of images in the first row generated by the model showed in the first column.

while the Object Bank models are capable of generating relatively accurate response maps corresponding to the “sailboat” locations in the images, randomly generated “pseudo” Object Banks does reflect consistency in generating the response maps. It worths noticing that the best performed random model in “sailboat” classification generates response maps which have high responses in every pixels in the images

except the “sailboat” regions. Among the response maps, those generated by customized Object Bank locate the “sailboat” most accurately. The significantly good performance of the customized Object Bank can be easily explained: it is trained on the UIUC scene images which generates object filters that are more semantically related and accurate in appearance modeling.

An important question is that given the objects are semantically related, would better appearance models improve the quality of Object Bank representation? We investigate two possibilities for improving appearance models. A few object candidates in the UIUC sports dataset has only a couple of training images, which leads to deteriorated detection ability. Our first option to improve the appearance models is to evaluate the detection performance of object candidates and filter out three models with low detection performance. We call the representation generated UIUC-22. We can further improve our appearance models by increasing the number of scales, i.e., the possibility of accurately capturing more object sizes. Specifically, we increase the number of scales for UIUC-22 and UIUC-25 from 6 to 43, which makes the final dimension of both enriched representations approximately the same as the original Object Bank representation. We explore these two aspects as an example case study.

With a small number of semantically related models trained from the UIUC training images, the classification is more accurate than that of all 177 object candidates in the original Object Bank representation. In addition, increasing the number of scales lead to richer appearance model which generates even better representation for classification.

5.4.7 Relationship of Objects and Scenes Discovered by OB

Object Bank is built upon the idea of using objects to describe images. It encodes rich semantic and spatial structural information, from which we can discover interesting relationship of the images that the Object Bank is extracted from. Intuitively, objects are closely related to scenes they often appear. In this experiment, we aim to examine the effectiveness of individual objects in different scene classes towards discovering the interesting relationship between objects and scene types from the Object Bank

representation. To dissect the relationship of each individual objects to the scene types, we perform classification on the UIUC sports dataset based upon Object Bank feature dimensions corresponding to individual object. A relationship map (Fig. 5.20) is generated based on how accurate the individual object captures the discriminative information in a scene.



Figure 5.20: Most related scene type for each object. Rows are objects and column represent scene types. Classification scores of individual objects are used as the measurement of relationship between objects and scene types. The higher the classification accuracy, the more transparent the mask is in the intersecting grid of the object and scene type.

Fig. 5.20 shows that objects that are “representative” for each scene can be discovered by our simple method based upon Object Bank, indicating that semantic

related objects indeed are important to scene classification. For example, “basketball frame”, “net” and “window” are objects with very high classification accuracy in the “badminton” scene class whereas “horse” has the highest classification score in “polo” class.

5.4.8 Relationship of Different Objects Discovered by OB

Objects related to each other often exhibit similarity in appearance or similar pattern in classifying scene types. Such relationship can again be reflected by our Object Bank representation. In a similar manner, we try to discover the relationship among objects based on their classification behavior in this experiment. We use prediction probabilities of each scene class as the feature of the objects and build the correlation map (Fig. 5.21) based upon the distance between different objects.

We observe that the objects that are intuitively related to each other are also those that have strong correlations in Fig. 5.21. For example, “row boat” always co-occurs with “water”, “oar” and “cloud”. It is connected to “sailboat” due to their similar connection with “water” and similarity structure in appearance. This suggests that essential information from each image has been successfully extracted and preserved in Object Bank, from which we can reconstruct the appearance similarity and co-occurrence information. On the other hand, we examine the similarity of the response maps generated by convolving each image with the trained object detectors, which is dramatically different than what we discovered in Fig. 5.21 and does not have a clear pattern. This is largely attributed to the fact that very few objects in the UIUC sports dataset share appearance similarity. Our observation in this experiment leads to a strong indication that the co-occurrence of objects is the main factor to relate different objects to each other here.

5.5 Analysis: Guideline of Using Object Bank

In this analysis, we analyze the robustness of Object Bank to different classification methods and provide simple guidelines for efficient computation.

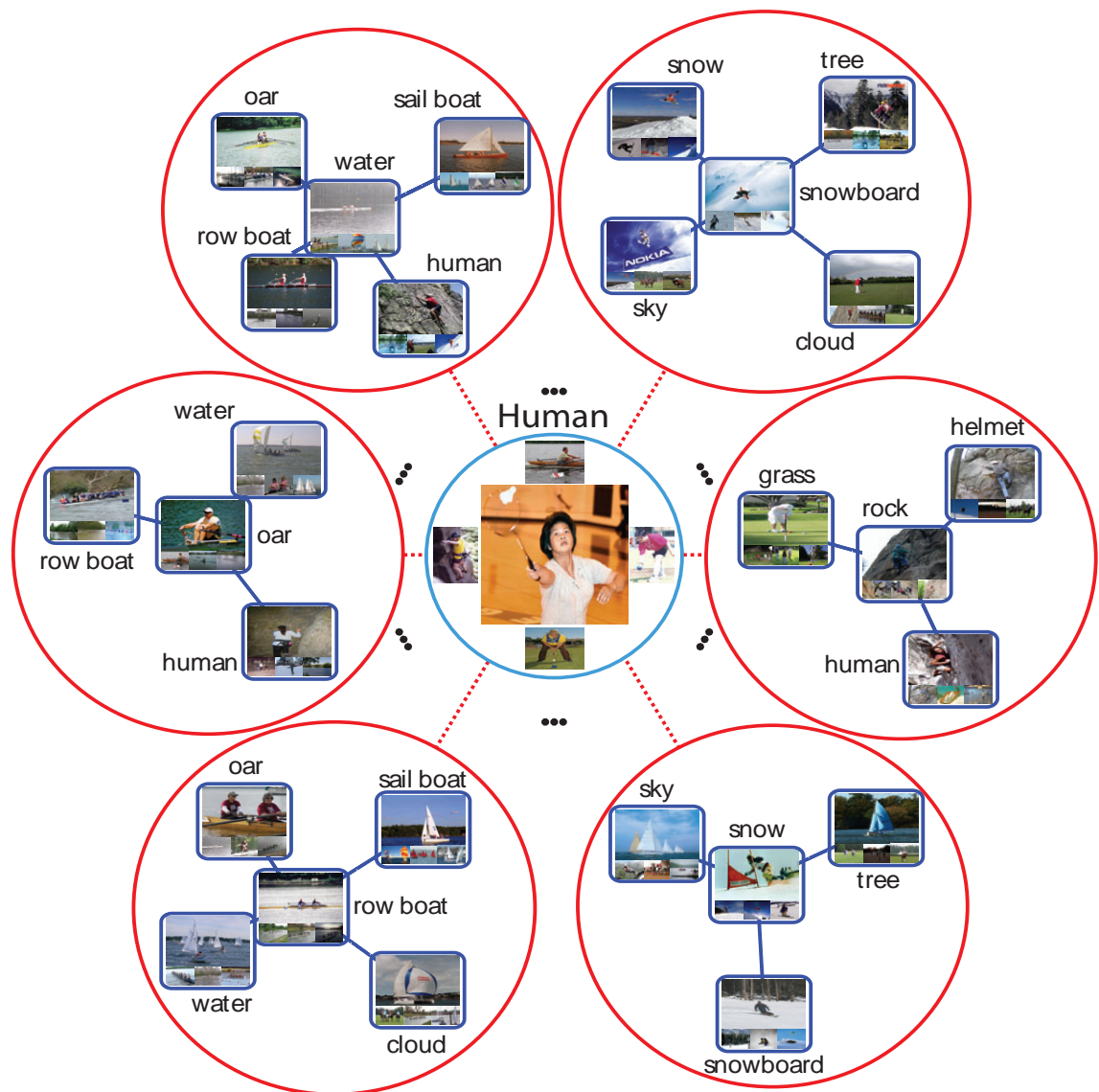


Figure 5.21: Relationship of objects. Classification scores of individual objects are used as the feature to measure the distance among objects.

Method	UIUC Sports	MIT Indoor
k -Nearest Neighbor, Euclidean metric	67.3% ($k = 1$)	25.4% ($k = 29$)
k -Nearest Neighbor, Gaussian Kernel	70.4% ($k = 17$)	28.1% ($k = 20$)
L2 regularized LR	80.2%	45.5%
L2 regularized L2 loss SVM	82.3%	46.6%
L2 regularized L1 loss SVM	82.3%	46.6%
L1 regularized L2 loss SVM	81.5%	42.1%
L1 regularized LR	82.0%	42.6%

Table 5.3: Classification performance of different methods.

5.5.1 Robustness to different classification models

We have established a clear advantage of using Object Bank representation for the image classification task. We now examine whether Object Bank features still maintains its advantage when we apply different off-the-shelf classification methods to it. In Table.7.2, we examine the effectiveness of different classification methods ranging from very simple nearest neighbor algorithm to more sophisticated regularized SVM and logistic regression models.

With this very descriptive image representation, even simple method such as k -nearest neighbor can achieve comparable performance to state-of-the-art methods with more complicate models. More sophisticated models can further boost the scene classification performance. We envisage that models customized to Object Bank can further maximize the potential of it on various high level visual recognition tasks.

5.5.2 Dimension Reduction by Using PCA

Object Bank representation is a robust representation with high dimension, which can be easily compressed to a low dimensional representation for efficiency. Here, we demonstrated the dimension reduction by using a simple projection method, i.e. PCA.

As shown in Fig. 5.22, simple dimension reduction method such as PCA can compress the Object Bank representation to much lower dimensions and still perform comparably well to the original Object Bank. The computation time also decreased dramatically along with the image representation dimensions. In addition, it performs

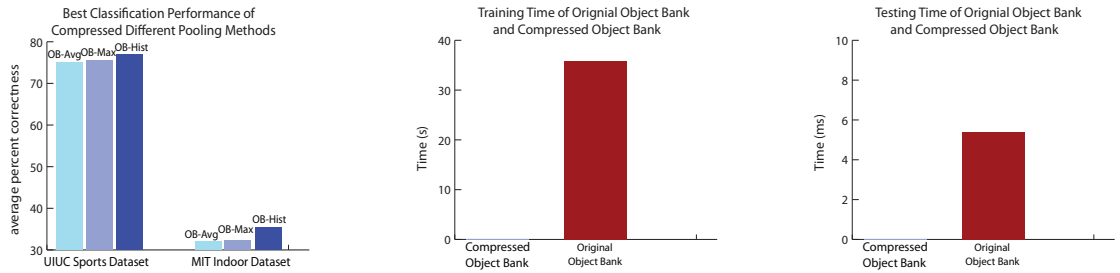


Figure 5.22: **Left:** Best classification performance of projected representations by using different pooling methods. All dimensions are below 150. **Middle:** Training time comparison of the original Object Bank and the compressed Object Bank using OB-Max as an example. **Right:** Testing time comparison of the original Object Bank and the compressed Object Bank for each image.

much better than low level features with the same dimension (SIFT) or significantly higher feature dimensions (GIST and SPM). Our observation reflects that Object Bank representation is a high dimensional representation with redundancy, which can be easily compressed and efficiently applied to high level tasks.

5.5.3 Dimension Reduction by Combining Different Views

When we design Object Bank, we incorporate multiple views of objects for more accurate description. Since the view points are complementary to each other, we show that simple methods for combining different views and reducing the dimensions of the Object Bank representation can be effective.

An object within an image can either be front view or side view, which is reflected by the statistics of the values in the response map. Here we show that the high average value of responses for one view of an object in an image is a strong indicator of an object appears in that image. Therefore, the classification performance is even higher than concatenating both views, where one of them might have low response indicating the object does not present in the image. Selecting the view point by using maximum variance is effective too. However, selection based on maximum response value of different view points deteriorates the classification performance due to the sensitivity of this selection method.

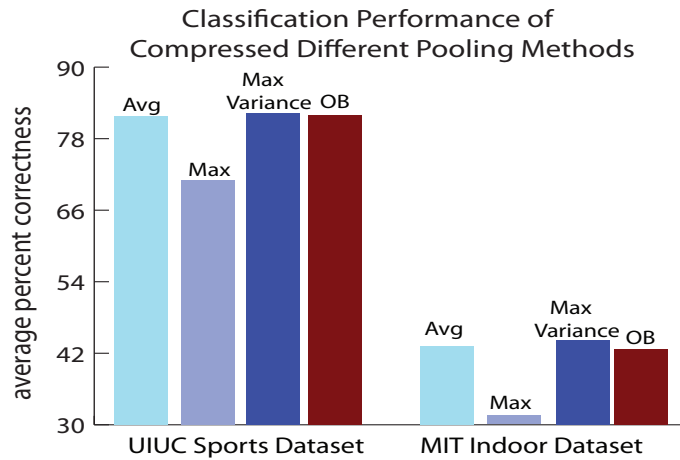


Figure 5.23: Classification performance of different pooling methods for dimension reduction. We select feature dimensions corresponding to the view point with higher average value, maximum value and maximum variance respectively for classification. This corresponds to 1/2 dimension reduction.

5.6 Discussion

The Object Bank representation described in this Chapter is a novel high level image representation that can effectively bridge the “semantic gap” between low level image representation and the high level visual recognition tasks. It is particularly useful given the semantic and spatial knowledge encoded in the representation. The semantic information is obtained by running object detectors over multiple scales of images to capture the possibility of objects appear in the images. A spatial pyramid structure is applied to the response map representing the possibility of objects in an image to summarize the spatial statistics of objects. We analyze in depth the effectiveness of each component in our Object Bank representation and provide useful guidelines for usage of Object Bank.

Chapter 6

Semantic Feature Sparsification of Object Bank

While the Object Bank representation offers a rich, high-level description of images, a key technical challenge due to this representation is the “curse of dimensionality”, which is severe because of the size (i.e., number of objects) of the Object Bank and the dimensionality of the response vector for each object. Typically, for a modest sized picture, even hundreds of object detectors would result into a representation of tens of thousands of dimensions. Therefore to achieve robust predictor on practical dataset with typically only dozens or a couple of hundreds of instances per class, structural risk minimization via appropriate regularization of the predictive model is essential.

In this chapter, we propose a regularized logistic regression method, akin to the group lasso approach for structured sparsity, to explore both *feature sparsity* and *object sparsity* in the Object Bank representation for learning and classifying complex scenes. We show that by using this high-level image representation and a simple sparse coding regularization, our algorithm not only achieves superior image classification results in a number of challenging scene datasets, but also can discover semantically meaningful descriptions of the learned scene classes.

6.1 Scene Classification and Feature/Object Compression via Structured Regularized Learning

We envisage that with the avalanche of annotated objects on the web, the number of object detectors in our Object Bank will increase quickly from hundreds to thousands or even millions, offering increasingly rich signatures for each images based on the identity, location, and scale of the object-based content of the scene. However, from a learning point of view, it also poses a challenge on how to train predictive models built on such high-dimensional representation with limited number of examples. We argue that, with an “overcomplete” Object Bank representation, it is possible to compress ultra-high dimensional image vector without losing semantic saliency. We refer this semantic-preserving compression as *content-based compression* to contrast the conventional information-theoretic compression that aims at lossless reconstruction of the data.

In this chapter, we intend to explore the power of Object Bank representation in the context of scene classification, and we are also interested in discovering meaningful (possibly small subset of) dimensions during regularized learning for different classes of scenes. For simplicity, here we present our model in the context of linear binary classifier in a 1-versus-all classification scheme for K classes. Generalization to a multiway softmax classifier is slightly more involved under structured regularization and thus deferred to future work. Let $\mathbf{X} = [\mathbf{x}_1^T; \mathbf{x}_2^T; \dots; \mathbf{x}_N^T] \in \mathbb{R}^{N \times J}$, an $N \times p$ matrix, represent the design built on the J -dimensional Object Bank representation of N images; and let $\mathbf{Y} = (y_1, \dots, y_N) \in \{0, 1\}^N$ denote the binary classification labels of N samples. A *linear classifier* is a function $h_\beta : \mathbb{R}^J \rightarrow \{0, 1\}$ defined as $h_\beta(x) \triangleq \arg \max_{y \in \{0, 1\}} \mathbf{x}\beta$, where $\beta = (\beta_1, \dots, \beta_J) \in \mathbb{R}^J$ is a vector of *parameters* to be estimated. This leads to the following learning problem $\min_{\beta \in \mathbb{R}^J} \lambda R(\beta) + \frac{1}{m} \sum_{i=1}^m L(\beta; x_i, y_i)$, where $L(\beta; x, y)$ is some non-negative, convex loss, $R(\beta)$ is a *regularizer* that avoids overfitting, and $\lambda \in \mathbb{R}$ is the regularization coefficient, whose value can be determined by cross validation.

A common choice of L is the *Log loss*, $L = \log(1/P(y_i|\mathbf{x}_i, \beta))$, where $P(y_i|\mathbf{x}_i, \beta)$

is the *logistic* function $P(y|\mathbf{x}, \beta) = \frac{1}{Z} \exp(\frac{1}{2}y(\mathbf{x} \cdot \beta))$. This leads to the popular logistic regression (LR) classifier¹. Structural risk minimization schemes over LR via various forms of regularizations have been widely studied and understood in the literature. In particular, recent asymptotic analysis of the ℓ_1 norm and ℓ_1/ℓ_2 mixed norm regularized LR proved that under certain conditions the estimated *sparse* coefficient vector β enjoys a property called *sparsistency* [137], suggesting their applicability for meaningful variable selection in high-dimensional feature space. In this chapter, we employ an LR classifier for our scene classification problem. And we investigate content-based compression of the high-dimensional Object Bank representation that exploits raw feature-, object-, and (feature+object)-sparsity, respectively, using LR with appropriate regularization.

Feature sparsity via ℓ_1 regularized LR (LR1) By letting $R(\beta) \triangleq \|\beta\|_1 = \sum_{j=1}^J |\beta_j|$, we obtain an estimator of β that is sparse. The shrinkage function on β is applied indistinguishably to all dimensions in the Object Bank representation, and it does not have a mechanism to incorporate any potential coupling of multiple features that are possibly synergistic, e.g., features induced by the same object detector. We call such a sparsity pattern *feature sparsity*, and denote the resultant coefficient estimator by β^F .

Object sparsity via ℓ_1/ℓ_2 (group) regularized LR (LRG) Recently, a mixed-norm (e.g., ℓ_1/ℓ_2) regularization has been used for recovery of joint sparsity across input dimensions. By letting $R(\beta) \triangleq \|\beta\|_{1,2} = \sum_{j=1}^J \|\beta^j\|_2$, where β^j is the j -th group (i.e., features grouped by an object j), and $\|\cdot\|_2$ is the vector ℓ_2 -norm, we set the feature group to be corresponding to that of all features induced by the same object in the OB. This shrinkage tends to encourage features in the same group to be jointly zero. Therefore, the sparsity is now imposed on object level, rather than merely on raw feature level. Such *structured sparsity* is often desired because it is expected to generate semantically more meaningful lossless compression, that is, out

¹We choose not to use the popular SVM which correspond to L being a *hinge* loss and $R(\beta)$ being a ℓ_2 -regularizer, because under SVM, content-based compression via structured regularization is much harder.

of all the objects in the OB, only a few are needed to represent any given natural image. We call such a sparsity pattern *object sparsity*, and denote the resultant coefficient estimator by β^O .

Joint object/feature sparsity via $\ell_1/\ell_2 + \ell_1$ (sparse group) regularized LR (LRG1) The group-regularized LR does not, however, yield sparsity within a group (object) for those groups with non-zero total weights. That is, if a group of parameters is non-zero, they will all be non-zero. Translating to the Object Bank representation, this means there is no scale or spatial location selection for an object. To remedy this, we proposed a composite regularizer, $R(\boldsymbol{\beta}) \triangleq \lambda_1 \|\boldsymbol{\beta}\|_{1,2} + \lambda_2 \|\boldsymbol{\beta}\|_1$, which conjoin the sparsification effects of both shrinkage functions, and yields sparsity at both the group and individual feature levels. This regularizer necessitates determination of two regularization parameters λ_1 and λ_2 , and therefore is more difficult to optimize. Furthermore, although the optimization problem for $\ell_1/\ell_2 + \ell_1$ regularized LR is convex, the non-smooth penalty function makes the optimization highly nontrivial. We derive a coordinate descent algorithm for solving this problem. To conclude, we call the sparse group shrinkage pattern *object/feature sparsity*, and denote the resultant coefficient estimator by β^{OF} .

6.2 Experiments and Results

We examine the behaviors of different structural risk minimization schemes over LR on the Object Bank representation, focusing on their classification performance and semantic interpretability. Specifically, we experimented ℓ_1 regularized LR (LR1), ℓ_1/ℓ_2 regularized LR (LRG) and $\ell_1/\ell_2 + \ell_1$ regularized LR (LRG1).

6.2.1 Semantic Feature Sparsification Over OB

In this subsection, we systematically investigate semantic feature sparsification of the Object Bank representation. We focus on the practical issues directly relevant to the effectiveness of Object Bank representation and quality of feature sparsification,

and study the following four aspects of the scene classifier: 1) robustness, 2) feasibility of lossless content-based compression, 3) profitability over growing OB, and 4) interpretability of predictive features.

Robustness with respect to training sample size

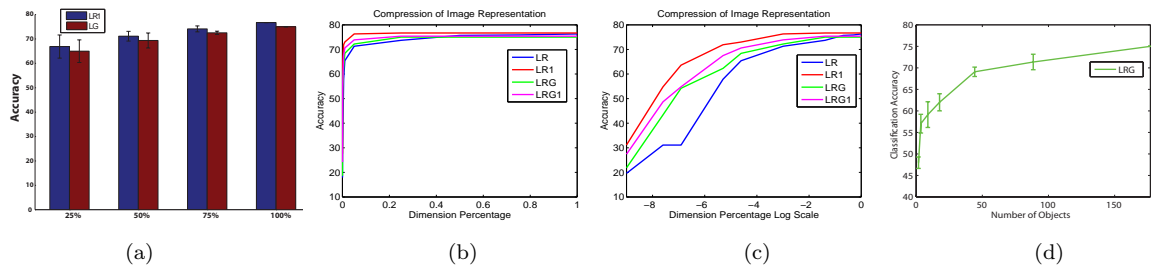


Figure 6.1: (a) Classification performance (and s.t.d.) w.r.t number of training images. Each pair represents performances of LR1 and LRG respectively. X-axis is the ratio of the training images over the full training dataset (70 images/class). (b) Classification performance w.r.t feature dimension. X-axis is the size of compressed feature dimension, represented as the ratio of the compressed feature dimension over the full Object Bank representation dimension (44604). (c) Same as (b), represented in Log Scale to contrast the performances of different algorithms. (d) Classification performance w.r.t number of object filters. X-axis is the number of object filters. 3 rounds of randomized sampling is performed to choose the object filters from all the object detectors.

The intrinsic high-dimensionness of the Object Bank representation raises a legitimate concern on its demand on training sample size. We investigate the robustness of the logistic regression classifier built on features selected by LR1 and LRG in this experiment. We train LR1 and LRG on the UIUC-Sports dataset by using multiple sizes of training examples, ranging from 25%, 50%, 75% to 100% of the full training data.

As shown in Fig. 6.1(a), we observe only moderate drop of performance when the number of training samples decreases from 100% to 25% of the training examples, suggesting that the Object Bank representation is a rich representation where discriminating information residing in a lower dimensional “informative” feature space, which are likely to be retained during feature sparsification, and thereby ensuring robustness under small training data. We explore this issue further in the next experiment.

Near losslessness of content-based compression via regularized learning

We believe that the Object Bank can offer an *over complete* representation of any natural image. Therefore, there is great room for possibly (near) lossless content-based compression of the image features into a much lower-dimensional, but equally discriminative subspace where key semantic information of the images are preserved, and the quality of inference on images such as scene classification are not compromised significantly. Such compression can be attractive in reducing representation cost of query image, and improving the speed of query inference.

In this experiment, we use the classification performance as a measurement to show how different regularization schemes over LR can preserve the discriminative power when we apply the content-based compression over Object Bank representation. For LR1, LRG and LRG1, cross-validation is used to decide the best regularization parameters, based on which image features can be ranked according to the magnitudes of the estimated regularization coefficients β in the regression function. To compare the extend of information loss as a function of different number of features being retained in the classifier, we re-train a LR classifier using features from the top $x\%$ percentile of the rank list, where x is a compression scale ranging from 0.05% to 100%. One might think that LR itself when fitted on full input dimensional can also produce a rank list of features for subsequent selection. But it is known that LR does not perform explicit feature sparsification, and can not zero-out irrelevant features as do LR1, LRG and LRG1, and the ordering from LR does not necessarily reflect true importance of features. For comparison purpose, we also include results from the LR-ranked features, as can be seen in Fig. 6.1(b,c), indeed its performance drops faster than all the regularization methods.

In Fig. 6.1 (b), we observe that the classification accuracy drops very slowly as the number of selected features decreases. By excluding 75% feature dimensions, classification performance of each algorithm decreases less than 3%. One point to notice here is that, the non-zero entries only appear in dimensions corresponding to no more than 45 objects for LRG at this point. Even more surprisingly, LR1 and LRG preserve accuracies above 70% when 99% of the feature dimensions are excluded.

Fig. 6.1 (c) shows more detailed information in the low feature dimension range,

which corresponds to a high compression ratio. We observe that algorithms imposing sparsity in features (LR1, LRG, and LRG1) outperform unregularized algorithm (LR) with a larger margin when the compression ratio becomes higher. This reflects that the sparsity learning algorithms are capable of learning the much lower-dimensional, but highly discriminative subspace.

Profitability over growing OB

We envisage the Object Bank will grow rapidly and constantly as more and more labeled web images become available. This will naturally lead to increasingly richer and higher-dimensional representation of images. We ask, are image inference tasks such as scene classification going to benefit from this trend?

As group regularized LR imposes sparsity on object level, we choose to use it to investigate how the number of objects will affect the discriminative power of Object Bank representation. To simulate what happens when the size of Object Bank grows, we randomly sample subsets of object detectors at 1%, 5%, 10%, 25%, 50% and 75% of total number of objects for multiple rounds. As in Fig. 6.1(d), the classification performance of LRG continuously increases when more objects are incorporated in the Object Bank representation. We conjecture that this is due to the accumulation of discriminative object features, and we believe that future growth of Object Bank will lead to stronger representation power and discriminability of images models build on OB.

6.2.2 Interpretability of the compressed representation

Intuitively, a few key objects can discriminate a scene class from another. In this experiment, we aim to discover the *object sparsity* and investigate its interpretability. Again, we use group regularized LR (LRG) since the sparsity is imposed on object level and hence generates a more semantically meaningful compression.

We show in Fig. 6.2 the object-wise coefficients of the compression results for 4 sample scene classes. The object weight is obtained by accumulating the coefficient of β^O from the feature dimensions of each object (at different scales and spatial locations)

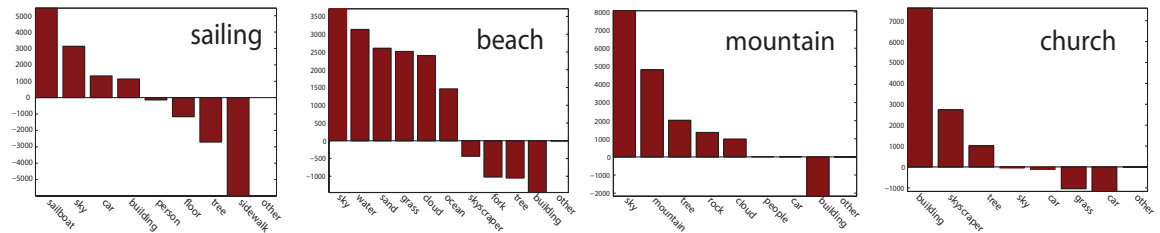


Figure 6.2: Object-wise coefficients given scene class. Selected objects correspond to non-zero β values learned by LRG.

learned by LRG. Objects with all zero coefficients in the resultant coefficient estimator are not displayed. Fig. 6.2 shows that objects that are “representative” for each scene are retained by LRG. For example, “sailboat”, “boat”, and “sky” are objects with very high weight in the “sailing” scene class. This suggests that the representation compression via LRG is virtually based upon the image content and is semantically meaningful; therefore, it is nearly “semantically lossless”.

Knowing the important objects learned by the compression algorithm, we further investigate the discriminative dimensions within the object level. We use LRG1 to examine the learned weights within an object. In Chapter 5, we introduce that each feature dimension in the Object Bank representation is directly related to a specific scale, geometric location and object identity. Hence, the weights in β^{OF} should reflect the importance of an object at certain scale and location for recognizing a scene. To verify the hypothesis, we examine importance of objects at different scales by summing up the weights at all related spatial locations and pyramid resolutions within the scale. We show one representative object in a scene and visualize the feature patterns within the object group. As it is shown in Fig. 6.3(Top), LRG1 has achieved joint object/feature sparsification by zero-out less relevant scales, thus only the most discriminative scales are retained. To analyze how β^{OF} reflects the geometric location, we further project the learned coefficient back to the image space by reversing the Object Bank representation extraction procedure. In Fig. 6.3(Middle), we observe that the regions with high intensities are also the locations where the object frequently appears. For example, cloud usually appears in the upper half of a scene in the beach class.

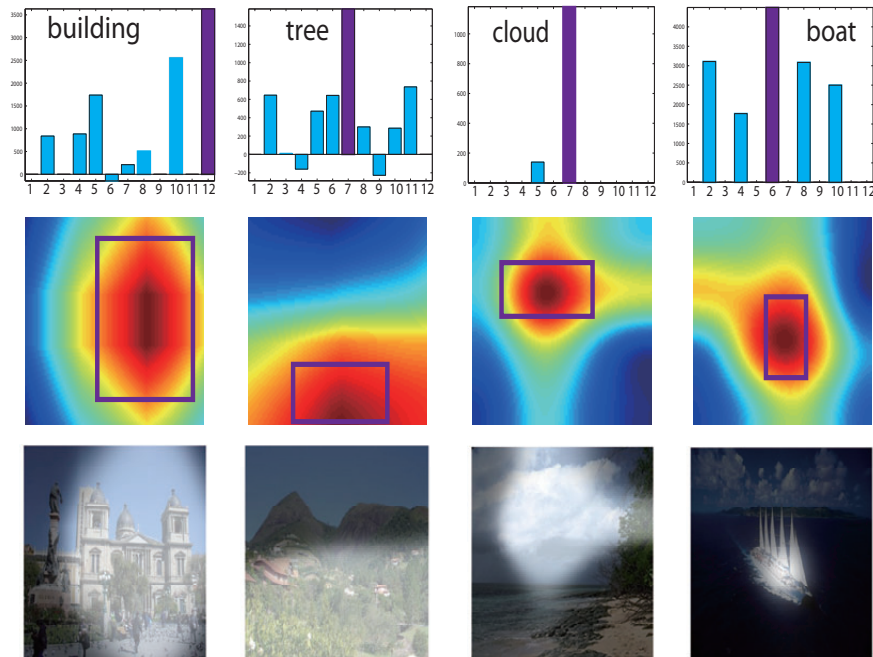


Figure 6.3: Illustration of the learned β^{OF} by LRG1 within an object group. Columns from left to right correspond to “building” in “church” scene, “tree” in “mountain”, “cloud” in “beach”, and “boat” in “sailing”. **Top Row:** weights of Object Bank dimensions corresponding to different scales, from small to large. The weight of a scale is obtained by summing up the weights of all features corresponding to this scale in β^{OF} . **Middle:** Heat map of feature weights in image space at the scale with the highest weight (purple bars above). We project the learned feature weights back to the image by reverting the Object Bank representation extraction procedure. The purple bounding box shows the size of the object filter at this scale, centered at the peak of the heat map. **Bottom:** example scene images masked by the feature weights in image space (at the highest weighted scale), highlighting the most relevant object dimension.

6.3 Discussion

As we try to tackle higher level visual recognition problems, we show that more semantic level image representation such as the Object Bank can capture important information in a picture without evoking highly elaborate statistical models to build up the features and concepts from pixels or low-level features. In this chapter, we apply regularized logistic regression schemes on the high dimensional Object Bank representation, and achieve nearly lossless semantic-preserving compression.

While the supervised regularized logistic regression method is effective in selecting

the essential components in OB, the supervision required is a big obstacle for scalability of this method. In the following chapter, we develop an algorithm exploring to learn a low dimensional latent space in an unsupervised manner via regularized projection, which is capable of generalizing knowledge to unseen classes.

Chapter 7

Multi-Level Structured Image Coding on Object Bank

7.1 Introduction and Background

In this chapter, we propose a novel Multi-Level Structured Image Coding Approach, or MUSIC, to encode the original high-dimensional Object Bank representation with a much more compact, lower-dimensional, and semantically interpretable representation called *image code*, upon which visual recognition tasks such as image classification or retrieval, to name a few, can be effectively carried out with superior performance. Given the Object Bank representations of a set of images, MUSIC learns a set of image bases that span a lower-dimensional semantic space in which the whole image dataset can be embedded, and the sparse codes for every image, by minimizing a reconstruction error over the input Object Bank features. The key innovation behind MUSIC is a two-layer latent sparse coding scheme that leverages on the rich semantic and spatial information encoded in Object Bank to achieve compactness at both object and image levels; and the usage of a *structured object dictionary* that consists of both unique bases corresponding to every specific object in OB, and shared bases generic to all objects, upon which the Object Bank response signals of an input image can be reconstructed from the image code with high fidelity. The learned bases of the image codes carries information that relates objects and their spatial distributions. An

efficient coordinate descent algorithm is developed to solve the nontrivial optimization problem of code generation and dictionary learning in a fully unsupervised fashion.

In order to learn a compact image code from the high dimensional and over-complete Object Bank representation, we base our model MUSIC on sparse coding (SPC) [110], which has shown lots of success in learning descriptive presentations for audio [66] and image [144, 116]. Recent progress has been made on learning structured dictionaries by using composite regularizers [10, 75] and on developing highly efficient algorithms [75, 95]. MUSIC fundamentally differs from these methods. First, MUSIC is a two-layer SPC method and explicitly leverages the rich semantic and spatial information in Object Bank to achieve compactness at both object and image levels, while most existing SPC methods learn a flat representation, e.g., the single-layer encodings of image patches. Second, MUSIC learns a structured dictionary by explicitly defining object-specific and shared bases¹.

In summary, our contributions are:

1. Our learned image representation is a compact and robust representation that captures object and spatial information. Thus, the learned representation is efficient for future usage. It encodes the object appearance and spatial property and consequently preserves the ‘gist’ of the structural information in the representation.
2. We learn a generic ‘structure dictionary’ which essentially captures the relationship of objects and their spatial locations. The original structural information of the high dimensional representation can be easily reconstructed using this dictionary.
3. we introduce the concept of ‘shared bases’ to absorb shared information across different groups in the structure, and the concept of ‘specific bases’ to capture

¹We note that the very recent work [78] uses a similar idea of learning with specific/shared (or private/shared) dictionary elements for multi-view data analysis, but it is fundamentally different from MUSIC which is two-layer and explores much richer structural information. Comparing to [78], which softly couples the dictionaries on multi-views by using a structured regularizer, MUSIC is a multi-level model, which first explicitly defines object-specific and shared bases and then enforces bases selection. Although [78] could potentially be more flexible in identifying the shared and private bases, they are computationally much more demanding to obtain the comparable number of bases.

group-specific information.

7.2 Object Bank Revisit: a Structured High Dimensional Image Representation

In this section, we summarize the Object Bank (OB) representation briefly. Given an image, a scale pyramid is first built by down-sampling the image into 12 different scales. At each scale, an array of pre-trained object and stuff filters is applied to solicit a response value from each pixel location², an operation generating a response map per scale. A response map is then divided into multiple grids similar to the spatial pyramid representation [85]: 3 levels are used, where 1, 4, and 16 evenly divided grids are obtained for each response map (Fig. 5.1). Within each grid, maximum response score for each object filter is selected to build the final Object Bank representation. Specifically, let O denote the number of object filters and G denote the total number of grids for each input image ($G = nScales \times nGridsperScale$ i.e. $12 \times (1 + 4 + 16) = 252$). An Object Bank representation for each image is then constructed by concatenating a set of O *object-wise* sub-vectors $\mathbf{x} = [\mathbf{x}_1; \dots; \mathbf{x}_O]$, where each $\mathbf{x}_o \in \mathbb{R}^G$ denotes the responses produced by the o th object filter across all grids³. The overall dimensionality of the Object Bank representation is $O \times G$, which can easily grow into tens of thousands as the number of object filters increases.

The Object Bank representation departs from traditional low-level image representation by encoding both semantic and spatial information of an image. For example, the response maps of ‘sky’ or ‘airplane’ would show stronger signals in the upper half of the image, whereas ‘beach’ or ‘car’ would be the opposite. Furthermore, the Object Bank representation is by construction an over-complete characterization of the image, in which redundant information can be across different scales, overlapping spatial regions, and even at object levels.

²In our experiments, we follow the set-up of [93] and use the same 177 detectors.

³There can be an alternative representation, in which \mathbf{x} is organized as a concatenation of set of *grid-wise* sub-vectors, where each sub-vector $\mathbf{x}_g \in \mathbb{R}^O$ in this case denotes the responses produced by all the object filters at the grid g . But for simplicity, in our presentation of the method we focus on the object-wise concatenation.

7.3 Multi-Level Structured Image Coding

The redundancy in Object Bank representation potentiates a robust compression without scarifying semantic contents. We present the *Multi-Level Structured Image Coding* (MUSIC) for inferring a compact representation of the original Object Bank features in a semantic latent space spanned by a set of learned bases. We begin with a brief recapitulation of the basic sparse coding, upon which MUSIC is based.

7.3.1 Basic Sparse Coding

Widely applied in natural image, video and audio signal analysis [111, 110, 66], the sparse coding technique (SPC) [110] computes a sparse representation of input data \mathbf{x} in a latent space spanned by a set of bases $\beta = [\beta_1; \dots; \beta_K]$. The basis set is also known as a *dictionary*. An input vector \mathbf{x} can be reconstructed from a linear combination of the basis vectors, that is, $\mathbf{x} \approx \sum_{k=1}^K s_k \beta_k$, where the weight vector $\mathbf{s} \in \mathbb{R}^K$ is called a *sparse code* or simply *code*, which usually has very few non-zero elements. This approach can be directly applied to the Object Bank features, treating the entire $O \times G$ dimensional Object Bank representation \mathbf{x} of an image as an input sample, from which the latent-space representation can be computed using some matrix factorization methods. The problem is that this approach can be extremely inefficient because all the bases need to be as high-dimensional as the original OB. Moreover, this method does not consider the rich structural information in Object Bank discussed earlier.

7.3.2 The MUSIC Approach

To address the challenges posed by ultra high-dimensionality, structured regularity, and over-completeness of the Object Bank representation, MUSIC builds on the following main innovations over the basic SPC: **1)** Rather than computing a global code directly from the whole input vector \mathbf{x} , it computes sparse codes at *object-level* for each response subvector \mathbf{x}_o in \mathbf{x} , thus avoids the expensive high-dimensional matrix factorization. **2)** Rather than using a universal set of bases for reconstructing all \mathbf{x}_o 's,

it employs both (small) basis-sets unique to each object and a basis-set shared by all objects, which we refer to as a *structured dictionary*, to reconstruct every \mathbf{x}_o , thereby it avoids laboring a large unstructured and semantically uninterpretable dictionary for high fidelity reconstruction. **3)** Rather than obtaining a sparse reconstruction simply by pre-specifying the dictionary size (thereby the reconstruction vector dimension), we impose both object-level and image-level sparsity-inducing bias in the code inference and dictionary learning process. This enables us to incorporate structural knowledge such as preferred co-occurrence of objects, or appearance and/or filter-response, directly to the image codes, which is impossible in the basic SPC scheme described above.

Specifically, we seek to obtain a low dimensional projection θ of the original Object Bank features \mathbf{x} in a latent space spanned by a structured dictionary. Fig. 7.1 shows the structure of MUSIC, which is a two-layer latent variable model. Below, we describe our multi-level sparse coding scheme.

Object Coding in MUSIC

Given an input vector \mathbf{x} resultant from a concatenation of O subvectors, each corresponding to an object-specific spatial response, we begin by reconstructing each subvector \mathbf{x}_o from some bases in a dictionary, as shown in the first layer from \mathbf{x} to \mathbf{s} in Fig. 7.1. Let β denote a *structured dictionary*, consisting of O sets of object-wise *unique* bases, each denoted by $\beta_o \equiv \{\vec{\beta}_{o,1}, \dots, \vec{\beta}_{o,M}\}$ where M is the number of bases unique to every object; and one set of *shared* bases, denoted by $\beta_c \equiv \{\vec{\beta}_{c,1}, \dots, \vec{\beta}_{c,L}\}$ where L is the number of bases shared by all objects. Each basis $\vec{\beta}_k \in \mathbb{R}^G$ in the dictionary roughly represents a canonical response pattern of one object detector on different spatial locations in an image. We adopt a linear scheme for object-signal reconstruction:

$$\mathbf{x}_o \approx \sum_{j=1}^M u_{o,j} \vec{\beta}_{o,j} + \sum_{j'=1}^L v_{o,j'} \vec{\beta}_{o,j'} = \beta_o \vec{u}_o + \beta_c \vec{v}_o, \quad (7.1)$$

where \vec{u}_o and \vec{v}_o represent the vectors of weights measuring the contributions of the

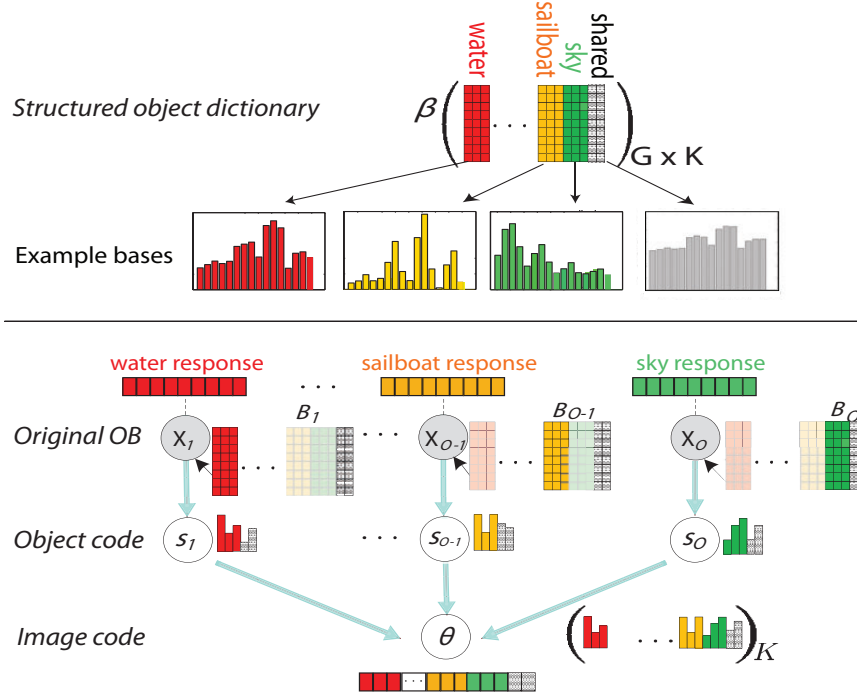


Figure 7.1: **Top:** an illustration of MUSIC for inferring a compact image code from high-dimensional Object Bank features. Here, $\mathbf{x}_o \in \mathbb{R}^G$ is the response of an object filter in the original OB, \mathbf{s}_o is the code of object o whose dimension is much lower than G (Sec. 7.3.2), θ is the image code (Sec. 7.3.2) that aggregates \mathbf{s}_o to achieve a single compressed representation for entire image, and B_o represents all the bases needed for reconstructing signals from object o . In B_o , the colored grids (column-wise) represent object-specific bases while the shaded grids represents shared bases. Across all variables, grids in the same color are directly correlated. **Bottom:** The learned structured object dictionary β . We show one example basis for each object and one shared basis. (This figure is best viewed in colors and with pdf magnification.)

unique bases and shared bases, respectively. We denote $\mathbf{s}_o \equiv \text{cat}(\vec{u}_o; \vec{v}_o)$, which is a concatenation of \vec{u}_o and \vec{v}_o , as the *code* of object o ; likewise, $B_o \equiv \text{cat}(\beta_o; \beta_c)$ ⁴ represents all the bases needed for reconstructing the response signals from object o .

Naively, one can code the signals $\mathbf{x}_o^{(d)}$ of image d by estimating $\mathbf{s}_o^{(d)}$ via a regularized loss minimization scheme:

$$\{\mathbf{s}_o^{(d)}\} = \arg \min \sum_d \|\mathbf{x}_o^{(d)} - B_o \mathbf{s}_o^{(d)}\|_2^2 + \rho \sum_d \|\mathbf{s}_o^{(d)}\|_1, \quad (7.2)$$

⁴Here $\text{cat}()$ denotes a column-wise matrix concatenation.

where ρ is a non-negative constant that balances the sparsity regularization term and the reconstruction error term. However, not only is B_o unknown in this function, but also, for different objects, the B_o 's contain a common element β_c that couples different codes for different objects. Finally, different \mathbf{x}_o 's from an image exhibit structural regularities, such as spatial preference of objects, appearance and/or filter-response similarities, which may render all \mathbf{s}_o 's of a single image not independent of each other. Therefore, in the sequel we need to furthermore consider a more global *image coding* built on the object coding.

Image Coding in MUSIC

To capture the correlations among the codes of different object-wise responses, MUSIC employs an additional layer of coding as shown in Fig. 7.1, which aggregates the codes \mathbf{s}_o of all object-wise sub-vectors to achieve a single compressed Object Bank representation θ of the entire image.

We define *image code* θ as a $K = (O \times M + L)$ dimensional vector of the form: $\theta \equiv \text{cat}(\theta_1; \dots; \theta_O; \theta_c)$, of which the elements $\theta_o, o = 1, \dots, O$ correspond to the *prototype code* of *each* of the unique portion \vec{u}_o of the object code \mathbf{s}_o ; and the last element θ_c corresponds to the *prototype code* underlying *all* of the shared portion \vec{v}_o . Similar to the object code extraction from \mathbf{x}_o 's, we adopt a regularized loss minimization scheme to extract θ from \mathbf{s}_o 's:

$$\{\theta\} = \arg \min \sum_{o=1}^O \|\theta_o - \vec{u}_o\|_2^2 + \|\theta_c - \vec{v}_o\|_2^2 + \lambda \sum_{i=1}^{O+1} \|\theta_i\|_2, \quad (7.3)$$

where λ is another regularization constant. The choice of loss function determines how the compact image code θ is aggregated from the individual representations \mathbf{s}_o . The use of a square error loss above will yield an average pooling for shared components and a re-weighted feature concatenation for object-specific components as we shall see. Note that, to introduce structured sparsity, here we employ an $\ell_{1,2}$ -norm over θ as a regularizer. This penalty can encourage joint sparsity of weights within an

object code over all regions, i.e., all elements in a subvector θ_o or θ_c is shrunk to zero simultaneously, which is a desirable bias to clean up spurious object filters. It is also possible to explore other structured sparsity, such as regional effects, but for simplicity, we leave these enhancements to the future work.

Just as for the case of \mathbf{s}_o 's, the coding problem for θ cannot be solved in isolation over each image only at the whole image level, because the object codes \mathbf{s}_o are not known in advance. Thus, as we shall show shortly, θ and \mathbf{s}_o must be estimated together in a joint optimization problem.

Structured Dictionary in MUSIC

As mentioned earlier, the *dictionary* β is a $G \times K$ matrix, where $K = O \times M + L$ denotes the total number of bases, and we use B_o to denote the sub-matrix $\text{cat}(\beta_o; \beta_c)$ for constructing the signals from object o . Fig. 7.1 illustrates the structure of such a dictionary, where the O object-specific sub-dictionaries β_o are indicated by different colors and the shared sub-dictionary β_c is indicated by shaded blocks. We can see that an object-specific basis represents a canonical spatial pattern for a particular object and a shared basis tends to capture the common spatial pattern of all the objects. Therefore, object-specific bases are sharp and vary a lot from one object to another, while a shared basis is much flatter. In MUSIC, the dictionary β is learned in conjunction with the coding process that renders \mathbf{s}_o and θ . We will present a closer examination of β in Sec. 7.4.1.

The MUSIC Model

Putting the three components presented above together, in order to learn an optimum dictionary β , and infer the optimum coding coefficients (\mathbf{s}, θ) , we define MUSIC as a coding/learning scheme based on minimizing a regularized square reconstruction error. Formally, given a set of images $\{\mathbf{x}^{(d)}\}$, we solve the following optimization problem which we call the MUSIC model:

Algorithm 2 Dictionary Learning

Input: image corpus $\{\mathbf{x}^{(d)}\}_{d=1}^D$, regularization constants (λ, γ, ρ) , basis numbers (M, L) .**Output:** dictionary β **repeat**Sparse Coding: *infer the sparse object codes \mathbf{s} and image code θ for each image using Alg. 3*Dictionary Learning: *solve the following convex problem for β with projected gradient descent*

$$\min_{\beta} \sum_{do} \|\mathbf{x}_o^{(d)} - B_o \mathbf{s}_o^{(d)}\|_2^2, \quad \text{s.t.}: \beta \in \mathbb{B}. \quad (7.5)$$

until convergence

$$\min_{\theta, \mathbf{s}, \beta} \sum_d L(\mathbf{x}^{(d)}; \mathbf{s}^{(d)}, \beta) + \gamma L(\mathbf{s}^{(d)}, \theta^{(d)}) + \rho \Omega(\mathbf{s}^{(d)}) + \lambda \psi(\theta^{(d)}) \quad \text{s.t.}: \beta \in \mathbb{B}, \quad (7.4)$$

where $L(\mathbf{x}^{(d)}; \mathbf{s}^{(d)}, \beta) = \sum_o \|\mathbf{x}_o^{(d)} - B_o \mathbf{s}_o^{(d)}\|_2^2$ is the square error between input features and their reconstructions; $L(\mathbf{s}^{(d)}, \theta^{(d)})$ is a similar square error between object codes and image code as defined in Sec. 7.3.2; $\Omega(\mathbf{s})$ is the ℓ_1 -norm of object codes as in problem (7.2); and $\psi(\theta)$ is the $\ell_{1,2}$ -norm of the image code as in problem (7.3). Here, (λ, γ, ρ) are pre-specified non-negative hyper-parameters, which can be chosen via cross validation. To make the problem identifiable, we put a constraint on the dictionary β . We define $\mathbb{B} = \{\beta : \sum_k \max_j |\beta_{kj}| \leq C\}$, which constrains the $\ell_{1,\infty}$ -norm of β to be less or equal to a threshold C . $\ell_{1,\infty}$ -norm is used because it can effectively avoid the spread of shared bases and bias the latent space towards being compact.

7.3.3 Optimization Algorithm: Coordinate Descent

We present an efficient procedure to solve problem (7.4). We note that the objective function is not joint convex, but it is bi-convex, that is, convex over one of (θ, \mathbf{s}) and β when the other is fixed. Therefore, a natural algorithm is the coordinate descent algorithm, which has been widely used in sparse coding [144] and high-dimensional

Algorithm 3 Sparse Coding for Compact Image Codes

Input: an image \mathbf{x} and object dictionary β , regularization constants (λ, γ, ρ) , basis numbers (M, L) .

Output: the image code θ and object codes \mathbf{s} .

repeat

for $o = 1$ **to** O **do**

 Solve the convex problem (7.2) for object-code \mathbf{s}_o .

end for

 Solve the convex problem (7.3) for θ .

until convergence

sparse learning problems [61]. The algorithm alternates between two steps of sparse coding and dictionary learning, as outlined in Alg. 2 and explained below.

Sparse coding: this step solves problem (7.4) for (θ, \mathbf{s}) with β fixed. This subproblem is convex and many optimization algorithms can be applied. Here, we adopt a coordinate descent strategy to iteratively solve it over θ and \mathbf{s} . Due to the independence assumption of different images, we can perform this step for each image separately. For notation simplicity, we will ignore the subscript of image index. The coordinate descent procedure is outlined in Alg. 3, where both problems (7.2) & (7.3) have closed-form solutions.

Since the sparse codes for different images are not coupled, we can perform this step for each image separately. We want to solve the problem for the image code θ and local object code \mathbf{s}_o . The coordinate descent procedure is as follows:

1. Optimize over θ : this step involves solving a group Lasso alike problem

$$\min_{\theta} \sum_{o=1}^O (\|\theta_o - \vec{u}_o\|_2^2 + \|\theta_c - \vec{v}_o\|_2^2) + \eta \sum_{i=1}^{O+1} \|\theta_i\|_2,$$

where $\eta = \lambda/\gamma$ is the ratio of the two regularization constants in problem 7.3. Here, we develop a coordinate descent method to solve for θ . For each element θ_o^k , the optimal condition is

$$-2(\vec{u}_o^k - \theta_o^k) + \eta p_o^k = 0,$$

where p_o^k is the k th component of the sub-gradient of $\|\theta_o\|_2$. It is easy to see that if $\theta_o \neq 0$, we have $p_o^k = \frac{\theta_o^k}{\|\theta_o\|_2}$; otherwise, we have $\|p_o\|_2^2 \leq 1$. Now, we check whether $\theta_o = 0$, which requires that $\|p_o\|_2^2 \leq 1$. Substituting $\theta_o = 0$ into the above optimal condition, we have $p_o^k = \frac{2}{\eta} \vec{u}_o^k$. Therefore, the sufficient condition for $\theta_o = 0$ is that $\|\vec{u}_o\|_2 \leq \frac{\eta}{2}$. If the condition holds, we have $\theta_o = 0$, and no update is needed for all its elements. Otherwise, if $\|\vec{u}_o\|_2 > \frac{\eta}{2}$, we have $\theta_o \neq 0$. Now, we need to solve for each of its element. From the optimal condition, we have

$$-2(\vec{u}_o^k - \theta_o^k) + \eta \frac{\theta_o^k}{\|\theta_o\|_2} = 0.$$

from which we have $\vec{u}_o^k = (1 + \frac{\eta}{2\|\theta_o\|_2})\theta_o^k$. Taking the ℓ_2 -norm, we have

$$\|\vec{u}_o\|_2 = (1 + \frac{\eta}{2\|\theta_o\|_2})\|\theta_o\|_2,$$

which implies that $\|\theta_o\|_2 = -\frac{\eta}{2} + \|\vec{u}_o\|_2$. Therefore, we have the solution: if $\|\vec{u}_o\|_2 \leq \frac{\eta}{2}$, we have $\theta_o = 0$; otherwise, we have

$$\theta_o^k = \vec{u}_o^k (1 - \frac{\eta}{2\|\vec{u}_o\|_2}).$$

We can follow the above procedure to derive the closed-form solution of the shared parameters θ_c . We omit it for clarity.

2. Optimize over \mathbf{s} : when β and θ are fixed, \mathbf{s}_o are not coupled. Therefore, we can solve the following reduced problem for each \mathbf{s}_o separately.

$$\min_{\mathbf{s}_o} \|\mathbf{x}_o - B_o \mathbf{s}_o\|_2^2 + \gamma(\|\vec{u}_o - \theta_o\|_2^2 + \|\vec{v}_o - \theta_c\|_2^2) + \rho \|\mathbf{s}_o\|_1.$$

We note that the objective function is a summation of a differentiable function and the non-differentiable ℓ_1 -norm. Here, we develop a coordinate descent algorithm, as popularly used in Lasso and many other variants. For the object-specific part \vec{u}_o , we have the closed-form solution.

$$\forall 1 \leq k \leq M, \quad \bar{u}_o^k = \begin{cases} \frac{\bar{s}_o^k - 0.5\rho}{\gamma + \|\beta_o^k\|_2}, & \text{if } \bar{s}_o^k > 0.5\rho \\ 0, & |\bar{s}_o^k| \leq 0.5\rho \\ \frac{\bar{s}_o^k + 0.5\rho}{\gamma + \|\beta_o^k\|_2}, & \text{if } \bar{s}_o^k < -0.5\rho, \end{cases}$$

where $\bar{s}_o^k = \gamma\theta_o^k + (\beta_o^k)^\top(\mathbf{x}_o - \sum_{i \neq k} \bar{u}_o^i \beta_o^i - \sum_i \bar{v}_o^i \beta_c^i)$ and β_o^k is the k th column of β_o , likewise for β_c^i . Because of the symmetry between \bar{u}_o and \bar{v}_o , the solution of the shared part \bar{v}_o has the similar form. We omit the derivation for clarity.

Update dictionary: this sub-step involves solving the convex problem (7.5), which is quadratic.

$$\min_{\beta} \sum_{do} \|\mathbf{x}_o^{(d)} - B_o \mathbf{s}_o^{(d)}\|_2^2, \quad \text{s.t.: } \beta \in \mathbb{B}.$$

This problem can be efficiently solved with projected gradient descent, where the projection to the $\ell_{1,\infty}$ -ball can be easily done [113].

7.4 Experiment

In this section, we thoroughly analyze a number of important properties of the MUSIC approach; and evaluate its performance on several high-level recognition tasks, including scene classification, image retrieval and annotation.

7.4.1 Analysis of MUSIC

Before showing performance of the proposed MUSIC approach on several benchmark applications, we examine some fundamental properties of MUSIC in this section. We focus on *interpretability* of the learned structured basis dictionary; *information content* of the image code, and *discriminability* of the MUSIC output over competing methods. We use the UIUC-Sports event dataset, which contains images from 8 event categories. 70 images from each class are used to learn the dictionary and compute the image codes. We use the same settings as [93], i.e. the original Object Bank

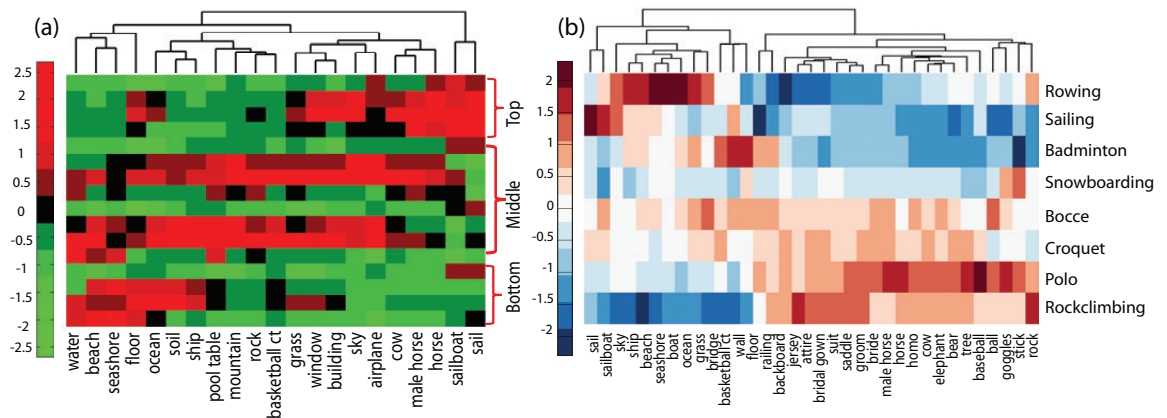


Figure 7.2: (a) A heat matrix of the bases in the structured dictionary. Each column corresponds to a basis, and each row corresponds to a spatial location (i.e., a grid), which are grouped as “Top”, “Middle” and “Bottom” locations in the image. Object names are displayed at the bottom. A high value of a basis-element in a row indicates that the object is likely to appear in the corresponding grid. The values of each object basis are standardized for salient visualization. (b) A heat matrix of the average image codes θ of images from different classes are displayed on the right. Names of the object-specific bases are displayed at the bottom.

is generated by using 177 object detectors. As described in Sec. 7.2, the overall dimensionality is $O \times G = 44604$. In MUSIC, 7 object specific bases for each object and 1 shared basis are used, which generates the image code with dimensionality of $7 \times 177 + 1 = 1240$, ~ 40 -fold reduction from the original Object Bank features.

A Close Examination of the Basis Dictionary

As mentioned earlier. MUSIC learns a structured dictionary that includes O subsets of object-specific bases, and 1 subset of shared bases. Each basis captures the canonical pattern of either an object-specific or a universal Object Bank signal in all the regions (i.e, grids) of the image. Fig. 7.2(a) shows a few examples of the learned object-specific bases (each as a column in the heat matrix); the columns are ordered as leaves of a hierarchical clustering of the columns for salient visualization. Each row in this matrix corresponds to a spatial location in the image where the bases are recording a canonical response of the object filters. As we can see, there is an apparent spatial bias revealed in different bases. Bases corresponding to objects such as “water”,

“beach”, “seashore”, etc. contain strong (i.e., red) signals at the bottom regions of the image; whereas bases corresponding to “maintain”, “pool table”, “rock”, etc. tend to have stronger signals in the middle regions of the image. Such spatial preference is consistent with human knowledge of the objects locations. A side effect of such spatial preference of objects is that a hierarchical clustering of the bases as we did here can sometime, but not always, generate semantically coherent grouping; for example, “water”, “beach” and “seashore” fall into a cluster on the left, whereas “sail” and “sailboat” are in the same cluster on the right. However, as spatial biases are not the only cue for grouping objects (e.g., see the apparent heterogeneous composition of the cluster in the middle), we are cautious about over-interpreting or exploiting this groupings.

A Close Examination of the Image Codes

We expect the image codes inferred by MUSIC from Object Bank to bear sufficient information content so that they lead to high-quality reconstruction of the original Object Bank signals, as well as being semantically interpretable.

Fig. 7.2(b) shows a heat matrix of subvectors of the average image codes obtained by extracting the object prototype code θ_o of 34 example objects for the 8 classes. Here, each row corresponds to an average image code for images from a category, and each column corresponds to an object-specific basis used. Again, columns are ordered by a hierarchical clustering simply for easy visualization. (We only show here the portion of image code corresponding to the object-specific bases, the remain corresponding to the shared bases, are, as expected, not as informative.) It can be seen that different image categories do exhibit biased usage of different object-specific bases, reflecting more frequent occurrences of the corresponding objects in the images. For example, “sail” and “sailboat” have higher image code values in the “sailing” class, while “sky”, “ship” and “boat” have higher image code values in the “rowing” class. Such content bias in the image code implies its potential in semantic-based discrimination, as we explore later.

Predictive Performance

Now we dissect the building blocks of MUSIC and examine their influences on the predictive power of the inferred image code. The evaluation is based on a small-scale scene classification experiment on the UIUC sports data. 70 images are used for training and 60 for test from each class, which is the default setting for all experiments on this data. In the following, if not specified, we employ a multi-class linear SVM as the default classifier operating on different image representations, including the image code from MUSIC. We compare with the following alternatives:

1. specSPC: image code from MUSIC, but using only object-specific bases (i.e., $L = 0$).
2. SPC: basic sparse coding that uses all shared bases to compute object codes separately, which were subsequently concatenated into a whole image-level representation⁵.
3. PCA: a representative dimensionality reduction method.
4. L1-LR: ℓ_1 -norm regularized logistic regression (LR) trained directly on the high-dimensional Object Bank representations, with ℓ_1 -norm regularization for direct feature selection.
5. SVM: a linear SVM learned from the original OB.

Table 7.1 summarizes the classification accuracy of different methods. For MUSIC, as we have stated, the dimension of the image codes is 1240, much lower than that of the original Object Bank (44604). We observe that the full MUSIC outperforms other algorithms. Specifically,

⁵If we use average or max-pooling in the SPC to obtain image-level representations, it is found that the performance will drop dramatically when only using a modest number (e.g., hundreds) of bases. Using a large number of bases could help but it is much more expensive than MUSIC. For example, suppose we use the same number of bases as the number of objects in both SPC and MUSIC (i.e., $M = 1$ & $L = 0$). Then, MUSIC will be roughly the number of objects times faster than the ordinary SPC. This is because SPC uses all the bases to reconstruct each object-wise Object Bank feature, while MUSIC uses only 1 basis to reconstruct each object-wise Object Bank feature.

Method	Accuracy
SVM [93]	77.9%
L1-LR	76.2%
PCA	77.2%
SPC	78.7%
specSPC	78.0%
MUSIC	81.8%

Table 7.1: Classification accuracy of different models.

1. The superior performance of MUSIC over specSPC demonstrates that shared bases can help separate common background information from the more semantic salient information regarding unique objects.
2. Although inferior to MUSIC, the basic SPC and PCA achieve fairly good performance, even slightly superior to L1-LR and linear SVM on the original Object Bank representation. This observation suggests that, possibly due to severe over-completeness of the original Object Bank features, standard feature selection methods such as L1-LR cannot effectively reduce the redundant features. However, using an appropriate linear transformation (e.g., SPC or PCA) can effectively reduce the redundancy and achieve a compact representation. MUSIC achieves a similar effect on dimensionality reduction, but is further benefited from the rich structure information in sparse coding.

7.4.2 Applications in High-level Image Recognition

In this section, we evaluate the potential of image code in high-level visual recognition tasks, specifically: scene classification, image retrieval and annotation.

Scene Classification

We first analyze the predictive power of the image codes learned by MUSIC for classifying scene images from two complex scene datasets – UIUC sports event [91] and MIT indoor scene [114]. For the MIT indoor scene dataset, we follow the settings

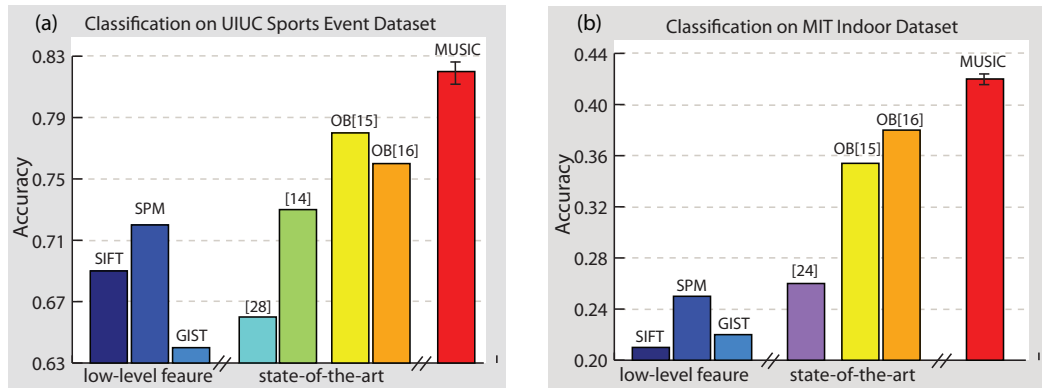


Figure 7.3: (a) Comparison of classification performance to the methods that use existing low-level representations or the original Object Bank representation and state-of-the-art approaches on the UIUC sports data. (b) Comparison of classification performance to the methods that use existing low-level representations, the original Object Bank representation, and state-of-the-art approaches on MIT Indoor data.

in [114], using 80 images from each of the 67 classes to train a multi-class linear SVM and test on a total of 1340 images (20 per class). We compare image code obtained by MUSIC with those methods using low-level features (e.g., SIFT, SPM and GIST) and the state-of-the-art algorithms. We use a linear SVM classifier for SIFT and GIST features and a more complex classifier (i.e. SVM with an intersection kernel) for SPM as in [85].

Fig. 7.3 shows the accuracy of different methods, defined as the average accuracy of a multi-way classification result. The improvement of image code from MUSIC over the low-level representations and the state-of-the-art approaches indicates image code can successfully preserve the rich structure and semantic meaning of the Object Bank representation. It is worth noticing that all state-of-the-art algorithms we compared with require extensive supervision during training ([138] and [91] use object labels within each training image, and [114] requires manual segmentation of a subset of training images.) whereas MUSIC does not require such supervision. The fact MUSIC outperforms the original Object Bank representation underscores the importance of obtaining a more compact feature, where richly discriminative information in both semantic and spatial domains are preserved, but the smaller dimensionality curtails the high-dimensionality challenge posed by the original OB.

Method	MUSIC-kNN	MUSIC-LR	MUSIC	Self-Taught
Accuracy	69.5%	79.2%	81.8%	80.4%

Table 7.2: Classification performance by using different classifiers and self-taught learning (we learn a MUSIC on the MIT indoor data and apply it to infer image codes for images in the the UIUC sports data) on the image codes inferred by MUSIC.

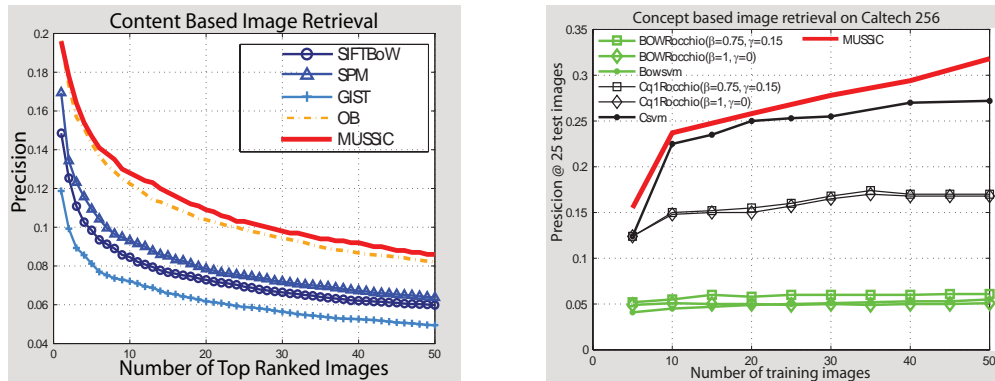


Figure 7.4: **Left:** Content based image retrieval: precision of the the top ranked images by using GIST, BOW, SPM, original OB, and image code on the UIUC sports event dataset. Cosine distance is used as the distance measurement. **Right:** Average precision of the top N images in Caltech 256 dataset. “Cq1Rocchio” and “Csvm” are obtained by applying Rocchio algorithm [27] and SVM to the Clasesmes [131], whereas “BoWRocchio” and “BoWsvm” are from Bag-of-Words representation. Performance scores are cited from Fig.4 in [131]

We also investigate how different end classifiers affect the classification performance by using our image codes from MUSIC. Here, we compare linear SVM (the default classifier) with kNN and logistic regression (LR), which are denoted by MUSIC-kNN and MUSIC-LR, respectively. We also report the performance of using our image codes for self-taught learning [116]. As shown in Table 7.2, the linear LR and SVM perform comparably. Although inferior to SVM or LR, kNN is comparable to the best method that uses a low-level representation (e.g., SPM [85]) and the state-of-the-art methods as shown in Fig. 7.3(a).

Image Retrieval

We investigate the usefulness of our image code inferred from MUSIC on image retrieval tasks in two scenarios:

Content-based Image Retrieval (CBIR) – use a query image to retrieve relevant images. We compare the retrieval performance of using image codes to the methods using low-level representations and the original OB [93] on the UIUC sports data, where 130 images from each of the 8 classes are used. We use precision of the top ranked images as the evaluation criterion (Fig. 7.4(Left)). Image code by MUSIC outperforms those using low-level representations with a large margin. We attribute this advantage of MUSIC to its encoding of rich semantic and spatial information. It is worth noticing that although achieving comparable performance, the image code has a much lower dimension (1240) and hence more efficient for practical applications than the Object Bank representation (>40k dimension).

Concept-based Retrieval – use a concept to retrieve images. As shown in Fig. 7.2(a), our object dictionary has clear object specific patterns. In Fig. 7.2(b), we also show that such patterns can directly relate a group of images sharing one concept based on the objects shared among them, which is useful for concept retrieval. Here, we apply the image code to concept retrieval on Caltech 256 dataset [64] and compare with the state-of-the-art algorithms, Torresani et. al. [131] and Rocchio algorithm [27]. Similar to [131], we train an SVM on each retrieval concept. We then rank the retrieved images based on the SVM prediction score. Following the evaluation criteria used in [131], we report the precision of top 25 retrieved images by using different representations in Fig. 7.4(Right). We observe superior performance to the method of Torresani [131], which is largely attributed to the incorporation of the spatial patterns of the objects⁶.

Image Annotation

Our last experiment is to apply the image code to an image annotation task, where a list of image or object concepts is inferred for an image. We conduct this experiment on the UIUC sports data, with 70 images per class for training (18 of them are used as the validation set) and 60 for testing. We train an SVM classifier of each object based on our representation of the images. Given a query image, the classifier

⁶The compactness of image code makes it a better choice over OB [93] for large scale retrieval application.

makes an annotation prediction for each concept. Our annotation result (48.27% in standard F-measure, used in [138]) is superior than the reported state-of-the-art performance of 38.20% in [138]. We attribute this improvement to the rich semantic information encoded by the object detectors, which is suitable for high-level visual tasks such as image annotation. Our method also outperforms the original Object Bank representations (45.46%), which suggests that the more compact image codes by MUSIC successfully preserve the semantic contents of the image while discarding the noise and redundancy.

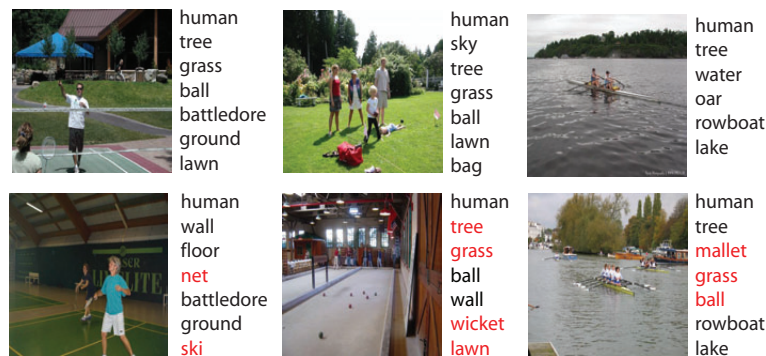


Figure 7.5: Example image annotation results by MUSIC. Proposed tags are listed on the right side of the image. Incorrect tags are highlighted in red. The average number of tags proposed is ~ 10 . For those images with more than 7 tags predicted, only the top 7 annotated tags with highest empirical frequencies in the tag list of that image are shown.

Fig. 7.5 shows a few example results annotated by MUSIC. One source of mistakes is due to semantic confusion, e.g. “net” in 2nd row and 1st column is proposed as a result of its expected occurrence in badminton images. Another source of errors is the object filters in OB, such as incorrectly labeling “ski” as “stick”. These observations point out a number of useful future directions in improving our work.

7.5 Discussion

We have proposed a novel MUSIC model that learns a structured object dictionary from images using a high-level representation (i.e., OB) and infers much more compact

image code representation ($\sim 1\text{k}$ dimension) than the original ultra-high dimensional Object Bank (44604 dimension). Our analysis demonstrates that the structural regularity in the learned dictionary and the inferred image codes are consistent with human knowledge. Using the inferred image codes, superior performance over original Object Bank can be obtained with a much lower computational cost on various high-level image recognition tasks. We plan to explore the potential of the compact image codes in large scale recognition problems, which are infeasible for the original Object Bank due to its high dimensionality and over-completeness.

Part IV

Conclusion

In this thesis, we have focused on two important aspects in semantic image understanding: model representation and feature representation. Specifically, we have proposed principle probabilistic models to represent the objects, scene environments, and event/activities in real-world complex images. To represent the semantic meaning in our complex visual world, we propose a fundamentally new high level image representation which encodes the objects appearance and geometric locations in the image. Below, we will summarize the contributions of my thesis.

In Chapter 1, we tackle a classical problem in computer vision, object recognition. We propose OPTIMOL (a novel framework for Online Picture collectiOn via Incremental MOdel Learning), an automatic dataset collecting and model learning system for object recognition. Our algorithm mimics the human learning process in such a way that, starting from a few training examples, the more confident data you incorporate in the training data, the more reliable models can be learnt. Our system uses the Internet as the (nearly) unlimited resource for images. The learning and image collection processes are done via an iterative and incremental scheme designed for large scale image recognition task. The goal of this work is to use this tremendous web resource to learn robust object category models in order to detect and search for objects in real-world scenes. OPTIMOL is able to automatically collect much larger object category datasets nearly as accurate as those collected by humans. It offers not only more images in each object category dataset, but also a robust object model and meaningful image annotation. It demonstrates excellent recognition and localization abilities in large scale object recognition as well as robot navigation in real world scenario. In Semantic Robot Vision Challenge, a competition designed to fuse the state of the art in image understanding and automatic acquisition of knowledge from large unstructured databases of images (such as those generally found on the web), OPTIMOL framework won the 1st place prize in the software league.

There is more meaningful content beyond objects in an image, e.g. scenes, events, activities, emotions, and intentions. In Chapter 2, we propose a first attempt to classify complex scenes/events in static images by integrating scene and object information. We observe that object recognition in the scene as well as scene environment classification of the image facilitate each other in the overall activity recognition

task. We formulate this observation in a principle probabilistic model representation where activity classification is achieved by combining information from both the object recognition and the scene classification pathways. While most existing research focus on recognizing isolated objects and object classes in an image, we propose a unified framework to classify an image by recognizing, annotating and segmenting the objects within the image. To our knowledge, this is the first model that performs all three tasks in one coherent framework. Our generative model jointly explains images through a visual model and a textual model. Visually relevant objects are represented by regions and patches, while visually irrelevant textual annotations are influenced directly by the overall scene class. We propose a fully automatic learning framework that is able to learn robust scene models from noisy web data such as images and user tags from Flickr.com. We demonstrate the effectiveness of our framework by automatically classifying, annotating and segmenting images from eight classes depicting sport scenes. By jointly modeling of classification, annotation and segmentation, our model significantly outperforms state-of-the-art algorithms in all three tasks.

In Chapter 3, we propose to construct a meaningful image hierarchy to ease the human effort in organizing thousands and millions of pictures (e.g., personal albums). Two types of hierarchies have recently been explored in computer vision for describing the relationship among images: language-based hierarchy and low-level visual feature-based hierarchy. Pure language-based lexicon taxonomies, such as WordNet, are useful to guide the meaningful organization of images. However, they ignore important visual information that connects images together. On the other hand, purely visual feature-based hierarchies are difficult to interpret, and arguably not as useful. Motivated by their drawbacks, we propose to automatically construct a semantically and visually meaningful hierarchy of texts and images on the Internet by learning a non-parametric hierarchical model. The quality of the hierarchy was quantitatively evaluated by human subjects. Furthermore, we demonstrate that a good image hierarchy can serve as a knowledge ontology for end tasks such as image retrieval, annotation and classification.

Traditional low level image representations based on color and/or gradient have achieved promising progress in visual recognition. However, color and gradient carry

very little semantic meaning resulting in the so called “semantic gap” between the low level image representation and the high level visual recognition tasks. On the other hand, it is a remarkable fact that images are related to objects constituting them. In Chapter 5, we introduce the novel concept of Object Bank, the first high-level image representation encoding object appearance and spatial location information in images. Object Bank represents an image based on its response to a large number of pre-trained object detectors, or “object filters”, blind to the testing dataset and visual recognition task. Our Object Bank representation is a fundamentally new image representation. It is a sharp departure from all previous image representations and provides essential information for semantic image understanding. It demonstrates promising potential in high level image recognition tasks. It achieves state-of-the-art performance in image classification on various benchmark image datasets by using simple, off-the-shelf classification algorithms such as linear SVM and logistic regression. In Chapter 6, we demonstrate that sparsity algorithms make our representation more efficient and scalable for large scene dataset, and reveal semantic meaningful feature patterns. Towards the goal of developing descriptive, efficient and scalable image representation, we further propose an unsupervised structural sparsification model for compressing Object Bank, with an innovative multi-level structural regularization scheme (Chapter 7). Our model learns an object dictionary that offers intriguing intuition of real-world image structures. Moreover, we show that the new, more compact representation outperforms a number of state-of-the-art image representations on a wide range of high-level visual tasks such as scene classification, image retrieval and annotation.

Bibliography

- [1] The PASCAL object recognition database collection. <http://www.pascal-network.org/challenges/VOC/databases.html>, 2006.
- [2] Shivani Agarwal, Aatif Awan, and Dan Roth. Learning to detect objects in images via a sparse, part-based representation. *PAMI*, 26(11):1475–1490, 2004.
- [3] N. Ahuja and S. Todorovic. Learning the Taxonomy and Models of Categories Present in Arbitrary Images. In *ICCV 2007. IEEE 11th International Conference on*, pages 1–8, 2007.
- [4] P. Arbeláez and L. Cohen. Constrained image segmentation from hierarchical boundaries. In *IEEE Conference on Computer Vision and Pattern Recognition, 2008. CVPR 2008*, pages 1–8, 2008.
- [5] K. Barnard, P. Duygulu, D. Forsyth, N. de Freitas, D.M. Blei, and M.I. Jordan. Matching words and pictures. *The Journal of Machine Learning Research*, 3, 2003.
- [6] K. Barnard and D. Forsyth. Learning the semantics of words and pictures. In *Eighth IEEE International Conference on Computer Vision*, pages 408–415, 2001.
- [7] E. Bart, I. Porteous, P. Perona, and M. Welling. Unsupervised Learning of Visual Taxonomies. *Proc. Computer Vision and Pattern Recognition 2008*.

- [8] P.N. Belhumeur, J.P. Hespanha, and D.J. Kriegman. Eigenfaces vs. Fisherfaces: recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):711–720, 1997.
- [9] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 509–522, 2002.
- [10] S. Bengio, F. Pereira, Y. Singer, and D. Strelow. Group sparse coding. In *NIPS*, 2009.
- [11] T.L. Berg and D.A. Forsyth. Animals on the web. In *Proc. Computer Vision and Pattern Recognition*, 2006.
- [12] J. Besemer, A. Lomsadze, and M. Borodovsky. GeneMarkS: a self-training method for prediction of gene starts in microbial genomes. Implications for finding sequence motifs in regulatory regions. *Nucleic Acids Research*, 29(12):2607, 2001.
- [13] I. Biederman. Recognition-by-components: A theory of human image understanding. *Psychological review*, 94(2):115, 1987.
- [14] C.M. Bishop. *Pattern recognition and machine learning*. Springer New York., 2006.
- [15] D.M. Blei, T.L. Griffiths, M.I. Jordan, and J.B. Tenenbaum. Hierarchical Topic Models and the Nested Chinese Restaurant Process. In *Advances in Neural Information Processing Systems*, 2004.
- [16] D.M. Blei and M.I. Jordan. Modeling annotated data. *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, 2003.
- [17] D.M. Blei, A.Y. Ng, and M.I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

- [18] A. Bosch, A. Zisserman, and X. Munoz. Scene classification via pLSA. *Proc. ECCV*, 4:517–530, 2006.
- [19] B.E. Boser, I.M. Guyon, and V.N. Vapnik. A training algorithm for optimal margin classifiers. *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152, 1992.
- [20] L. Bourdev and J. Malik. Poselets: Body Part Detectors Trained Using 3D Human Pose Annotations. *ICCV*, 2009.
- [21] L. Cao and L. Fei-Fei. Spatially Coherent Latent Topic Model for Concurrent Segmentation and Classification of Objects and Scenes. *ICCV 2007*, 2007.
- [22] L. Cao, J. Yu, J. Luo, and T.S. Huang. Enhancing semantic and geographic annotation of web images via logistic canonical correlation regression. In *Proceedings of the seventeen ACM international conference on Multimedia*, pages 125–134. ACM, 2009.
- [23] P. Carbonetto, N. de Freitas, and K. Barnard. A statistical model for general contextual object recognition. *Proceedings of ECCV*, 2004.
- [24] C. Carson, M. Thomas, S. Belongie, J.M. Hellerstein, and J. Malik. Blobworld: A system for region-based image indexing and retrieval. *Third International Conference on Visual Information Systems*, pages 509–516, 1999.
- [25] Jonathan Chang, Jordan Boyd-Graber, Chong Wang, Sean Gerrish, and David M. Blei. Reading tea leaves: How humans interpret topic models. In *NIPS*, 2009.
- [26] Y. Chen, J.Z. Wang, and R. Krovetz. Content-based image retrieval by clustering. *Proceedings of the 5th ACM SIGMM international workshop on Multimedia information retrieval*, pages 193–200, 2003.
- [27] M. Christopher, R. Prabhakar, and S. Hinrich. Introduction to information retrieval, 2008.

- [28] B Collins, J. Deng, K. Li, and L. Fei-Fei. Toward scalable dataset construction: An active learning approach. In *Proc. ECCV*, 2008.
- [29] D.J. Crandall, L. Backstrom, D. Huttenlocher, and J. Kleinberg. Mapping the world’s photos. In *Proceedings of the 18th international conference on World wide web*, pages 761–770. ACM, 2009.
- [30] N. Cristianini and J. Shawe-Taylor. *An introduction to support Vector Machines: and other kernel-based learning methods*. Cambridge Univ Pr, 2000.
- [31] G. Csurka, C. Bray, C. Dance, and L. Fan. Visual categorization with bags of keypoints. *Workshop on Statistical Learning in Computer Vision, ECCV*, pages 1–22, 2004.
- [32] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, page 886, 2005.
- [33] R. Datta, W. Ge, J. Li, and J.Z. Wang. Toward bridging the annotation-retrieval gap in image search. *IEEE MULTIMEDIA*, 14(3):24, 2007.
- [34] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.
- [35] Y. Deng, BS Manjunath, C. Kenney, MS Moore, and H. Shin. An efficient color representation for image retrieval. *Image Processing, IEEE Transactions on*, 10(1):140–147, 2001.
- [36] G. Dorko and C. Schmid. Object class recognition using discriminative local features. *IEEE PAMI*, submitted.
- [37] P. Duygulu, K. Barnard, N. de Freitas, and D. Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. *European Conference on Computer Vision*, 2002.
- [38] B. Edition and BNC Sampler. British National Corpus.

- [39] A. Farhadi, I. Endres, and D. Hoiem. Attribute-centric recognition for cross-category generalization. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2352–2359. IEEE, 2010.
- [40] L. Fei-Fei, R. Fergus, and P. Perona. A Bayesian approach to unsupervised one-Shot learning of object categories. In *Proceedings of International Conference on Computer Vision*, pages 1134–1141, October 2003.
- [41] 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. In *Workshop on Generative-Model Based Vision*, 2004.
- [42] L. Fei-Fei, R. Fergus, and P. Perona. One-Shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2006.
- [43] L. Fei-Fei, R. Fergus, and A. Torralba. Recognizing and learning object categories. Short Course CVPR: <http://people.csail.mit.edu/torralba/shortCourseRLOC/index.html>, 2007.
- [44] L. Fei-Fei, A. Iyer, C. Koch, and P. Perona. What do we see in a glance of a scene? *Journal of Vision*, 7(1):10, 1–29, 2007. <http://journalofvision.org/7/1/10/>, doi:10.1167/7.1.10.
- [45] L. Fei-Fei and L.-J. Li. What, Where and Who? Telling the Story of an Image by Activity Classification, Scene Recognition and Object Categorization. *Studies in Computational Intelligence- Computer Vision*, pages 157–171, 2010.
- [46] L. Fei-Fei and P. Perona. A Bayesian hierarchy model for learning natural scene categories. *Computer Vision and Pattern Recognition*, 2005.
- [47] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object Detection with Discriminatively Trained Part Based Models. *Journal of Artificial Intelligence Research*, 29, 2007.
- [48] P. Felzenszwalb and D. Huttenlocher. Pictorial structures for object recognition. *IJCV*, 1:55–79, 2005.

- [49] P.F. Felzenszwalb and D.P. Huttenlocher. Efficient Graph-Based Image Segmentation. *IJCV*, 2004.
- [50] H.M. Feng and T.S. Chua. A bootstrapping approach to annotating large image collection. *Proceedings of the 5th ACM SIGMM international workshop on Multimedia information retrieval*, pages 55–62, 2003.
- [51] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. Learning Object Categories from Google Image Search. *ICCV 2005*, 2, 2005.
- [52] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proc. Computer Vision and Pattern Recognition*, pages 264–271, 2003.
- [53] R. Fergus, P. Perona, and A. Zisserman. A visual category filter for Google images. In *ECCV*, 2004.
- [54] R. Fergus, P. Perona, and A. Zisserman. A sparse object category model for efficient learning and exhaustive recognition. In *Proc. Computer Vision and Pattern Recognition*, 2005.
- [55] T.S. Ferguson. A Bayesian Analysis of Some Nonparametric Problems. *The Annals of Statistics*, 1(2):209–230, 1973.
- [56] V. Ferrari and A. Zisserman. Learning visual attributes. *NIPS*, 2007.
- [57] M. Fink and S. Ullman. From aardvark to zorro: A benchmark of mammal images. 2007.
- [58] W.T. Freeman and E.H. Adelson. The design and use of steerable filters. *IEEE Transactions on Pattern analysis and machine intelligence*, 13(9):891–906, 1991.
- [59] Y. Freund and R.E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *Computational Learning Theory: Second European Conference, EuroCOLT'95, Barcelona, Spain, March 13-15, 1995: Proceedings*, page 23. Springer, 1995.

- [60] Y. Freund and R.E. Schapire. Experiments with a new boosting algorithm. In *MACHINE LEARNING-INTERNATIONAL WORKSHOP THEN CONFERENCE-*, pages 148–156. MORGAN KAUFMANN PUBLISHERS, INC., 1996.
- [61] J. Friedman, T. Hastie, and R. Tibshirani. A note on the group lasso and a sparse group lasso. *preprint, available at <http://www-stat.stanford.edu/tibs>*, 2010.
- [62] A.E. Gelfand and A.F.M. Smith. Sampling-based approaches to calculating marginal densities. *Journal of the American statistical association*, 1990.
- [63] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans Pattern Analysis and Machine Intelligence*, pages 721–741, 1984.
- [64] G. Griffin, A. Holub, and P. Perona. Caltech-256 Object Category Dataset. 2007.
- [65] G. Griffin and P. Perona. Learning and Using Taxonomies For Fast Visual Categorization. In *CVPR*, 2008.
- [66] R. Grosse, R. Raina, H. Kwong, and A. Ng. Shift-invariant sparse coding for audio classification. In *UAI*, 2007.
- [67] A. Hauptmann, R. Yan, W. Lin, M. Christel, and H. Wactlar. Can high-level concepts fill the semantic gap in video retrieval? a case study with broadcast news. *IEEE Transactions on Multimedia*, 9(5):958, 2007.
- [68] G. Heitz, S. Gould, A. Saxena, and D. Koller. Cascaded classification models: Combining models for holistic scene understanding. *Proceedings of Neural Information Processing Systems. Vancouver, Canada: NIPS*, 8, 2008.
- [69] T. Hofmann. Probabilistic latent semantic indexing. *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57, 1999.

- [70] D. Hoiem, A.A. Efros, and M. Hebert. Automatic photo pop-up. *Proceedings of ACM SIGGRAPH 2005*, 24(3):577–584, 2005.
- [71] D. Hoiem, A.A. Efros, and M. Hebert. Putting Objects in Perspective. *CVPR*, 2006.
- [72] D. Hoiem, A.A. Efros, and M. Hebert. Putting Objects in Perspective. *Proc. IEEE Computer Vision and Pattern Recognition*, 2006.
- [73] N. Ide and C. Macleod. The american national corpus: A standardized resource of american english. In *Proceedings of Corpus Linguistics 2001*, pages 274–280. Citeseer, 2001.
- [74] A.K. Jain and A. Vailaya. Image retrieval using color and shape. *Pattern Recognition*, 29(8):1233–1244, 1996.
- [75] R. Jenatton, J. Mairal, G. Obozinski, and F. Bach. Proximal methods for sparse hierarchical dictionary learning. In *ICML*, 2010.
- [76] J. Jeon, V. Lavrenko, and R. Manmatha. Automatic image annotation and retrieval using cross-media relevance models. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 119–126. ACM New York, NY, USA, 2003.
- [77] J. Jia, N. Yu, and X.S. Hua. Annotating personal albums via web mining. 2008.
- [78] Y. Jia, M. Salzmann, and T. Darrell. Factorized latent spaces with structured sparsity. In *NIPS*, 2010.
- [79] Y. Jin and S. Geman. Context and hierarchy in a probabilistic image model. *CVPR*, 2006.
- [80] Y. Jin, L. Khan, L. Wang, and M. Awad. Image annotations by combining multiple evidence & wordNet. In *Proceedings of the 13th annual ACM international conference on Multimedia*, pages 706–715. ACM New York, NY, USA, 2005.

- [81] D. Joshi, R. Datta, Z. Zhuang, WP Weiss, M. Friedenberg, J. Li, and J.Z. Wang. Paragrab: A comprehensive architecture for web image management and multimodal querying. In *VLDB*, 2006.
- [82] T. Kadir and M. Brady. Scale, saliency and image description. *International Journal of Computer Vision*, 45(2):83–105, 2001.
- [83] S. Krempp, D. Geman, and Y. Amit. Sequential learning with reusable parts for object detection. Technical report, Johns Hopkins University, 2002.
- [84] M. Pawan Kumar, Philip H. S. Torr, and A. Zisserman. Obj cut. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1*, pages 18–25, Washington, DC, USA, 2005. IEEE Computer Society.
- [85] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. 2006.
- [86] Y. LeCun, F. Huang, and L. Bottou. Learning methods for generic object recognition with invariance to pose and lighting. In *CVPR*, 2004.
- [87] B. Leibe and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *Proc. Workshop on statistical learning in computer vision*, Prague, Czech Republic, 2004.
- [88] T. Leung and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *International Journal of Computer Vision*, 43(1):29–44, 2001.
- [89] J. Li and JZ Wang. Automatic Linguistic Indexing of Pictures by a statistical modeling approach. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(9):1075–1088, 2003.
- [90] J. Li, J.Z. Wang, and G. Wiederhold. IRM: integrated region matching for image retrieval. *Proceedings of the eighth ACM international conference on Multimedia*, pages 147–156, 2000.

- [91] L-J. Li and L. Fei-Fei. What, where and who? classifying events by scene and object recognition. In *ICCV*, 2007.
- [92] L-J. Li, R. Socher, and L. Fei-Fei. Towards total scene understanding: classification, annotation and segmentation in an automatic framework. In *CVPR*, 2009.
- [93] L-J. Li, H. Su, Y Lim, and L. Fei-Fei. Objects as attributes for scene classification. In (*ECCV*), *Workshop on PaA*, 2010.
- [94] D. Lowe. Object recognition from local scale-invariant features. In *ICCV*, 1999.
- [95] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. *JMLR*, 11:19 – 60, 2010.
- [96] Tomasz Malisiewicz and Alexei A. Efros. Recognition by association via learning per-exemplar distances. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, June 2008.
- [97] O. Maron and A.L. Ratan. Multiple-instance learning for natural scene classification. In *In The Fifteenth International Conference on Machine Learning*. Citeseer, 1998.
- [98] M. Marszalek and C. Schmid. Constructing Category Hierarchies for Visual Recognition. *Proc. European Conference of Computer Vision*.
- [99] M. Marszalek and C. Schmid. Semantic Hierarchies for Visual Object Recognition. In *CVPR*, pages 1–7, 2007.
- [100] A. McCallum, D. Freitag, and F. Pereira. Maximum entropy Markov models for information extraction and segmentation. *Proc. 17th International Conf. on Machine Learning*, pages 591–598, 2000.
- [101] D. McClosky, E. Charniak, and M. Johnson. Effective self-training for parsing. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational*

- Linguistics*, pages 152–159. Association for Computational Linguistics Morristown, NJ, USA, 2006.
- [102] K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *Proc. European Conference on Computer Vision*, volume 1, pages 128–142, 2002.
- [103] G.A. Miller. WordNet: A Lexical Database for English. *COMMUNICATIONS OF THE ACM*, 1995.
- [104] K. Murphy, A. Torralba, and W.T. Freeman. Using the forest to see the trees: a graphical model relating features, objects and scenes. In *NIPS (Neural Info. Processing Systems)*, 2004.
- [105] R. Neal and G. Hinton. A view of the EM algorithm that justifies incremental, sparse and other variants. In M.I. Jordan, editor, *Learning in Graphical Models*, pages 355–368. Kluwer academic press, Norwell, 1998.
- [106] R.M. Neal. Markov Chain Sampling Methods for Dirichlet Process Mixture Models. *JOURNAL OF COMPUTATIONAL AND GRAPHICAL STATISTICS*, 9(2):249–265, 2000.
- [107] R. Nevatia Thomas O. Description and recognition of curved objects* 1. *Artificial Intelligence*, 8(1):77–98, 1977.
- [108] S. Obdrzalek and J. Matas. Object recognition using local affine frames on distinguished regions. *Proc. British Machine Vision Conference*, pages 113–122, 2002.
- [109] A. Oliva and A. Torralba. Modeling the shape of the scene: a holistic representation of the spatial envelope. *IJCV.*, 2001.
- [110] B. A. Olshausen and D. J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, 1996.

- [111] B. A. Olshausen and D. J. Field. Sparse coding of sensory inputs. *Current Opinion in Neurobiology*, 14:481–487, 2004.
- [112] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *PAMI*, 12(7):629–639, 1990.
- [113] A. Quattoni, X. Carreras, M. Collins, and T. Darrell. An efficient projection for $\ell_{1,\infty}$ regularization. In *ICML*, 2009.
- [114] A. Quattoni and A. Torralba. Recognizing indoor scenes. *CVPR*, 2009.
- [115] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie. Objects in context. In *IEEE International Conference on Computer Vision*, 2007.
- [116] Rajat Raina, Alexis Battle, Honglak Lee, Benjamin Packer, and Andrew Y. Ng. Self-taught learning: Transfer learning from unlabeled data. In *ICML*, 2007.
- [117] E. Rosch, C.B. Mervis, W. Gray, D. Johnson, and P. Boyes-Braem. Basic objects in natural categories. *Cognitive Psychology: Key Readings*, page 448, 2004.
- [118] C. Rosenberg, M. Hebert, and H. Schneiderman. Semi-supervised selftraining of object detection models. In *Seventh IEEE workshop on applications of computer vision*, 2005.
- [119] B.C. Russell, A. Torralba, K.P. Murphy, and W.T. Freeman. Labelme: a database and web-based tool for image annotation. 2005.
- [120] F. Schroff, A. Criminisi, and A. Zisserman. Harvesting image databases from the web. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, 2007.
- [121] J. Sethuraman. A constructive definition of Dirichlet priors. *Statistica Sinica*, 4(2):639–650, 1994.
- [122] J. Sivic, B.C. Russell, A. Efros, A. Zisserman, and W.T. Freeman. Discovering object categories in image collections. In *ICCV*, 2005.

- [123] J. Sivic, B.C. Russell, A. Zisserman, W.T. Freeman, and A.A. Efros. Unsupervised discovery of visual object class hierarchies. In *Proc. CVPR*, 2008.
- [124] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Ninth IEEE International Conference on Computer Vision, 2003. Proceedings*, pages 1470–1477, 2003.
- [125] R. Snow, D. Jurafsky, and A.Y. Ng. Semantic taxonomy induction from heterogeneous evidence. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 801–808. Association for Computational Linguistics Morristown, NJ, USA, 2006.
- [126] E. Sudderth, A. Torralba, W. Freeman, and A. Willsky. Describing visual scenes using transformed dirichlet processes. *Advances in Neural Information Processing Systems*, 18:1297–1304, 2005.
- [127] E. Sudderth, A. Torralba, W.T. Freeman, and A. Willsky. Learning hierarchical models of scenes, objects, and parts. In *Proc. International Conference on Computer Vision*, 2005.
- [128] M. Szummer and R. Picard. Indoor-outdoor image classification. In *Int. Workshop on Content-based Access of Image and Video Databases*, Bombay, India, 1998.
- [129] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical Dirichlet processes. *To appear in Journal of the American Statistical Association*, 2006.
- [130] A. Torralba, R. Fergus, and W.T. Freeman. Tiny images. Technical report, MIT-CSAILTR-2007-024, Computer Science and Artificial Intelligence Lab, Massachusetts Institute of Technology, 2007.
- [131] L. Torresani, M. Szummer, and A. Fitzgibbon. Efficient Object Category Recognition Using Classemes. *ECCV*, 2010.

- [132] Z. Tu, X. Chen, A.L. Yuille, and S.C. Zhu. Image Parsing: Unifying Segmentation, Detection, and Recognition. *International Journal of Computer Vision*, 63(2):113–140, 2005.
- [133] S. Vijayanarasimhan and K. Grauman. Keywords to visual categories: Multiple-instance learning for weakly supervised object categorization. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.
- [134] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. Computer Vision and Pattern Recognition*, volume 1, pages 511–518, 2001.
- [135] J. Vogel and B. Schiele. A semantic typicality measure for natural scene categorization. In *DAGM'04 Annual Pattern Recognition Symposium*, Tuebingen, Germany, 2004.
- [136] L. Von Ahn. Games with a purpose. *Computer*, 39(6):92–94, 2006.
- [137] M.J. Wainwright, P. Ravikumar, and J.D. Lafferty. High-dimensional graphical model selection using ℓ_1 -regularized logistic regression. *Advances in neural information processing systems*, 19:1465, 2007.
- [138] C. Wang, D. Blei, and L. Fei-Fei. Simultaneous image classification and annotation. In *Proc. CVPR*, 2009.
- [139] G. Wang, Y. Zhang, and L. Fei-Fei. Using dependent regions for object categorization in a generative framework. In *CVPR*, 2006.
- [140] M. Weber, M. Welling, and P. Perona. Unsupervised learning of models for recognition. In *ECCV*, 2000.
- [141] J. Winn, A. Criminisi, and T. Minka. Object categorization by learned universal visual dictionary. In *CVPR*, 2005.

- [142] John Winn and Christopher M. Bishop. Variational message passing. *J. Mach. Learn. Res.*, 6:661–694, 2004.
- [143] K. Yanai and K. Barnard. Probabilistic web image gathering. *ACM SIGMM workshop on Multimedia information retrieval*, pages 57–64, 2005.
- [144] Jianchao Yang, Kai Yu, Yihong Gong, and Thomas Huang. Linear spatial pyramid matching using sparse coding for image classification. In *CVPR*, 2009.
- [145] Z-Y. Yao, X. Yang, and S-C. Zhu. Introduction to a large scale general purpose groundtruth dataset: methodology, annotation tool, and benchmarks. In *6th Int'l Conf on EMMCVPR*, 2007.
- [146] H. Zhang, A. Berg, M. Maire, and J. Malik. Svm-knn: Discriminative nearest neighbor classification for visual category recognition. *Proc. CVPR*, 2006.
- [147] Q. Zhang, S.A. Goldman, W. Yu, and J.E. Fritts. Content-based image retrieval using multiple-instance learning. In *MACHINE LEARNING-INTERNATIONAL WORKSHOP THEN CONFERENCE-*, pages 682–689. Citeseer, 2002.
- [148] X.S. Zhou and TS Huang. Unifying keywords and visual contents in image retrieval. *Multimedia, IEEE*, 9(2):23–33, 2002.
- [149] L. Zhu, Y. Chen, and A. Yuille. Unsupervised learning of a probabilistic grammar for object detection and parsing. *Advances in neural information processing systems*, 19:1617, 2007.
- [150] X. Zhu. Semi-supervised learning literature survey. *Computer Science, University of Wisconsin-Madison*, 2006.