Linear Distance Metric Learning for Large-scale Generic Image Recognition

（線形距離計量学習による大規模一般画像認識）

Hideki Nakayama

中山 英樹

Graduate School of Information Science and Technology

The University of Tokyo

A thesis submitted for the degree of

Doctor of Philosophy

I would like to dedicate this thesis to my beloved family.

Acknowledgements

This thesis would not have been possible without the help of many people. My deepest appreciation goes to Prof. Yasuo Kuniyoshi who has provided many insightful comments and continuous encouragement. His philosophy has always inspired and motivated my work. I am proud being able to finish my Ph.D. work under his supervision.

I am also deeply grateful to Prof. Tatsuya Harada, who has advised me not only about my work, but about the wider world as well. He is an energetic and warm-hearted person. I owe a great deal of my life at ISI to him.

I would also like to express my gratitude to Prof. Yoichi Sato, Prof. Masayuki Inaba, and Prof. Taketoshi Mori for their many constructive comments and discussions that helped refine my work.

I appreciate the comments and support from Prof. Nobuyuki Otsu. Much of my work and knowledge are based on the advice he gave while at The University of Tokyo.

Finally, I would like to thank my family for their sincere love and encouragement over many years.

Abstract

Generic image recognition is a technique that enables computers to recognize unconstrained real-world images and describe their content in a natural language. It is known to be an extremely difficult problem due to the wide variety of targets and the ambiguity of the task. The key to realizing versatile and high performance generic image recognition is statistical machine learning using a large number of examples. However, since previous methods lack scalability with respect to the number of training samples, hitherto it has been practically impossible to utilize a large-scale image corpus for training and recognition.

In this thesis, we develop a scalable and accurate generic image recognition (image annotation) algorithm. To perform accurate image annotation, the semantic gap, that is, the gap between low-level image features and high-level meanings, need to be relaxed. The following two processes are essential in tackling this problem.

1. Extracting diverse and expressive image features.
2. Learning distance metrics between samples.

To realize a scalable system, it is extremely important to consider the compatibility of these processes. For large-scale problems, it is desirable that the complexity of training is linear in the number of training samples. Therefore, to learn a discriminative distance metric, we focus on canonical correlation analysis (CCA), a technique for bimodal dimensionality compression. By exploiting the probabilistic structure of CCA, we derive a theoretically optimal distance metric, called the canonical contextual distance (CCD). Image annotation based on CCD is shown to achieve comparable performance to state-of-the-art works with lower computational costs for learning and recognition.

Moreover, to use CCD efficiently, image features should be embedded in a Euclidean space. Specifically, the inner products in the feature space should appropriately reflect the similarity of features in terms of a generative process. Therefore, we develop a new framework to extract powerful

image features that satisfy this requirement. We propose the global Gaussian approach, in which we model the distribution of local features in an image with a single Gaussian. Further, using the technique of information geometry, we approximately code a Gaussian into a feature vector, which we call the generalized local correlation (GLC).

Using a combination of CCD and GLC, we can realize a scalable high-performance image annotation system. We show the effectiveness of our system using a large-scale dataset consisting of twelve million web images.

Contents

Contents v

List of Figures ix

List of Tables xiii

1 Introduction 1
- 1.1 Background . . . 1
- 1.2 Objective . . . 2
- 1.3 Structure of the Thesis . . . 4

2 Outline of the Image Recognition Method 7
- 2.1 History and Current Status of Image Recognition . . . 7
- 2.2 Generic Image Recognition . . . 10
  - 2.2.1 Generic Image Recognition vs. Specific Image Recognition . . . 10
  - 2.2.2 Semantic Gap . . . 11
  - 2.2.3 Tasks of Generic Image Recognition . . . 13
- 2.3 Training Image Corpus . . . 14
  - 2.3.1 Small Datasets . . . 14
  - 2.3.2 Large Datasets . . . 15
- 2.4 Designing the Image Recognition Method . . . 20
  - 2.4.1 Tackling the Image Annotation Task . . . 21
  - 2.4.2 Scalability for a Large-scale Training Corpus . . . 21

3 Related Work in Image Annotation 23
- 3.1 Previous Work . . . 23
  - 3.1.1 Region-based Generative Model . . . 23
  - 3.1.2 Local Patch Based Generative Model . . . 26
  - 3.1.3 Binary Classification Approach . . . 26
  - 3.1.4 Graph-based Approach . . . 28
  - 3.1.5 Regression Approach . . . 29

3.1.6 Topic Model Approach . . . 29
3.1.7 Non-parametric Approach . . . 31
3.1.8 Summary . . . 32
3.2 Bridging the Semantic Gap for Non-parametric Image Annotation . . . 34
3.2.1 Distance Metric Learning . . . 34
3.2.2 Bimodal Dimensionality Reduction Methods . . . 36

4 Development of a Scalable Image Annotation Method **41**
4.1 Non-parametric Image Annotation . . . 41
4.1.1 *k*-Nearest Neighbor Classification . . . 41
4.1.2 MAP Classification . . . 42
4.2 Distance Metric Learning Using Probabilistic Canonical Correlation Analysis . . . 43
4.2.1 Canonical Correlation Analysis . . . 43
4.2.2 Probabilistic Canonical Correlation Analysis . . . 43
4.2.3 Proposed Method: Canonical Contextual Distance . . . 45
4.3 Embedding Non-linear Metrics of Image Features . . . 47
4.4 Label Features . . . 48
4.5 Application to Keyword-based Image Retrieval . . . 48
4.6 Discussion . . . 49
4.6.1 Summary of Proposed Methods . . . 49
4.6.2 Relation to Other Methods Based on Topic Models . . . 50

5 Evaluation of Image Annotation Method **53**
5.1 Datasets . . . 53
5.2 Basic Experiment . . . 54
5.2.1 Image Features . . . 54
5.2.2 Experimental Setup . . . 55
5.2.3 Experimental Results . . . 56
5.3 Comparison with Previous Research . . . 66
5.3.1 Image Features . . . 66
5.3.2 Experimental Results . . . 67
5.3.3 Computational Costs . . . 70
5.4 Discussion . . . 71

6 Development of Image Feature Extraction Scheme **73**
6.1 Coding Global Image Features Using Local Feature Distributions . . . 73
6.2 Related Work . . . 74
6.2.1 Non-parametric Method . . . 74
6.2.2 Gaussian Mixtures . . . 75
6.2.3 Bag-of-Visual-Words . . . 75

6.2.4 Covariance Descriptor . . . 76
6.3 Proposed Method: Global Gaussian Approach . . . 76
6.3.1 Coding Gaussian with Information Geometry . . . 76
6.3.2 Brief Summary of Information Geometry . . . 77
6.3.3 Gaussian Embedding Coordinates: Generalized Local Correlation (GLC) . . . 78
6.3.4 Kernel Functions . . . 79
6.4 Rigorous Evaluation using Kernel Machines . . . 81
6.4.1 Datasets . . . 81
6.4.2 Classification Methods . . . 82
6.4.3 Experimental Setup . . . 84
6.4.4 Experimental Results . . . 85
6.4.5 Discussion . . . 90
6.5 Scalable Approach Using GLC and Linear Methods . . . 90
6.5.1 Compressing GLC . . . 90
6.5.2 Datasets . . . 92
6.5.3 Experimental Setup . . . 92
6.5.4 Experimental Results . . . 94

7 Evaluation of Large-scale Image Annotation 105
7.1 Dataset Construction (Flickr12M) . . . 105
7.1.1 Downloading Samples . . . 105
7.1.2 Statistics of Flickr12M Dataset . . . 107
7.2 Preliminary Experiments . . . 110
7.2.1 Image Features . . . 110
7.2.2 Evaluation Protocol . . . 111
7.2.3 Experimental Results . . . 111
7.3 Large-scale Experiments . . . 113
7.3.1 Quantitative Evaluation . . . 113
7.3.2 Qualitative Effect of Large-scale Data . . . 113

8 Conclusion and Future Works 121
8.1 Conclusion . . . 121
8.2 Unsolved Problems . . . 124
8.3 Future Works . . . 125

Appendix A: Evaluation Protocol for Image Annotation and Retrieval 127

Appendix B: Kernel Principal Component Analysis 131

Appendix C: Details of HLAC Features 135

Appendix D: Experimental Results for Subsets of Flickr12M 137

Appendix E: Hashing-based Rapid Annotation 155

References 169

Publications 191

List of Figures

1.1 Illustration of generic image recognition. Several meanings (symbols) can be extracted from a single image. . . . 2
1.2 Appearance changes due to various real-world conditions. . . . 3
1.3 A variety of "chairs". Credit: Li Fei-Fei *et al.* CVPR'07 object recognition tutorial slides. . . . 3
1.4 Structure of the thesis. . . . 5

2.1 Three levels of variance in generic images. . . . 10
2.2 (a) A query image. (b) The closest image in terms of the color histogram. Credit: Jing et al. [90]. . . . 11
2.3 Various tasks of generic image recognition. . . . 14
2.4 Three standard benchmarks for image auto-annotation. Top: Corel5K [50]. Middle: IAPR-TC12 [125]. Bottom: ESP Game [125]. . . . 16
2.5 Example images from Caltech-101 dataset [55; 56]. . . . 16
2.6 Example images from Caltech-256 dataset [69]. . . . 17

3.1 Graphical model of the CRM and MBRM. Credit: Feng *et al.* [59]. . . . 25
3.2 Illustration of SML. Credit: Carneiro *et al.* [29]. . . . 27
3.3 A topic model for image annotation. . . . 30

4.1 Graphical model of PCCA. . . . 44
4.2 Illustration of canonical contextual distances. Estimation of distance between a query and training sample: (a) from the x-view only (CCD1); and (b) considering both the x- and y-views (CCD2). . . . 46
4.3 (a): Typical topic model approach. (b), (c): Approaches to the annotation problem using PCCA. . . . 50

5.1 Results for the Corel5K dataset (1000-dimensional SIFT BoVW). Methods are compared using different features with designated dimensionality (d). For each entry, the left set of bars corresponds to normal linear methods, while the right set corresponds to those with KPCA embedding. . . . 59

5.2 Results for the IAPR-TC12 dataset (1000-dimensional SIFT BoVW). 59
5.3 Results for the Corel5K dataset (100-dimensional hue BoVW). 60
5.4 Results for the IAPR-TC12 dataset (100-dimensional hue BoVW). . . 60
5.5 Results for the Corel5K dataset (4096-dimensional HSV color histogram). 61
5.6 Results for the IAPR-TC12 dataset (4096-dimensional HSV color histogram). 61
5.7 Results for the Corel5K dataset (512-dimensional GIST). 62
5.8 Results for the IAPR-TC12 dataset (512-dimensinal GIST). 62
5.9 Results for the Corel5K dataset (2956-dimensional HLAC). Only linear methods are compared. 63
5.10 Results for the IAPR-TC12 dataset (2956-dimensional HLAC). 63
5.11 Results for the NUS-WIDE dataset (edge histogram). Methods are compared using different features with designated dimensionality (d). 64
5.12 Results for the NUS-WIDE dataset (color correlogram). 64
5.13 Results for the NUS-WIDE dataset (grid color moment). 65
5.14 Results for the NUS-WIDE dataset (SIFT BoVW). 65
5.15 Annotation performance (F-measure) with a varying number of base samples for kernel PCA embedding. 69

6.1 Images from benchmark datasets. Top left: LSP15 [106]. Bottom left: 8-sports [111]. Right: Indoor67 [156]. 82
6.2 Merging the global Gaussian and BoVW approaches for use with the LSP15 dataset. κ is the parameter for weighting the kernels (Eq. 6.24). 88
6.3 Merging the global Gaussian and BoVW approaches for use with the 8-sports dataset. κ is the parameter for weighting the kernels (Eq. 6.24). 88
6.4 Sample images from the OT8 dataset. 92
6.5 Effect of sampling density on performance ($P = 16$, $m = 30$). 97
6.6 Effect of the dimensionality of PCA compression ($P = 16$, $M = 5$). . . 97
6.7 Effect of the scale parameter of the SIFT-descriptor ($m = 30$, $M = 5$). 98
6.8 Effect of the weight parameter using at most the 2nd layer ($P = 16$, $m = 30$, $M = 5$, $\gamma = 5.0e-06$). 99
6.9 Effect of the weight parameter using at most the 3rd layer ($P = 16$, $m = 30$, $M = 5$, $\gamma = 5.0e-06$). 100
6.10 Results using different dimensionality compression methods ($P = 16$, $m = 30$, $M = 5$). We used two different projection matrices (one from OT8 and the other from Caltech-101), and random sampling. 101

7.1 Examples of Flickr data: images and corresponding social tags. 106

7.2 Examples of near-duplicate images in the Flickr dataset. Each row corresponds to a duplicate set. These images are annotated with the same social tags. 108
7.3 Word frequencies in the Flickr12M dataset. 109
7.4 Annotation performance of each feature with CCD2 (<1.6M samples). 112
7.5 Annotation performance of SURF BoVW-sqrt, SURF GLC, and HLAC features (<12.3M samples). 114
7.6 Annotation performance of combinations of SURF BoVW-sqrt, SURF GLC, and HLAC features (<12.3M samples). 115
7.7 Comparison of annotation performance with CCD2 (<12.3M samples). 116
7.8 (1/3) Example of annotation using a varying number of training samples. Correct annotations are written in red. For each case, the 25 nearest images are shown. 117
7.9 (2/3) Example of annotation using a varying number of training samples. Correct annotations are written in red. For each case, the 25 nearest images are shown. 118
7.10 (3/3) Example of annotation using a varying number of training samples. Correct annotations are written in red. For each case, the 25 nearest images are shown. 119

1 Annotation scores for the Corel5K dataset with varying numbers of output words. The proposed method (linear) + HLAC feature is used. 128
2 Illustration of "car" retrieval results. Correct images are ranked 2nd, 5th, and 7th, respectively. 129

3 Mask patterns of at most the first order Color-HLAC features. 136

4 F-measures of **Tiny image** features for the 100K, 200K, and 400K subsets. 138
5 F-measures of **Tiny image** features for the 800K and 1.6M subsets. . 139
6 F-measures of the **RGB color histogram** for the 100K, 200K, and 400K subsets. 140
7 F-measures of the **RGB color histogram** for the 800K and 1.6M subsets. 141
8 F-measures of **GIST** features for the 100K, 200K, and 400K subsets. . 142
9 F-measures of **GIST** features for the 800K and 1.6M subsets. 143
10 F-measures of **HLAC** features for the 100K, 200K, and 400K subsets. 144
11 F-measures of **HLAC** features for the 800K and 1.6M subsets. 145
12 F-measures of **SURF GLC** features for the 100K, 200K, and 400K subsets. 146
13 F-measures of **SURF GLC** features for the 800K and 1.6M subsets. . 147
14 F-measures of **BoVW** features for the 100K, 200K, and 400K subsets. 148

15 F-measures of **BoVW** features for the 800K and 1.6M subsets. 149
16 F-measures of **BoVW-sqrt** features for the 100K, 200K, and 400K subsets. 150
17 F-measures of **BoVW-sqrt** features for the 800K and 1.6M subsets. . 151
18 F-measures of **RGB-SURF GLC** features for the 100K, 200K, and 400K subsets. 152
19 F-measures of **RGB-SURF GLC** features for the 800K and 1.6M subsets. 153

20 Retrieval performance with a varying number of bits for the LabelMe dataset. 160
21 Retrieval performance as a function of retrieved images for the LabelMe dataset. 160
22 Examples of retrieved images (15 neighbors) for the LabelMe dataset. 161
23 Retrieval performance with a varying number of bits for the Flickr12M dataset. 163
24 Retrieval performance as a function of retrieved images for the Flickr12M dataset. 163
25 Examples of retrieved images (15 neighbors) for the Flickr12M dataset. 164
26 Annotation scores (F_W) with a varying number of bits for the full Flickr12M dataset. 167
27 Annotation scores (F_I) with a varying number of bits for the full Flickr12M dataset. 167
28 Annotation scores (F_W) with a varying amount of memory (MB). . . . 168
29 Annotation scores (F_I) with a varying amount of memory (MB). . . . 168

List of Tables

1.1 Computational complexity of a non-linear SVM. N is the number of training samples. . . . 3

3.1 Performance of previous works using Corel5K. . . . 33
3.2 Relationship between dimensionality reduction methods. All methods can be interpreted as special cases of PLS. . . . 39
3.3 Computational complexity of PCA, PLS, and CCA based methods: (1) calculating covariances, (2) solving eigenvalue problems, and (3) projecting training samples using the learned metric. . . . 39

5.1 Statistics of the training sets of the benchmarks. . . . 54
5.2 Computation times for training the system on the NUS-WIDE dataset using each method[s]. We found that the differences in running times between PCA and PCAW, and between CCA and CCD are negligible for a small d. . . . 58
5.3 Performance comparison using Corel5K. . . . 68
5.4 Performance comparison using IAPR-TC12. . . . 69
5.5 Performance comparison using ESP game dataset. . . . 69
5.6 Comparison of annotation performance (F-measure) using TagProp. . . . 70
5.7 Comparison of computational costs against the number of samples. N is the number of whole training samples, while n_K is the number of those used for kernelization. . . . 71

6.1 Summary of previous work and our work from the viewpoint of local feature statistics. . . . 74
6.2 Basic results of the global Gaussian approach with the LSP15 and 8-sports datasets using different kernels (%). No spatial information is used here. . . . 86
6.3 Performance comparison with spatial information for LSP15 (%). The SURF descriptor is used. . . . 86
6.4 Performance comparison with spatial information for the 8-sports dataset (%). The SIFT descriptor is used. . . . 87

6.5 Performance of the global Gaussian, BoVW, and combined approach (%). An $L = 2$ spatial pyramid is implemented. Kernel PDA is used for classification. The SURF descriptor is used for LSP15, while the SIFT descriptor is used for the 8-sports dataset. 87
6.6 Performance comparison with previous work (%). For our method, an $L = 2$ spatial pyramid is implemented, and kernel PDA is used for classification. We used the SURF descriptor for LSP15 and Indoor67, and the SIFT descriptor for the 8-sports dataset. 89
6.7 Baseline performance for OT8 (%) using GLC in different types. Classification is conducted via PDA and SVM. Regarding the results for the SVM, the plain number indicates the classification score using a linear kernel, while the italic number in parenthesis indicates that using the RBF kernel. The best score for each descriptor is shown in bold. . . . 95
6.8 Classification performance of GLC and bag-of-visual-words (BoVW) for OT8 (%). We implement BoVW with 200, 500, 1000, and 1500 visual words. 96
6.9 Comparison of the performance using two scene datasets and Caltech-101 (%). 103

7.1 The most popular 145 tags on Flickr. These tags were used for the initial download. 107
7.2 Statistics of the Flickr12M dataset. 108
7.3 Word frequencies in Flickr12M. 108
7.4 Most frequently used words in Flickr12M. 109

1 Retrieval time per image for Flickr12M (s) using a single CPU. 165
2 Computation time for training with the Flickr12M dataset using an 8-core desktop machine. 165

Chapter 1

Introduction

1.1 Background

Generic image recognition[1] (Figure 1.1) is a technique that allows computers to recognize unconstrained real-world images and to describe the content thereof in a natural language [154; 229]. As humans also recognize objects and scenes from visual information to decide actions, generic image recognition is one of the most essential abilities for real-world intelligent systems. Since generic image recognition is a valuable technique in both science and engineering, it has drawn the attention of many researchers in a variety of fields.

A scientific interest is to realize and understand the image recognition ability of humans. This ability has been studied enthusiastically for decades in many areas including cognitive psychology and brain science. An approach from Computer Science can also provide significant insight.

Moreover, because generic image recognition tackles the symbol grounding problem, which is a fundamental problem in artificial intelligence, its commercial impact would be immense. For example, real-world recognition systems for robots and automobiles would become straightforward applications. Moreover, it could be used for lifelogs, surveillance systems, and web image search engines.

However, despite its long history, generic image recognition has not yet been realized, and is still regarded as one of the ultimate goals in computer vision. The difficulty of generic image recognition stems from the diversity of the images and target objects. Even at an instance level, the appearances of images change widely according to viewpoints, illumination, and occlusion conditions (Figure 1.2). Furthermore, generic objects include a variety of instances, resulting in more diversity of appearance. For example, although the samples in Figure 1.3 are all "chairs", their colors

[1] Also called generic object recognition. "Generic objects" include not only rigid objects, but also abstract symbols such as scenes or adjectives.

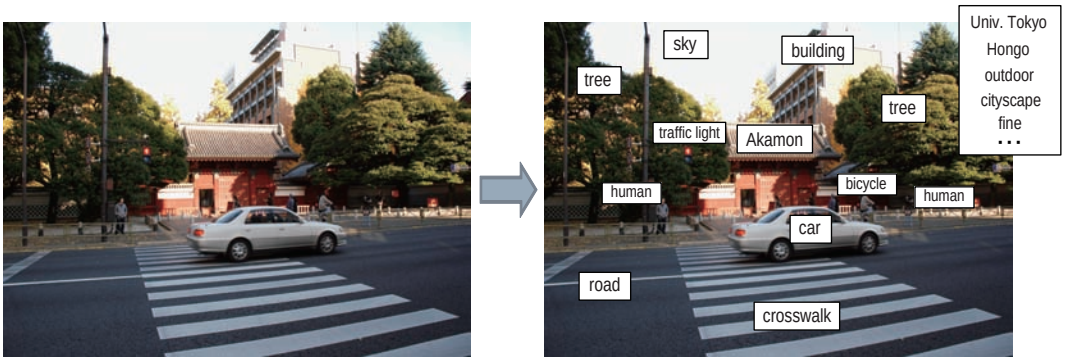


Figure 1.1: Illustration of generic image recognition. Several meanings (symbols) can be extracted from a single image.

and shapes differ considerably. Moreover, the goal of generic image recognition is to recognize various generic objects in the world as humans do. A psychological study showed that humans can recognize tens of thousands of categories using only visual information [16]. To do this on a computer, we need to deal with even more diversity in image appearance.

As discussed in detail in Chapter 2, it has been shown that it is difficult to design a prototype explicitly for generic image recognition due to the diversity and ambiguity of the process. Consequently, the statistical machine learning approach is now flourishing in this area. In particular, learning with a huge number of examples is thought to be the most promising methodology to realize generic image recognition. However, since previous methods generally lack scalability, it has been practically impossible to train a system with a large-scale image corpus. This is a severe bottleneck in generic image recognition techniques. For example, Table 1.1 gives the complexity of a nonlinear SVM, the standard learning method for generic image recognition. It is clear that computational costs for training increase dramatically with the number of training samples. Furthermore, since large scale data rarely fits in the available memory and standard methods based on gradient descent require storage access during optimization, this leads to extremely slow training. Therefore, large-scale generic image recognition is not just a matter of the size of the dataset, but rather a new research field that requires a qualitative breakthrough.

1.2 Objective

In this thesis, we develop a scalable and accurate generic image recognition (image annotation) algorithm. Specifically, we first develop a statistical machine learning method to learn discriminative distance metrics between samples, which we call the

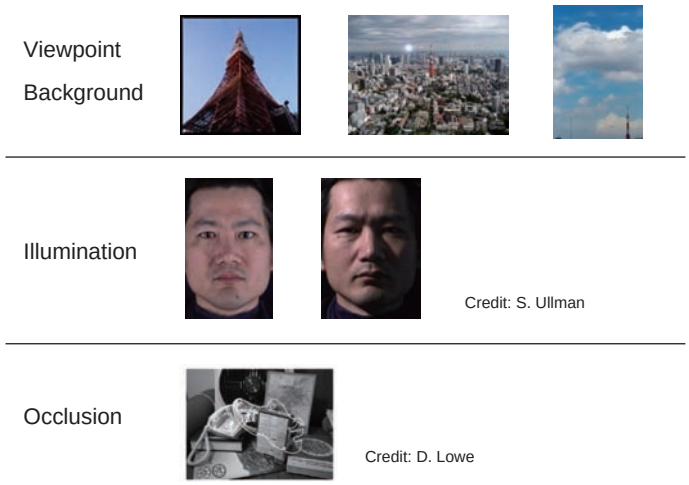


Figure 1.2: Appearance changes due to various real-world conditions.

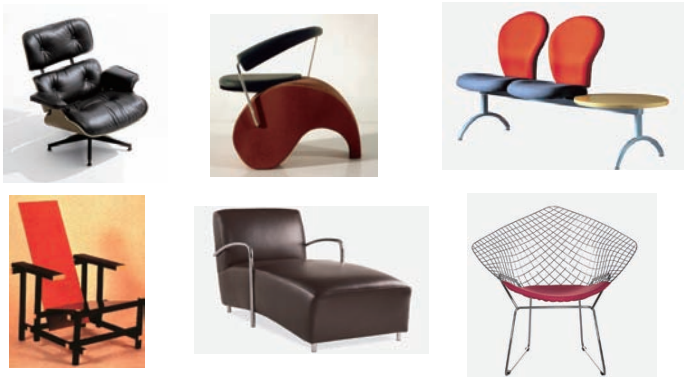


Figure 1.3: A variety of "chairs". Credit: Li Fei-Fei *et al.* CVPR'07 object recognition tutorial slides.

Table 1.1: Computational complexity of a non-linear SVM. N is the number of training samples.

	Complexity	Memory
Training	$O(N^2) \sim O(N^3)$	$O(N^2)$
Recognition	$O(N)$	$O(N)$

canonical contextual distance (CCD). Further, to extract image features, we develop a new generic framework that is compatible with the above mentioned method.

1.3 Structure of the Thesis

The structure of this thesis is as follows (Figure 1.4). The background and objective of the thesis is given in this chapter. In Chapter 2, we survey the history and current status of generic image recognition, and present the design of our system. We start to tackle the image annotation problem. In Chapter 3, we review related research on image annotation. In Chapter 4, we develop our image annotation method based on scalable distance metric learning. In Chapter 5, we evaluate our image annotation method using standard datasets. In Chapter 6, we develop a framework to extract image features that is compatible with our image annotation method. The combination of these two technologies leads to a scalable and accurate image recognition system, our final goal. In Chapter 7, we evaluate the proposed system using a large-scale web image dataset and show its effectiveness. Finally, in Chapter 8, we conclude the thesis and present our future works.

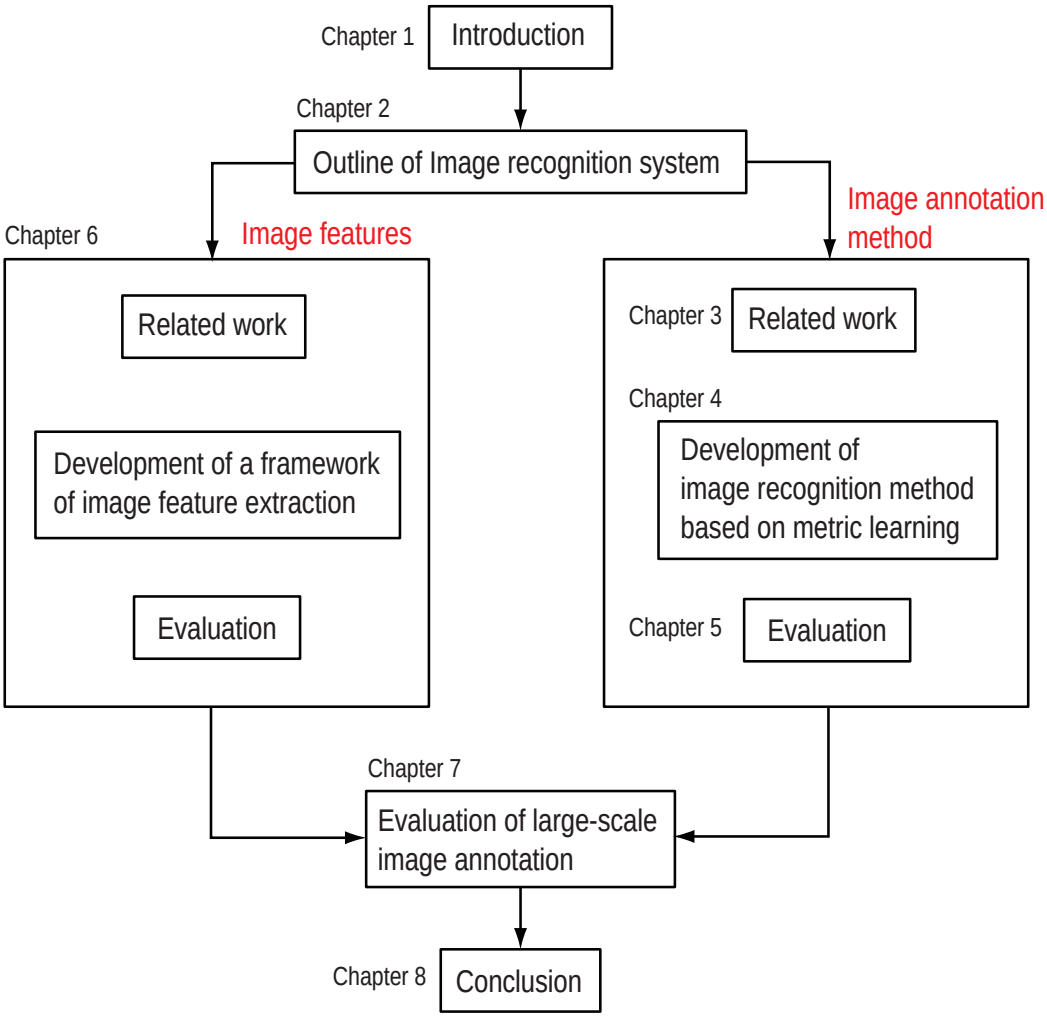


Figure 1.4: Structure of the thesis.

Chapter 2

Outline of the Image Recognition Method

2.1 History and Current Status of Image Recognition

While humans are said to be able to recognize tens of thousands of visual categories [16], it is extremely difficult for computers to recognize even one object category. Computerized image recognition has attracted the attention of many researchers, and having originated in the 1950s, has a history of more than half a century.

In the 1950s, research began with recognizing two-dimensional patterns such as characters and fingerprints. During this era, a statistical pattern recognition approach was mainly used. Several methods designed geometrically invariant features such as moment features [83]. The statistical approach once again flourished in the 1990s, despite its place having been taken for decades by a model based approach.

In the mid 1950s, an "artificial intelligence" paradigm was established by Marvin Minsky and John McCarthy, making the statistical approach obsolete. This new paradigm began by thoroughly simplifying world descriptions to adequately model the cognitive ability of humans using mathematical tools. The "blocks world" [158] was the earliest example in computer vision. In this world, objects were limited to polyhedrons, and a uniform background was assumed. The objective was to recover three-dimensional object alignment from a two-dimensional image taken from an arbitrary viewpoint. Later, this idea evolved to line drawing interpretation [73] to handle curved surfaces. However, in the first place, it was extremely difficult to extract line drawings reliably from real-world images. Consequently, a generalized cylinder [17] was used to decompose real-world objects [26; 216]. Nevertheless, it was difficult to extract components by bottom-up segmentation. This is a fundamental problem of real image recognition, and is still unsolved today despite much research effort.

Many methods based on generalized cylinders are classified as model-based image

recognition approaches [155]. In a model-based approach, three-dimensional geometric models of target objects are pre-defined. Recognition is carried out by matching a query image with the models. However, because shape models are used directly for recognition, this approach can recognize only a specific object. To recognize a generic object category, we need to prepare a sufficient number of models covering the diversity of the category, which is unrealistic in real problems. Moreover, it cannot deal with generic object categories that have no explicit shapes, such as "sea" and "street". A further example of a knowledge based approach, is the image expert system [39]. However, none of these methods was successful because of the problems mentioned above.

As interest in model-based recognition gradually faded, statistical approaches were once again studied in the early 1990s. As background, the significant progress in computer hardware made it possible for anyone to exploit statistical analysis methods. Moreover, many powerful machine learning methods such as the SVM were proposed during this era. The core idea is an "appearance-based" approach, in which recognition is conducted directly using 2D images without restoring 3D alignment. This approach has been the mainstream technique in generic image recognition till today. Whereas models are designed by humans in a model-based approach, in a statistical approach, distinctive features are automatically selected through the learning from training examples. The eigen face method [180] is a representative example of this approach. This method compresses raw image vectors using eigen subspaces and then uses the compressed vectors as the features. The parametric eigen subspace method [136] is an extension of the eigen face method to generic objects. In addition, during this era, several researchers developed low-level image features that represent statistical properties of images, such as color and texture. Color histograms [161; 175] are typical early works. Since color histograms are simple features that can be extracted quickly, they have been widely used for many tasks including content-based image retrieval (CBIR) [171].

A problem with the methods from the 1990s was that they were sensitive to occlusion and changes in scale and orientation, because they mainly used global image features. In the 2000s, this problem was relaxed to some extent, owing to the huge success of a local feature based approach. Local features describe properties of a small local area surrounding a certain point (keypoint). In general, local feature descriptors are designed so that they are invariant or robust to rotation, illumination, and scale changes. Scale-invariant feature transform (SIFT) [120; 121] is a representative example of local feature descriptors. Though keypoint detection methods, especially corner detection methods, had been studied for a long time [11; 75], SIFT was the first method that proposed a sophisticated pipeline of keypoint detection, normalization, feature description, and sub-pixel localization. SIFT enabled robust keypoint matching in real images. Moreover, local feature based image matching is relatively tolerant to occlusions. As a result of these advantages, SIFT has been used in various areas of computer

vision and has undergone continuous improvement as a result of the many successive works [94; 174]. Moreover, besides SIFT, many other local feature descriptors have been proposed [10; 28; 185].

Originally, local features were designed for specific image recognition[1], which is a technique to recognize a single instance. Nevertheless, it has been proved that local features are also effective for generic image recognition. First, a part-based approach was proposed. This approach models an object using local characteristics and their spatial alignment. The constellation model [54; 61], which is a representative method, exploits several local features. This method learns rough spatial alignment of representative local features for each object category, and attempts to perform robust recognition against deformation. However, in generic image recognition, it is difficult to express an image stably using only several local features. This is because keypoints are selected according to their saliency and do not necessarily capture essential information for recognition. Furthermore, the computational cost of training the constellation model is immense because it optimizes spatial alignment of local features in a brute-force manner.

On the contrary, many people find it effective to represent images using the statistical properties of many local features without spatial information. The most well-known method is the bag-of-visual-words (BoVW) [40], which is based on vector quantization. This is an application to computer vision, of the bag-of-words (BoW) [126] method, a technique developed for the field of natural language processing. First, thousands of local features are extracted from each image. By clustering all local features in a training dataset, we obtain some centroids (visual words). Finally, each image is represented by a visual word histogram. BoVW based image recognition shows surprisingly high performance in many tasks, and has been studied intensively. Although one of the reasons for its success is the fact that spatial information of each local feature is discarded, rough spatial alignment of an image is thought still to be effective for recognition[2]. Therefore, spatial pyramid matching (SPM) [106] exploited spatial information by hierarchically dividing images and matching BoVW histograms in each region. Despite its simplicity, SPM substantially improved the performance of the original BoVW. Currently, SIFT BoVW + SPM + SVM is the de-facto standard algorithm for generic image recognition.

Thereafter, studies on generic image recognition focused on two main problems. The first of these is how to efficiently exploit a distribution of local features in an image. This fundamental problem encompasses BoVW. We address this problem in Chapter 6. The second problem is how to combine multiple features obtained by different descriptors. Multiple kernel learning [103] is a typical approach to this problem,

[1] Also called specific object recognition. We discuss the difference between generic image recognition and specific image recognition in the next section.

[2] For example, "sky" tends to appear at the top, while "sea" tends to appear at the bottom, etc.

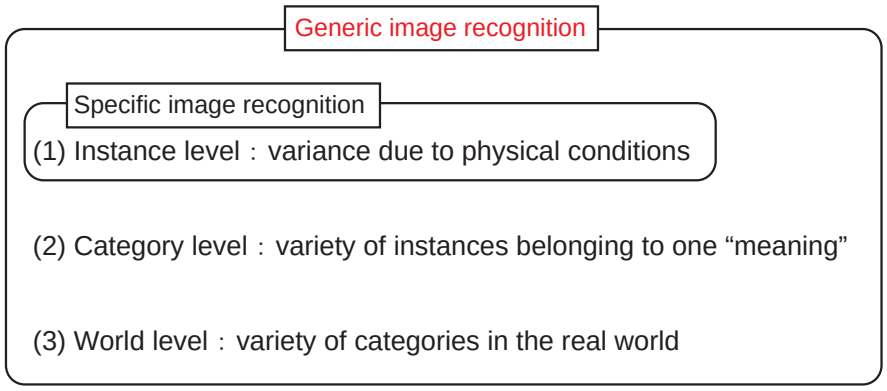


Figure 2.1: Three levels of variance in generic images.

and has been studied at length. Although these methods have continuously been studied and improved, they were basically well established in the 2000s. Now, in the 2010s, research has reached the next phase, in which other resources and methods are integrated with these basic tools. For example, context information represented by multiple objects [81; 179], hierarchical structure of categories [45; 70], discovering unseen categories [108], exploiting external knowledge [46; 99; 177] such as the WordNet [58], and interactive learning [170] are seen to be important topics.

2.2 Generic Image Recognition

2.2.1 Generic Image Recognition vs. Specific Image Recognition

To illustrate an essential difficulty of generic image recognition, we first describe specific image recognition, a contrasting paradigm. Specific image recognition implies instance-level recognition. Take “car” image recognition as an example. In generic image recognition the system recognizes various car images including trucks and buses as “cars”, whereas specific image recognition judges only “whether this object is a Toyota Corolla or not.”

Their fundamental difference is explained with reference to the variance in image appearance. As described in the introduction, this can be roughly divided into three levels (Figure 2.1). Specific image recognition focuses on problem (1) (Figure 1.2). Although many factors need to be considered, such as viewpoint and illumination changes or occlusions, all these factors are basically due to certain physical constraints. For these kinds of appearance changes, we can design robust local features, such as SIFT, in a top-down manner. Thus, specific image recognition has been making steady progress using local feature based approaches. During the past few

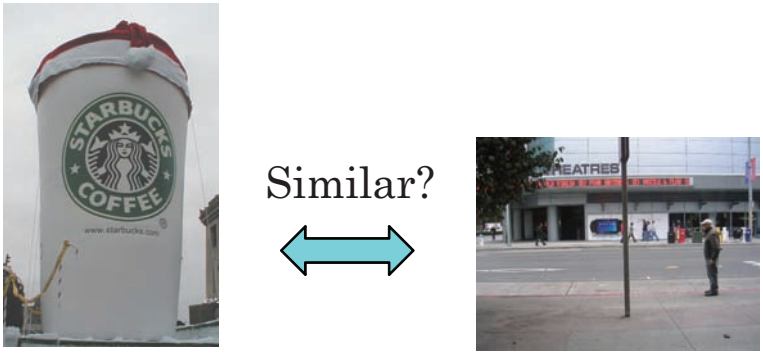


Figure 2.2: (a) A query image. (b) The closest image in terms of the color histogram. Credit: Jing et al. [90].

years, near-commercial applications such as Google Goggles[1] have appeared.

In addition to this, generic image recognition needs to consider problems (2) and (3). Of these, problem (2) is fundamentally the more difficult one. Take Figure 1.3 as an example. Although humans recognize that these images are all "chair" images, their appearances differ substantially. Because the meaning of images depends on our experience, it cannot always be explained by physical properties such as color and shape. The gap between low-level image features and high-level meanings is called the semantic gap [171], which characterizes the generic image recognition problem.

2.2.2 Semantic Gap

Take the two images in Figure 2.2 as an example. While their meanings are entirely different, they are similar in terms of some low-level features such as color histograms. This means that distinguishing the meanings from their appearances is computationally difficult. This is a typical illustration of the semantic gap problem. To relax the semantic gap, the following two processes are important.

Extracting expressive image features

First, the system needs to have high performance in distinguishing samples at an instance-level. For example, to distinguish the two images in Figure 2.2, shape features such as edge histograms would be necessary in addition to color features. Needless to say, there are numerous objects and scenes in the real world. Therefore, for

[1] http://www.google.com/mobile/goggles/

versatile image recognition systems, we should exploit as many features as possible that illustrate different properties of an image. Consequently, a representation of an image becomes rather high-dimensional.

Discriminative distance metric learning

Generic image recognition estimates the distance to a concept (class), rather than a specific instance. Therefore, we need to consider intra-class variance. Since the variance is not always related to physical conditions, we cannot design invariant features in a top-down manner. Take the "chair" images in Figure 1.3 as an example. It is expected that the flat shape of the seat would be critical in discriminating chairs. However, other features such as color are entirely different for each example. Therefore, we cannot estimate the semantic distance to the "chair" category merely by matching the visual features of examples, because semantically irrelevant features could disturb the inference. This is the essential problem that characterizes generic image recognition.

One naive approach is to exploit a sufficiently large number of training examples that can fill the image feature space. For example, using as many kinds of "chair" examples as possible, the possibility of finding a visually similar example increases for an arbitrary input image. This means that visual distance approaches semantic distance as the number of training examples grows. We can conduct recognition using simple non-parametric methods, such as k-nearest neighbor classification. Although a knowledge based approach once failed in the era of expert systems, we can now easily utilize an incomparably large amount of high-quality data because of the advances in web technologies. The promising effect of web-scale data has been shown in recent works [177; 197].

However, as previously mentioned, since image features are very high-dimensional, it is still unrealistic to fill the feature space generatively. Moreover, the system would suffer from the "curse of dimensionality problem", which states that finding neighbors in a high-dimensional space is rather difficult. Therefore, it is important to pre-select important features in a machine learning approach. Take the "chair" example once again. We first prepare some positive and negative examples of chairs. Applying a discriminative classifier (*e.g.* SVM) to these, a hyperplane for separating positive and negative examples is automatically aligned using important features for discrimination. As for the "chair" category in Figure 1.3, distance to the hyperplane would depend mainly on shape features, rather than color features. In other words, a one-dimensional new space is obtained, where distance is strongly related to semantic meanings compared to the original feature space. Thus, using a machine learning technique, we can obtain a new distance metric reflecting the semantic meanings of images.

2.2.3 Tasks of Generic Image Recognition

Generic image recognition is roughly divided into two tasks: (1) labeling a whole image, and (2) labeling each region in an image (Figure 2.3). In task (2), generally rigid objects, whose correspondence to image regions is apparent, are likely to be targeted. On the other hand, task (1) includes symbols that are not explicitly related to image regions such as abstract scenes, in addition to concrete objects. Also, whereas a training dataset for task (1) requires only images and corresponding labels, one for task (2) also requires manual segmentation of image regions for each label. Therefore, the cost of preparing datasets for task (2) is generally high.

Labeling a whole image

In this framework, the recognition system estimates only whole-image level correspondence with labels, rather than region level. Within, **image categorization,** is the task of assigning a single category label exclusively to one image. This is the most basic theme of generic image recognition that has been studied from the beginning. Since the evaluation methodology is clear, it has served as a testbed for developing basic tools such as image features and learning methods.

In contrast, **image annotation** is the task of assigning multiple labels to a single image. Annotation is a more generic problem that includes the categorization problem. Whereas during categorization, we can simply treat training samples without the targeted label as negative examples, we need to consider relationships between labels during annotation. As a result, annotation is a more difficult problem than categorization. Many approaches, as described in Chapter 3, have been proposed to tackle this problem.

Since both categorization and annotation techniques are currently well established for toy datasets, the interest of the community has moved to handling large-scale data [46; 177; 197] to realize practical systems. Moreover, since training data for these tasks are relatively easy to prepare owing to advances in web technologies, the number of studies on large-scale problems is rapidly increasing.

Labeling each region of an image

Object detection is the task of recognizing each object in an image and its region. Regions are roughly represented by a bounding box or convex polygon. Frontal face recognition, which is now implemented in many industrial applications, is a typical example. Basically, this can be done using a sliding window approach, in which the system scans an image with windows of different sizes and performs binary categorization within each window. In this sense, detection shares many techniques with categorization. In addition, there are many problems specific to the detection task, including non-maxima suppression and occlusion handling [47]. Moreover, since a

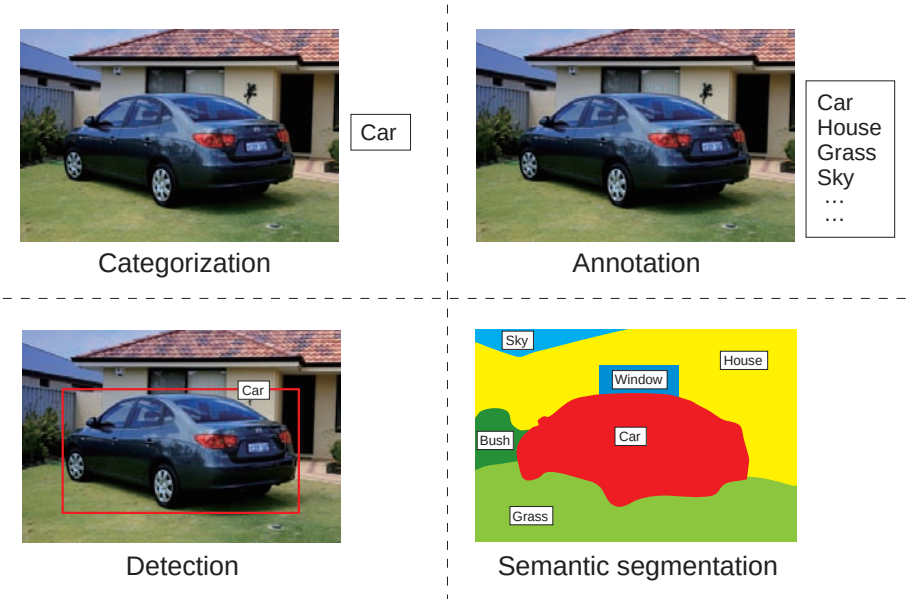


Figure 2.3: Various tasks of generic image recognition.

brute-force search of all windows is practically intractable, efficient search methods, such as the subwindow search [102] and Hough voting [109; 124], have been proposed.

Image segmentation is the task of pixel-level region estimation for each object. Here the term "segmentation" includes not only bottom-up region partitioning, but also semantic understanding. Recently, conditional random fields [101] have become the standard approach for this task [100; 168].

2.3 Training Image Corpus

2.3.1 Small Datasets

The difficulty of generic image recognition changes substantially depending on the selection of object categories and the nature of the images. Although researchers prepared their own datasets until the early 2000s, some standard benchmark datasets appeared as the research area became more popular. Ref. [153] summarizes the benchmark datasets published prior to 2006.

In the field of image annotation, the Corel5K dataset [50] has been the de-facto standard benchmark for a long time (Figure 2.4, top). This is a subset of the Corel Stock Photo Library published by Corel Inc., consisting of 80,000 images. Each image is manually labeled with several words so that it can be retrieved using keywords. Corel5K contains 5,000 pairs of image and labels from the library. Its dictionary com-

prises 371 words, which is relatively large considering the size of the dataset. Although Corel5K has been used in this area for a long time, it has been pointed out that this is an easy dataset because test images are very similar to the training ones. Therefore, in addition to Corel5K, recent works have used the IAPR-TC12 (Figure 2.4, middle) and ESP game (Figure 2.4, bottom) datasets, which were proposed by Makadia *et al.* [125] at ECCV 2008. IAPR-TC12 was originally used in ImageCLEF [1], a workshop for cross-lingual image retrieval, while the ESP game dataset is a subset of an image-label database obtained from an online image labeling game called the ESP collaborative image labeling task [189]. These datasets are described later.

For categorization tasks, the Caltech-101 dataset [55; 56] has been the de-facto standard benchmark. This dataset consists of 9144 images collected from the Internet using image search engines. It has 101 object classes and a background class, each of which has between 31 and 800 images. Some examples are shown in Figure 2.5. Caltech-101 has a wide variety of classes, though position, scale, and direction of objects are roughly aligned. In 2007, the more difficult Caltech-256 dataset [69] was released, with 256 classes (Figure 2.6). Compared to Caltech-101, it is characterized by an increased number of classes and high intra-class variations. Current state-of-the-art methods achieve an 80% classification rate on Caltech-101, and 50% on Caltech-256 [65; 210] respectively.

The PASCAL Visual Object Classes (PASCAL VOC) [51; 52] are also widely used benchmarks. They were introduced in a workshop for generic image recognition called the VOC Challenge. While there are only 20 classes, many tasks are evaluated including categorization, detection, and segmentation. Recently, they have often been used as benchmarks for detection methods.

2.3.2 Large Datasets

Crowd sourcing

The above mentioned small datasets are mainly used for benchmarking algorithms, where the utility of the resulting recognition system is not considered. To realize useful generic image recognition, it is vitally important to build a large-scale training corpus covering a wide variety of object appearances in the real world. This requires enormous human effort. One promising approach is crowd sourcing, where an "anonymous crowd" participates in building datasets.

The LabelMe project [159] is an early example of this approach. Anonymous users label object regions in images using the annotation tool provided over the Internet. Moreover, users can upload new images freely. In exchange for a little labeling work, researchers get an entire dataset. In 2009, the LabelMe framework was extended to support movies [215]. However, a problem with this approach is that the system is totally dependent on volunteers. Since image labeling is a tiring work, it is unrealistic

2.3. Training Image Corpus

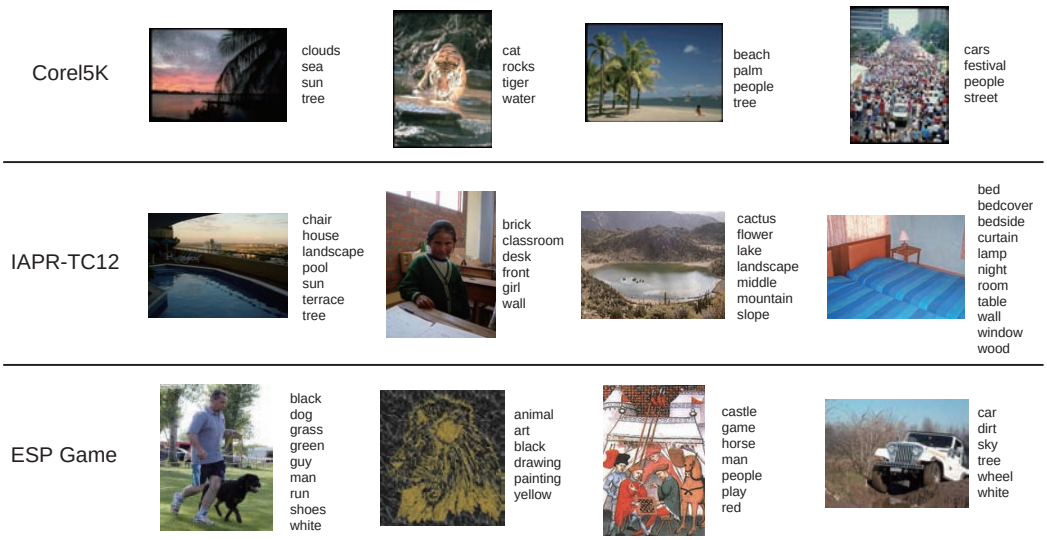


Figure 2.4: Three standard benchmarks for image auto-annotation. Top: Corel5K [50]. Middle: IAPR-TC12 [125]. Bottom: ESP Game [125].

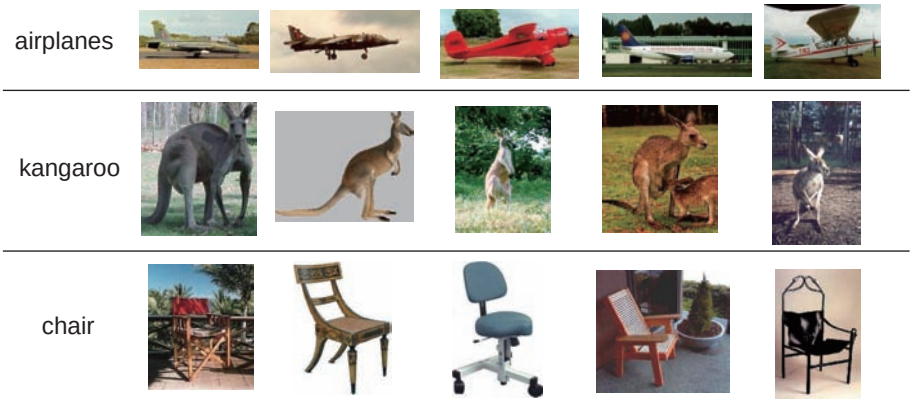


Figure 2.5: Example images from Caltech-101 dataset [55; 56].

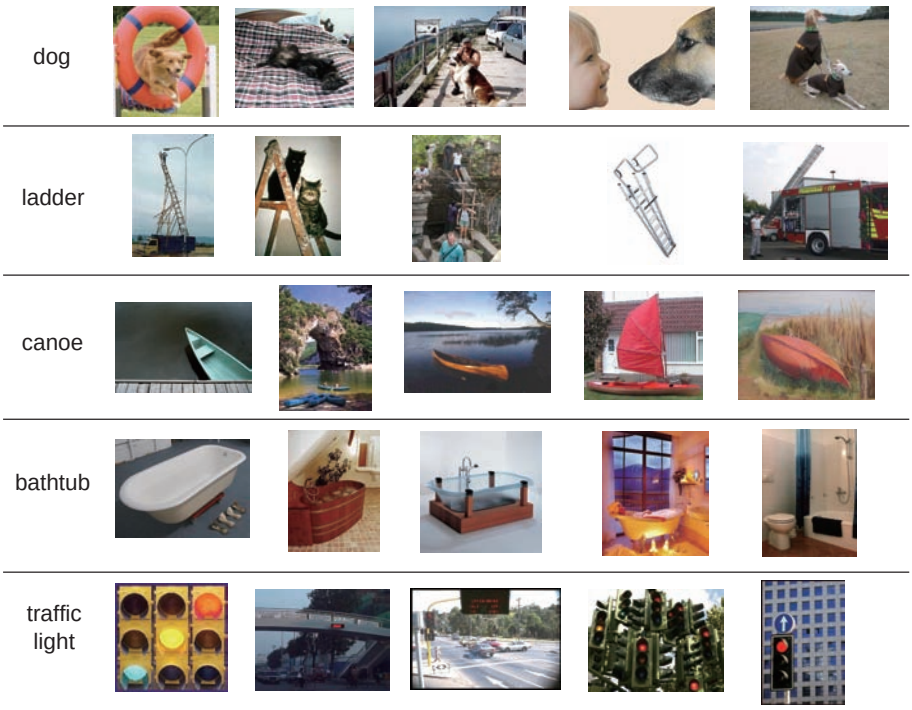


Figure 2.6: Example images from Caltech-256 dataset [69].

to expect significant effort from public contributors other than the researchers.

In contrast, some works attempt to motivate users by turning the labeling process itself into a game. The ESP game [189] provides the following image annotation game. First, two anonymous players are randomly connected over the Internet. The system displays the same image to both players. The players freely annotate the image with certain words, and are scored based on the number of words matching those used in the annotation by the other player. Corresponding annotations are assumed to be reliable. By repeating the game with different players, the system finally extracts reliable words as the ground truth. Similarly, Peekaboom [190] provides a game for object region labeling. Thus, by minimizing the effort of image labeling, these works have succeeded in building large-scale datasets consisting of millions of images. However, since most players' objective is merely to play the game, the quality of the annotations is not always high. Moreover, user annotations tend to be limited to common words and lack diversity, which is a serious problem if we want a versatile system.

The Lotus Hill dataset [212], provided by ImageParsing.com, is a large-scale multi-purpose image dataset constructed by paid human experts. By hiring human experts, this dataset guarantees high quality and detailed annotations compared to other datasets. However, since human resources are more or less limited, it is difficult to handle further large-scale data. Moreover, because the greater part of the dataset is not free, it is not always of benefit to academic studies.

Recently, Amazon Mechanical Turk (AMT) [173] has attracted much attention as a breakthrough. AMT is an online job posting service, through which anonymous workers can be employed for image labeling. Depending on the task and salary, certain motivated workers might participate in labeling. Moreover, since we can arbitrarily design the labeling task, we can construct various datasets for each research objective. AMT has already been used in many studies, with the biggest project being ImageNet [46]. The ImageNet dataset is still under construction based on the WordNet ontology [58]. At present (February 2011), it has 12 million images of 17,624 classes. In ECCV 2010, Deng *et al.* [45] reported classifying 10,000 categories using ImageNet data. Also, a workshop for large-scale image recognition was held, where participants worked on classifying 1,000 categories [13]. This is significant in that generic image recognition on this scale is now able to be evaluated quantitatively. As a further example of an AMT based dataset, Xiao *et al.* published the SUN dataset [207] consisting of 800 scene categories.

Web image mining

Thanks to crowd sourcing, we can now obtain comparatively large-scale supervised datasets. Still, dataset construction depends on human workers and the amount of processable data is limited. In fact, far more data are available on the Internet, with the amount growing by the day. For example, by 2010, more than four billion images were

stored in Flickr [197]. Google has indexed even more images. Web images are often accompanied by some semantic information such as a text document. It is expected that we can use such image-text data for training image recognition systems. This approach is called web image mining [60; 206; 228]. The system could learn a broad picture of the world as there are an infinite number of images taken under various conditions and environments on the Internet. Also, because semantic information attached to images was prepared by many different people, the system might be able to extract "common knowledge" for image understanding that can satisfy everyone.

In the natural language processing field, large-scale statistical learning using web data has been studied since the early 2000s [6]. It has been shown that performance improves proportional to the log number of training samples. Probably the first work in the computer vision field is AnnoSearch [196]. This system provides a collaborative annotation framework, where users need to input at least one exact keyword describing the query image. However, effective tasks for this approach are limited because user-provided keywords are not always available.

Regarding fully automatic image recognition, many methods are based on classical text-based image search engines. Specifically, a target word is used as the query for image retrieval. Then, retrieved images are used as positive examples of the word class. In this approach, we need to design classes that the system learns. In the beginning, several classes were selected experimentally [60; 228]. Recently, many works have made use of the WordNet ontology [58] to construct universal image dictionaries [46; 177]. Furthermore, some works exploit data-driven ontologies such as the Normalized Google Distance [38] and Flickr Distance [206] to realize fully automatic image knowledge acquisition.

A problem, however, is that the performance of text-based image search engines could be the bottleneck. Since it is difficult for current engines to exploit ontologies efficiently, each word is simply used as a query in many cases. In reality, it is very difficult to obtain a high-quality dataset, because many irrelevant images may be included due to homographs and noise. To address this problem, some early works proposed filtering methods such as rank-based filtering [60; 118], denoising via clustering [228], and visual similarity based filtering [118]. However, since the bottleneck is the accuracy of the search engines, it is difficult to improve the quality with ad-hoc post-processing. Therefore, many approaches have been proposed to obtain a high-quality dataset from the Internet, including interactive learning [14], online learning based on a semi-supervised framework [110], re-ranking using both visual and textual similarities [163], and spam tag filtering [53].

In spite of the above mentioned problems, web image mining has attracted more and more attention due to its ability of realizing extremely large datasets. Torralba *et al.* [177] downloaded 80 million images and performed k-nearest neighbor classification using simple image features. Despite the high noise ratio, annotation accuracy has consistently improved in proportion to the log number of training samples. Torralba *et*

al. reported that as the number of samples increases, the probability of finding a visually similar image also increases. Moreover, if the visual similarity exceeds a certain threshold, the probability of the images belonging to the same semantic category grows rapidly. The same phenomenon was observed in the ARISTA project [197], in which image annotation was conducted using two billion Internet images. This means that visual distance approaches semantic distance as the size of the dataset grows. Therefore, using web-scale training data is now considered one of the most promising approaches to bridge the semantic gap.

2.4 Designing the Image Recognition Method

In this chapter, we have discussed the current status of generic image recognition, where the semantic gap is a fundamental unsolved problem. The key to bridging the semantic gap is statistical machine learning using a large number of examples. Thus, learning methods and training datasets are equally indispensable factors and both of these must be considered in the design of a recognition system.

Although there are many tasks in generic image recognition, none has reached a practical level. This is mainly due to the lack of large-scale datasets to train a versatile recognition system. However, weakly labeled datasets, in which each image is globally labeled with certain words, are now growing exponentially owing to the advances in crowd sourcing and web mining technologies. Therefore, global labeling methods are now making considerable progress.

Moreover, global labeling methods can be useful to region labeling methods such as detection. Since detection is a high cost process in general, it is impractical to scan detectors of all target objects. Therefore, preprocessing to limit possible objects and scanning areas is important. It is expected that we can do this with global labeling methods that provide a rough description of an image in a short time.

Based on this background, we tackle the problem of global image labeling. We believe this is now the most important step in constructing practical generic image recognition systems. We require the following properties for our system.

1. Supports a framework of labeling a single image with multiple words. Here, the system should be able to learn from a weakly labeled dataset[1].
2. Relaxes the semantic gap using "contexts" represented by multiple labels.
3. Is scalable with respect to the number of training samples for both training and recognition.

To reach this goal, we focus on the following two challenges.

[1] (1) The absence of a label does not necessarily mean that the corresponding concept is not present in an image. (2) The correspondence between labels and image regions is not shown.

2.4.1 Tackling the Image Annotation Task

Our goal is to recognize unconstrained real-world images. In general, real-world images are miscellaneous and ambiguous, with various objects and scenes as their content. Take the upper-left image in Figure 2.4 as an example. What labels are likely to be used in annotating this image? There are many choices depending on one's subjectivity such as "sunset," and "water", although a rough understanding of the scene would be similar. Moreover, on a photo sharing site like Flickr, many adjectives and impression terms also appear such as "beautiful," and "impressive." The ground truth labels of this image are: "clouds," "sea," "sun," and "tree." All these words describe the image from a certain viewpoint, and can be said to be correct. Thus, ambiguity and redundancy problems are essential for generic image recognition. Recognition systems should flexibly learn from such data. This is the problem that is addressed in the field of image annotation.

Further, it should be noted that categorization is a specific case of annotation. In other words, if only one label is attached to each image, they become equivalent. Therefore, annotation methods can also be applied to the categorization problem without loss of generality. On the contrary, it is difficult to use categorization methods for the annotation problem, since they explicitly utilize the constraint that each sample has only one label. We discuss this further in Section 3.1.

For these reasons, we believe image annotation is the most important problem to be addressed.

2.4.2 Scalability for a Large-scale Training Corpus

As previously mentioned, the key to successful recognition systems is statistical learning using a large number of examples obtained by crowd sourcing or web mining. Moreover, the system should repeatedly learn from data when qualitatively new samples are added. However, because previous methods have emphasized recognition accuracy on small toy datasets, they generally lack scalability. It is practically impossible to apply these methods to web-scale problems. Therefore, we must develop an efficient method that is scalable enough to tackle this problem. Since recognition accuracy and computational costs are generally a trade-off, it is important to balance them at a high level.

To realize large-scale training, we must consider two factors: computational complexity and memory use. To relax computational complexity, one may wish to use simple linear classifiers that scale linearly in the number of training samples. However, using only linear classifiers is actually far from adequate in real problems. First, many practically used image features are embedded in non-linear manifolds. If we merely apply linear learning methods to these features, we cannot benefit from large-scale data. Furthermore, because existing linear methods often need to access data

iteratively, without sufficient memory, they require disk access during training. Since large-scale data rarely fit in the available memory, the time needed for disk access inevitably becomes a serious bottleneck [214]. Considering that the speed of storage devices has not increased at the same rate as CPUs, this problem is not negligible. Thus, large-scale training is a highly challenging task. We need to carefully design our algorithm to overcome the above mentioned problems.

Chapter 3

Related Work in Image Annotation

3.1 Previous Work

3.1.1 Region-based Generative Model

Currently, the aim of image annotation is to label a whole image, rather than region labeling. However, image annotation started with a region based approach [50; 133], the objective of which is to estimate the correspondence between a label and a region.

The word-image co-occurrence model [133] is a pioneering work. First, images are divided into grids of different resolutions. Then, some basic image features (*e.g.* color histograms and edge histograms) are extracted from each region. We refer to these as region features. Next, all region features from the training corpus are clustered into groups (clusters). It is expected that each cluster has visually similar region features. By taking the co-occurrence of region features and labels in each cluster, we estimate the posterior probability of each word. Although this is a simple method, it has the basic structure of the region based annotation approaches that flourished in the last decade.

Thereafter, the "blobworld" [32] approach, in which an image is represented by several region features (blobs), was applied to image annotation. While this is basically similar to the co-occurrence model, it differs in that it is based on image segmentation methods. The most well-known early work is the word-image translation model [50], which exploits a statistical machine translation method [27] for the image annotation problem. First, region segmentation is performed via normalized-cut [166]. Next, blobs are extracted from each region. These are vector quantized via clustering in the same manner as the co-occurrence model. The translation model assumes vector-quantized blobs as image-side "words"[1], and attempts to translate them into text words.

[1]Note that "blob" here means a vector-quantized region feature, although some previous works exploit raw region features [7; 9].

The translation model uses the EM algorithm to estimate the posterior probability of words for given blobs, whereas the co-occurrence model merely exploits co-occurrence frequencies. Similarly, in [89] another statistical translation method was presented based on maximum entropy [15] which reported greater annotation accuracy than that of the translation model.

The machine translation approach assumes that each blob has a one-to-one relation with a word. However, this assumption does not always hold in real problems because blobs are generated in a bottom-up manner. Therefore, it is important to model the entire relationship between multiple blobs and multiple words within an image. The Cross-Media Relevance Model (CMRM) [88] does this using a sample-based joint model. The CMRM is an application of a cross-lingual relevance model [104] for image annotation problems. Each image is represented by a histogram of its blobs. A query image is annotated using the weighted average of training labels of the similarity of blob histograms. Intuitively, this is similar to a k-nearest neighbor classification using blob histograms as the image features. In this sense, the CMRM is interpreted as an early example of a non-parametric approach, which is the current mainstream approach. Meanwhile, unlike the translation model, the CMRM cannot annotate regions because it does not model the blob-to-word relation explicitly. Nevertheless, this approach was shown to achieve high performance, probably because it is well-suited to estimating global image similarities.

The co-occurrence model, translation model, and CMRM are all blob-based methods, where region features are quantized as blobs. However, in practice, blob-based methods could cause performance to deteriorate owing to quantization errors. The Continuous Relevance Model (CRM) [105] was the turning point in this respect. Both the CRM and CMRM basically follow the same approach as illustrated in Figure 3.1, where an image and words are connected using the instances of training samples. The major difference is that the CRM uses raw region features directly for computing sample similarities. In other words, similarity between a query and a training sample is computed in terms of the product of the similarities of their region features. It is interesting that the CMRM significantly improves performance compared with previous methods, while the implementation is simpler because it does not require clustering or vector quantization to compute blobs.

In the CRM, the initial region segmentation depends on a Normalized-cut [166]. However, a fundamental problem is that annotation accuracy is strongly influenced by the performance of the region segmentation. This means that the segmentation process could be the bottleneck for the entire system. Therefore, in CRM-Rectangles [59] the segmentation process was replaced by simple grids, and achieved better performance than the original CRM. The Multiple Bernoulli Relevance Model (MBRM) [59], which is an updated version of the CRM, also exploits grids for region separation. In addition, [127] further exploits the Inference Network (InfNet)[181] scheme, which is a technique to formulate query operators (*e.g.* AND, OR) explicitly in a graphical model.

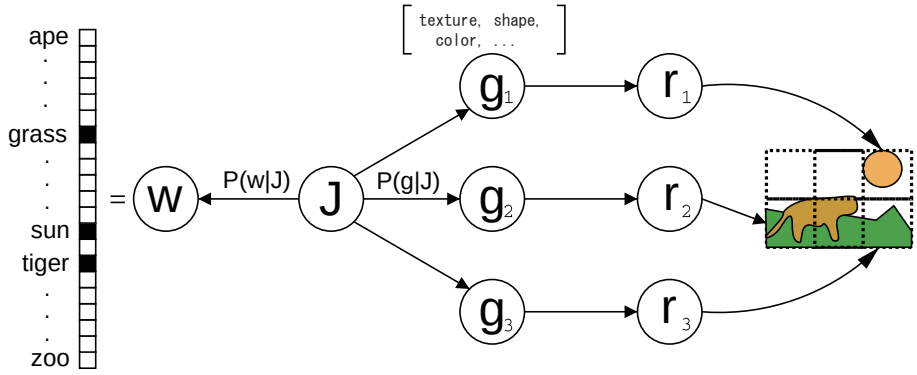


Figure 3.1: Graphical model of the CRM and MBRM. Credit: Feng *et al.* [59].

Despite the CRM and MBRM being rooted in region-based approaches, today they are often interpreted as the earliest non-parametric works. In fact, after the CRM and MBRM, the research trend changed from region-based approaches to non-parametric approaches. In this sense, they are milestone works in the history of image annotation.

Below, we present the algorithms for the CRM and MBRM. In these methods, images are first partitioned into n regions. An image is then represented by its region features $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2 ..., \boldsymbol{x}_n\}$. In the experiments, images are divided into 5×5 tiles (n=25). Also, we let $\boldsymbol{w} = \{w_1, ..., w_q\}$ denote sample labels, where each w_i is a word.

The joint probability $P(X, \boldsymbol{w})$ can be represented by averaging over the training samples as follows.

$$P(X, \boldsymbol{w}) = \sum_{J \in T} P(J) P(X, \boldsymbol{w}|J) = \sum_{J \in T} P(J) P(X|J) P(\boldsymbol{w}|J), \tag{3.1}$$

where T is the training dataset and J represents a sample. We assume conditional independence of X and $\boldsymbol{w}$ for a given J. For simplicity, the prior probability of samples is set to a constant. Specifically, letting N denote the number of training images,

$$P(J) = \frac{1}{N}. \tag{3.2}$$

The conditional probability of X for a given J is defined as follows.

$$P(X|J) = \prod_{i=1}^{n} P(\boldsymbol{x}_i|J). \tag{3.3}$$

Specifically, we assume region features are conditionally independent of J. The conditional probability of a region feature is defined as follows.

$$P(\boldsymbol{x}|J) = \frac{1}{n} \sum_{j=1}^{n} \frac{\exp\{-(\boldsymbol{x} - \boldsymbol{x}_j^J)^T \Sigma^{-1} (\boldsymbol{x} - \boldsymbol{x}_j^J)\}}{\sqrt{2^k \pi^k |\Sigma|}}, \tag{3.4}$$

where $\Sigma = \beta I$. β is the bandwidth of the kernels.

The CRM and MBRM share the same model as described above. The main difference is the implementation of the language model $P(\boldsymbol{w}|J)$ as given below.

$$P_{CRM}(\boldsymbol{w}|J) = \prod_{w \in \boldsymbol{W}} P(w|J), \tag{3.5}$$

$$P_{MBRM}(\boldsymbol{w}|J) = \prod_{w \in \boldsymbol{W}} P(w|J) \prod_{w \notin \boldsymbol{W}} (1 - P(w|J)). \tag{3.6}$$

$P(w|J)$ is common to both methods, that is,

$$P(w|J) = \mu \frac{\delta_{w,J}}{N_J} + (1 - \mu) \frac{N_w}{N_W}, \tag{3.7}$$

where N_J is the number of ground truth labels in J, N_w is the number of images that contain w in the training dataset, $\delta_{w,J}$ is one if label w is annotated in training sample J, otherwise zero, and μ is a parameter between zero and one. As μ approaches one, each sample label is more emphasized.

3.1.2 Local Patch Based Generative Model

In a region based approach, we consider a joint generative model for region features and words. Here, we consider a model for local features and words. Since a local feature can be interpreted as the region feature of a small region, a patch based approach is somewhat analogous to a region based approach. However, whereas an image is represented using only several region features in a region based approach, we need to describe each image using thousands of local features. Therefore, we need to consider efficient implementations to deal with the substantial computational costs.

Figure 3.2 illustrates the algorithm for Supervised Multiclass Labeling (SML) [29; 30; 31], a representative example of this approach. First, the system models the distribution of local features of each image with a Gaussian mixture model. Then, by averaging the distributions over all samples with the targeted word, it obtains a generative model of local features specific to the word. Because SML trains a large-scale parametric model, training would quickly become infeasible as the numbers of images and words increase. Still, it is scientifically important to attempt to train a local feature based generative model; such a work was a major interest in the community.

3.1.3 Binary Classification Approach

This is one of the most classical approaches, along with the region based one. Classifiers are constructed independently for each word class, while annotations are sorted with respect to the response of each classifier. This strategy is called the one-versus-all

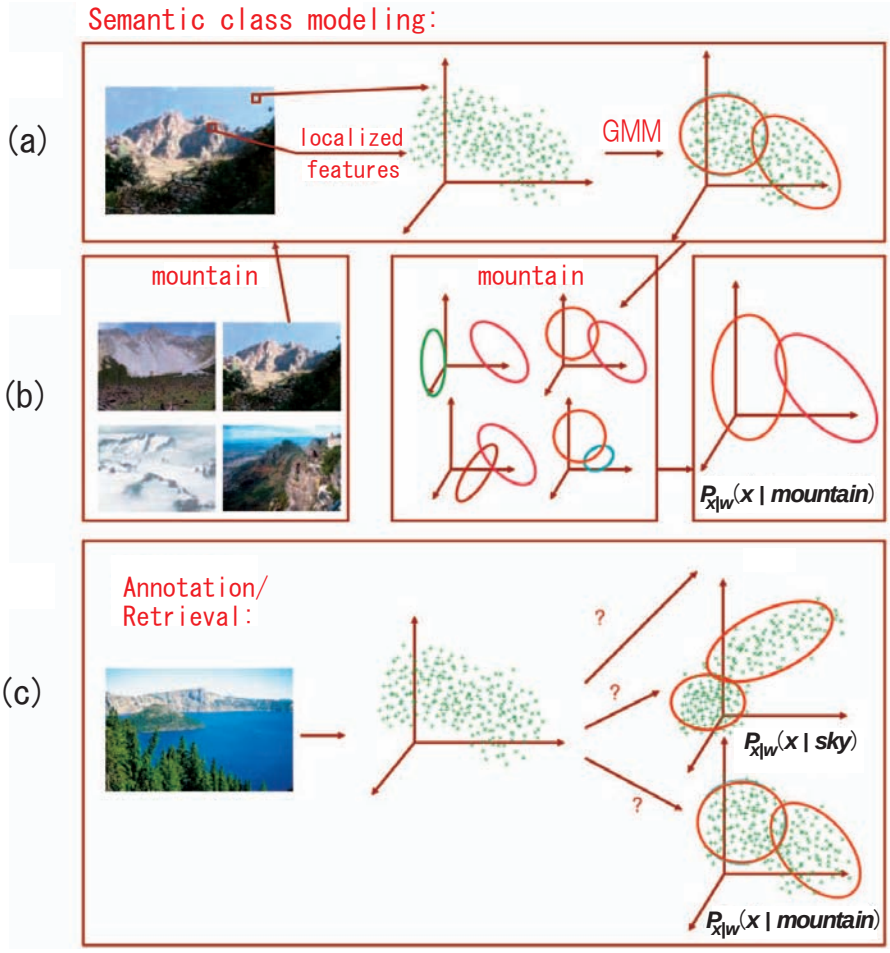


Figure 3.2: Illustration of SML. Credit: Carneiro *et al.* [29].

approach, which is frequently used for categorization tasks. It has also been applied to image annotation tasks. For example, the Support Vector Machine (SVM) [41] and Bayes point machine [34] are commonly referred to in the literature. Although the SML also builds word-specific classifiers, a major difference is that these methods follow a discriminative approach, whereas the SML is a generative method.

A problem with these methods is that they do not consider between-word dependencies. In the annotation framework, several words are used as labels for a single image. These words are thought to be mutually correlated. For example, "sky," "clouds," and "sun" often appear together in an image. A binary classification approach neglects such dependencies and leads to a redundant model. Since the importance of context described by multiple words has been pointed out in recent works, the simple binary classification approach is now considered unsuitable for annotation problems.

In [119] a method to exploit multiple words via matrix factorization was presented. This method first derives a semantic subspace shared by all classes, and then trains SVM classifiers in the subspace for each class. With this approach, annotation performance is substantially improved compared to conventional SVM based methods.

In addition, in the field of attribute-based object recognition, which has attracted much attention recently, some methods exploit the co-occurrence information of several properties of an object (*e.g.* object name, color, texture) in training binary classifiers [193]. However, they assume that each property corresponds to the same region in an image. This assumption does not always hold in image annotation problems and therefore, it is difficult to apply this approach to image annotation.

3.1.4 Graph-based Approach

Graph-based image captioning (GCap) [146; 147] constructs an undirected graph as follows. First, region features and labels in an instance (image) are connected to the instance itself. Then each region feature is connected to neighboring region features in the whole training dataset. A query image is connected to this graph using its region features. Annotation is then carried out by means of a random walk, taking the query itself as the starting point. More specifically, annotations are ranked by stationary probability.

The adaptive graph-based annotation method (AGAnn) [117] uses global image features for graph construction, rather than region features. A graph construction method, called the nearest spanning chain (NSC), which can adapt to local structures of data distributions, was proposed for the AGAnn. The benefit of this method is that it is less sensitive to parameters compared to general k-NN based methods. Each instance is represented as a node on a connected graph. By propagating sample labels on this graph, annotation results can be estimated. The two-phrase Graph Learning Method (TGLM) [116] improves annotation accuracy by employing a graph based on word similarities in addition to one based on image similarities. Word similarities

can be estimated not only from manually prepared training datasets, but also from the Internet.

The core of graph-based methods is to propagate labels of training samples similar to a query. In this sense, they are relatively similar to non-parametric methods.

3.1.5 Regression Approach

If a regression model of words can be constructed, we can perform annotation by a simple projection. In [224], annotation is conducted using a linear projection obtained by canonical correlation analysis. Further, in [225], linear regression models are trained to predict the area of each object in an image. However, it is difficult to model correspondence between image features and labels using a simple linear method.

In [74], Kernel Canonical Correlation Analysis (KCCA) is used to build a non-linear regression model. Further, in [209], multiple kernel learning [103] is applied to KCCA and Kernel Multiple Linear Regression (KMLR) to construct a powerful regression model. In addition, a non-parametric annotation rule is proposed that utilizes the projected point of a query image. However, in general, kernel methods seriously lack scalability as discussed earlier in this thesis.

3.1.6 Topic Model Approach

Topic models have been developed mainly by the natural language processing community. They have successfully been applied to clustering and data mining of text documents. Latent Semantic Analysis (LSA) [44], probabilistic Latent Semantic Analysis (pLSA) [79], and Latent Dirichlet Allocation (LDA) [20] are representative methods. These models have also been applied to the problem of image recognition. While the original topic models contended with the problem of one modal compression (text), we need to consider two modals (image and words) for image recognition.

In this framework, we consider the graphical model illustrated in Figure 3.3. We assume an unobserved latent node $\boldsymbol{l}$ above the image and words. A latent variable is first selected, and then it generates the image and words. Here, we impose the naive Bayes assumption that $\boldsymbol{x}$ and $\boldsymbol{w}$ are conditionally independent for a given $\boldsymbol{l}$. The joint probability of image features and words are represented as follows.

$$P(\boldsymbol{x}, \boldsymbol{w}) = \int P(\boldsymbol{x}, \boldsymbol{w}|\boldsymbol{l})\, P(\boldsymbol{l})\, d\boldsymbol{l} \tag{3.8}$$

$$= \int P(\boldsymbol{x}|\boldsymbol{l})\, P(\boldsymbol{w}|\boldsymbol{l})\, P(\boldsymbol{l})\, d\boldsymbol{l}. \tag{3.9}$$

Here, we exploit the assumption of conditional independence $P(\boldsymbol{x}|\boldsymbol{w}, \boldsymbol{l}) = P(\boldsymbol{x}|\boldsymbol{l})$.

A latent variable can be interpreted as a "topic" that captures an essential relationship between an image and its words. As shown, the image and words are modeled

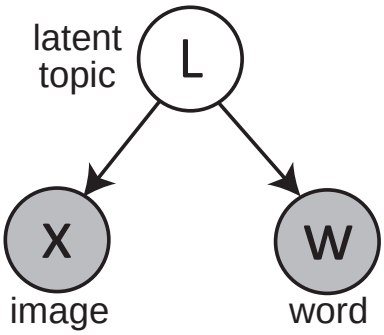


Figure 3.3: A topic model for image annotation.

by a mixture of probabilistic distributions generated from each topic. By averaging in terms of topics, it is expected that we can represent a complex structure of images and words with relatively simple density functions.

For example, in the real world, there are many "fish" images belonging to different topics, such as the "sea" or "food" topic. Although they are all "fish", their appearances would differ widely depending on their topics. This fact makes it difficult to estimate the "fish" model directly. In contrast, it would be easier to estimate a topic-specific "fish" model since within-topic images are expected to be similar. Thus, we can obtain the final "fish" model as a mixture of topic-specific models.

The essential problem is, how to define a latent variable (topic) and estimate it. Ref. [7; 9] hierarchically clusters blobs and words using the EM algorithm. Each cluster serves as a latent variable and generates image features and words with a Gaussian distribution and a multinomial distribution, respectively. However, the model lacks flexibility because all region features and words within a sample are exclusively generated by one topic.

To address this problem, [8; 19] proposed several models based on LDA. Gaussian-Multinomial LDA (GM-LDA), a baseline method, samples the latent variables of each region feature and word using a multinomial distribution specific to each sample. Parameters of multinomial distributions are sampled with a Dirichlet distribution, which is tuned with a hyper parameter. With this model, it is possible to represent multiple region features and words within a sample as a mixture of multiple topics. This property makes the model highly expressive. Also, GM-LDA can perform region labeling. However, because latent variables of region features and words are randomly generated, their dependency is not explicitly considered. While GM-LDA is suited to a word-image generative model, it is not always effective for image annotation problems where the posterior probabilities of words are important. As a solution, Multi-Modal Latent Dirichlet Allocation (MoM-LDA) explicitly models a cross-modal dependency of latent variables. It achieved better performance than GM-LDA.

More recently, LDA has eagerly been studied for application to image recognition. For example, in [57], LDA was used to classify 13 scene classes, while in [208] a hierarchical extension of MoM-LDA was proposed and applied to image annotation.

Besides LDA, pLSA is also a representative method of topic models. Although the original pLSA was developed earlier than LDA, their applications to image recognition were published around the same time. Ref. [129] simply concatenates an image feature vector and a text feature vector, and applies original LSA and pLSA to it. However, since query images do not have associated text, we cannot immediately perform annotation. Although the method in [129] heuristically placed zeros in the text, its theoretical basis is unclear. Moreover, it has been pointed out that the learned result of this model is almost equal to that of a text-only model, because text features are far more descriptive than image features in general. Ref. [130] improved the performance by introducing an asymmetric model, where the topic of each document is estimated using text features only. Details of these methods are summarized in [131].

pLSA has been studied in many works. In particular, it has attracted much attention since the bag-of-visual-words [40] technique was established. This method enabled images to be interpreted in the same manner as for text documents, making a pLSA based approach more reasonable. Ref. [23; 25] employed pLSA for dimensionality reduction and achieved good performance in scene classification. Ref. [114] investigated large-scale pLSA methods and applied them to image retrieval, while [113] proposed a multimodal pLSA, which is a hierarchical combination of modal-specific pLSAs.

3.1.7 Non-parametric Approach

This approach uses the training labels of neighboring samples of a query image directly, and is represented by a classical k-NN classifier. As mentioned in Section 3.1.1, CRM [105] and MBRM [59] are early examples of non-parametric methods. Despite their simplicity, they showed surprisingly high performance in annotation. Since then, many other methods have been proposed in this direction. Non-parametric Density Estimation (NPDE) [213] uses global image features for kernel density estimation. Correlated Label Propagation (CLP) [92] and Context-Based Keyword Propagation (CBKP) [122] are label propagation methods that consider co-occurrences. Also, the Dual Cross-Media Relevance Model (DCMRM) [118] exploits not only training labels, but also external ontologies to improve performance. Multi-label Sparse Coding (MSC) [191] uses a sparse coding technique to compute similarities between a query and the training samples.

Recent works have shown that we can substantially improve annotation accuracy by incorporating multiple image features. For example, Makadia *et al.* proposed the Joint Equal Contribution (JEC) method [125], which exploits multiple visual features (*e.g.* color histograms and Haar wavelets) to improve performance. For each feature, a base distance is defined using an appropriate metric in the feature space (*e.g.* χ^2 dis-

tance for color histograms and L1 distance for Haar wavelets). Then, all the base distances are concatenated with equal weights to retrieve the nearest neighbors. Despite its simplicity, JEC achieved the best performance as of 2008. Furthermore, Guillaumin *et al.* proposed TagProp [71], which realizes state-of-the-art performance. This method makes use of 15 powerful global and local features, including bag-of-visual-words [40] and GIST features [144], amongst others. TagProp differs from JEC in that the weights for the base distances are optimized in the metric learning framework by directly maximizing the log-likelihood of the tag prediction.

The success of these methods is thought-provoking, and is somewhat analogous to that of the multiple kernel learning [103] approach in categorization tasks. The key issue here for improving performance is to employ rich visual features with appropriate distance metrics defined in raw feature spaces.

3.1.8 Summary

First, we compare the annotation accuracy of previous works. In the field of image annotation, Corel5K [50] has been used as the de-facto standard benchmark dataset for a long time. For details of Corel5K, refer to Chapter 5. Performance is evaluated with mainly three scores: mean recall (MR), mean precision (MP), and F-measure. Higher scores imply greater annotation accuracy. For details, refer to Appendix A.

Table 3.1 summarizes the scores of previous studies. Scores are shown in ascending order of F-measure, while the names of non-parametric methods are given in bold face. As illustrated, non-parametric methods have historically achieved good performance. Although we cannot compare the scores directly because each method uses different image features, it is interesting that the state-of-the-art methods developed after JEC all follow a non-parametric approach.

The superiority of non-parametric methods is due to the nature of the image annotation task, where the system outputs multiple words for a single image. Unlike the exclusive categorization task, words are mutually correlated in the annotation task. Therefore, we need to consider co-occurrence information of labels in the dataset. Non-parametric methods can do this implicitly by directly using the sample labels. Moreover, since image annotation is a highly generic task that needs to model a complex probabilistic distribution, parametric methods tend to become more complicated since a number of parameters must be estimated. In contrast, non-parametric methods are relatively stable since they estimate a distribution in an example-based manner. Moreover, they can accept qualitatively new samples instantly by merely adding them to the dictionary. For these reasons, we believe that non-parametric image annotation is the most practical methodology. In the remainder of this thesis, we develop our image annotation method based on this approach.

Table 3.1: Performance of previous works using Corel5K.

	Year	MR	MP	F-m	N+	MAP	MAP (R+)
Co-occurrence [133]	1999	0.02	0.03	0.02	19	-	-
Translation [50]	2002	0.04	0.06	0.05	49	-	-
CMRM [88]	2003	0.09	0.10	0.09	66	0.17	-
Maximum Entropy [89]	2004	0.12	0.09	0.11	-	-	-
CRM [105]	2003	0.19	0.16	0.17	107	0.24	-
NPDE [213]	2005	0.18	0.21	0.19	114	-	-
InfNet [127]	2004	0.24	0.17	0.20	112	0.26	-
CRM-Rectangles [59]	2004	0.23	0.22	0.23	119	0.26	0.30
Independent SVMs [119]	2008	0.22	0.25	0.23	-	-	-
MBRM [59]	2004	0.25	0.24	0.25	122	0.30	0.35
AGAnn [117]	2006	0.27	0.24	0.25	126	-	-
SML [29]	2007	0.29	0.23	0.26	137	0.31	0.49
DCMRM [118]	2007	0.28	0.23	0.26	135	-	-
TGLM [116]	2009	0.29	0.25	0.27	131	-	-
MSC [191]	2009	0.32	0.25	0.28	136	0.42	**0.79**
Matrix Factorization [119]	2008	0.29	0.29	0.29	-	-	-
JEC [125]	2008	0.32	0.27	0.29	139	0.33	0.52
CBKP [122]	2009	0.33	0.29	0.31	142	-	-
Group Sparsity [220]	2010	0.33	0.30	0.31	146	-	-
TagProp [71]	2009	**0.42**	**0.33**	**0.37**	**160**	**0.42**	-

3.2 Bridging the Semantic Gap for Non-parametric Image Annotation

As shown in a previous section, an example-based non-parametric approach is effective for the image annotation problem. However, two major problems need to be addressed.

The first one is the semantic gap which we discussed in Section 2.2. In general, similarity between samples is evaluated by the distance between image features. However, low-level image features are not necessarily related to the meanings of images. To address this problem, we need to use as many training samples as possible. Furthermore, the system must learn a discriminative distance metric using label information provided by humans in a machine learning framework.

The second problem is that the computational costs, of both complexity and memory use, tend to be high with the growth of the training datasets. In general, image representations need to be high-dimensional to build a versatile system[1]. A non-parametric method must store all training instances in memory to compute their respective distances from the input queries. This cost becomes prohibitive when high-dimensional features and a large number of training samples are used.

For these reasons, we require a method that performs both dimensionality reduction and discriminative metric learning. With this in mind, we summarize the related methods in this section.

3.2.1 Distance Metric Learning

Let $\boldsymbol{x} \in R^p$ denote an input feature vector in the original feature space. For simplicity, we assume that the image features are originally embedded in a Euclidean space[2]. Without any prior knowledge, the distance between two samples i, j is computed by the Euclidean distance.

$$dist_E(i, j) = \sqrt{(\boldsymbol{x}_i - \boldsymbol{x}_j)^T(\boldsymbol{x}_i - \boldsymbol{x}_j)}. \tag{3.10}$$

Mahalanobis distance metric learning (MDML) is a framework to learn the Mahalanobis distance defined by a positive semi-definite symmetric matrix M as follows.

$$dist_M(i, j) = \sqrt{(\boldsymbol{x}_i - \boldsymbol{x}_j)^T M(\boldsymbol{x}_i - \boldsymbol{x}_j)}. \tag{3.11}$$

We can rewrite $M = WW^T$ using a Cholesky decomposition. Therefore, Equation 3.11 can be interpreted as the Euclidean distance in a new space (subspace) defined by the

[1] For example, TagProp uses 15 features, resulting in more than 37,000 dimensions.
[2] Without loss of generality, we can derive non-linear extensions using kernel methods.

projection W. In fact, classical dimensionality reduction methods (*e.g.* PCA) empirically do exactly this. MDML presents a generic framework from the viewpoint of defining similarity between samples.

The definition of M is dependent on the objective of the task. For each task, M is trained by optimizing a task-specific evaluation function. For example, Locality Preserving Projections (LPP) [76], a typical manifold learning method, attempts to preserve local neighborhood structures in the original space. Our goal is to learn a discriminative distance metric that relaxes the semantic gap with the help of label information. Next we introduce previous methods related to this issue.

Basically, discriminative MDML methods are designed for k-nearest neighbor classification. Neighborhood Components Analysis (NCA) [67], a pioneering work, learns the metric so that the leave-one-out k-NN classification score in the training dataset can be maximized. Maximally Collapsing Metric Learning (MCML) [66] designs a convex evaluation function that forces within-class samples to be mapped to the same point, while out-of-class samples are placed at an infinite distance. Similarly, Large Margin Nearest Neighbor (LMNN) [200] optimizes the metric so that k neighboring samples of each training sample belong to the same class, while out-of-class samples are placed as far away as possible. Fast-LMNN [201], a speeded up version of LMNN, has also been proposed, while Information-Theoretic Metric Learning (ITML) [43] exploits prior knowledge in a information-theoretic manner by introducing a Gaussian distribution specific to M and optimizes its LogDet divergence.

Additionally, there are many MDML methods designed for various tasks other than k-nearest neighbor classification. For example, ranking-based distance metric learning [192] is designed for similar image retrieval. This method learns the metric using the accuracy of ranked retrieval as the evaluation function. Also, [36; 169] incrementally learns the metric exploiting the user log data in retrieval. Thus, MDML has been successfully applied to many tasks including similar image search [36; 80; 87; 192] and facial image recognition [72]. Moreover, more recently, local distance metric learning [63; 64; 157; 192], a technique to train different Mahalanobis distance metrics in each local area in the feature space, has been thoroughly studied, although this is beyond the scope of this discussion.

The advantage of the above mentioned MDML methods is that they can train the distance metric explicitly utilizing local structures of data distributions. However, they also have some disadvantages. First, they lack scalability. Many methods are based on pair-wise or triplet-wise computation for training. The training costs are inevitably $O(N^2) \sim O(N^3)$ (where N is the number of training samples). Second, they do not consider dimensionality reduction explicitly. Although dimensionality reduction can be performed by forcing a low-rank constraint on W, retraining is necessary to change the dimensionality. Furthermore, since most of the methods iteratively access training data, they will inevitably have to deal with the memory problem in true large-scale settings as discussed in Section 2.4.2. Considering these problems, we focus on MDML

based on simple linear dimensionality reduction methods as described below.

3.2.2 Bimodal Dimensionality Reduction Methods

The dimensionality reduction methods discussed here can be interpreted as simple MDML using a global evaluation function. They are suitable for our objective for the following reasons.

- Training complexity is $O(N)$, where N is the number of training samples.
- Memory use for training is constant in N.
- Iterative access to training data is not necessary.
- A global optimal solution is analytically obtained.
- Dimensionality can be set arbitrarily once the training phase is done.

Suppose we have a p-dimensional image feature $\boldsymbol{x}$, and a q-dimensional label feature $\boldsymbol{y}$. Suppose also, that we have N labeled training samples $\{\boldsymbol{x}_i, \boldsymbol{y}_i\}_{i=1}^N$. We let $C = \begin{pmatrix} C_{xx} & C_{xy} \\ C_{yx} & C_{yy} \end{pmatrix}$ denote the sample covariance matrix obtained from the training dataset, where

$$C_{xx} = \frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^T, \tag{3.12}$$

$$C_{yy} = \frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{y}_i - \bar{\boldsymbol{y}})(\boldsymbol{y}_i - \bar{\boldsymbol{y}})^T, \tag{3.13}$$

$$C_{xy} = \frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{y}_i - \bar{\boldsymbol{y}})^T, \tag{3.14}$$

$$C_{yx} = C_{xy}^T, \tag{3.15}$$

In the above equations, $\bar{\boldsymbol{x}}$ and $\bar{\boldsymbol{y}}$ denote the sample means. The objective is to obtain a new d-dimensional small vector $\boldsymbol{r}$ ($d \ll p$), whose distance metric could be the L2 distance. We call this the compressed feature.

Partial Least Squares (PLS)

Partial least squares (PLS) [204] is a common tool for multi-modal dimensionality compression. It finds linear transformations $\boldsymbol{s}_{PLS} = V_x^T(\boldsymbol{x} - \bar{\boldsymbol{x}})$ and $\boldsymbol{t}_{PLS} = V_y^T(\boldsymbol{y} - \bar{\boldsymbol{y}})$ that

maximize the covariance between the new values $\boldsymbol{s}_{PLS}$ and $\boldsymbol{t}_{PLS}$. The projection matrices V_x and V_y are obtained by the following eigenvalue problems:

$$C_{xy}C_{yx}V_x = V_x\Theta \quad (V_x^T V_x = I_d), \tag{3.16}$$
$$C_{yx}C_{xy}V_y = V_y\Theta \quad (V_y^T V_y = I_d). \tag{3.17}$$

where Θ is a diagonal matrix with eigenvalues as elements. A latent vector is obtained by $\boldsymbol{r}_{PLS} = V_x^T(\boldsymbol{x} - \bar{\boldsymbol{x}})$. Therefore, the new distance metric obtained via PLS is given by:

$$dist_{PLS}(i,j) = \sqrt{(\boldsymbol{x}_i - \boldsymbol{x}_j)^T V_x V_x^T (\boldsymbol{x}_i - \boldsymbol{x}_j)}. \tag{3.18}$$

The result of PLS is strongly influenced by the variances of the original features. Therefore, we also test PLS after normalizing the variances of the original feature elements. Specifically, for image features, we perform the following normalization.

$$\boldsymbol{x}' = \Sigma_X^{-1}(\boldsymbol{x} - \bar{\boldsymbol{x}}), \tag{3.19}$$

where Σ_X is a diagonal matrix with the standard deviation of each feature as its elements. The same normalization is applied to label features. We refer to this as normalized PLS (nPLS).

Although PLS is a classical method, it has been employed successfully in a state-of-the-art human detection method [164]. The authors compressed 170,000-dimensional features into 20-dimensional latent features without much deterioration in performance, making large-scale training tractable. Whereas the semantic aspect (y-view) in [164] is binary (human or non-human), we have multiple labels for a single image. These labels are expected to provide rich semantic information.

Canonical Correlation Analysis (CCA)

CCA was first proposed by Hotelling [82] in 1936, and has hitherto been one of the most basic and important multivariate analysis methods. CCA is closely related to PLS. Whereas PLS finds the projections that maximize the covariance between the two new values, CCA finds those that maximize the correlation. That is, it finds linear transformations $\boldsymbol{s}_{CCA} = V_x^T(\boldsymbol{x} - \bar{\boldsymbol{x}})$ and $\boldsymbol{t}_{CCA} = V_y^T(\boldsymbol{y} - \bar{\boldsymbol{y}})$ that maximize the correlation between the new values $\boldsymbol{s}_{CCA}$ and $\boldsymbol{t}_{CCA}$. Further details can be found in [22]. We obtain projection matrices U_x and U_y by solving the following eigenvalue problems:

$$C_{xy}C_{yy}^{-1}C_{yx}U_x = C_{xx}U_x\Lambda^2 \quad (U_x^T C_{xx} U_x = I_d), \tag{3.20}$$
$$C_{yx}C_{xx}^{-1}C_{xy}U_y = C_{yy}U_y\Lambda^2 \quad (U_y^T C_{yy} U_y = I_d). \tag{3.21}$$

where Λ is the diagonal matrix of the first d ($\min\{p,q\} \geq d \geq 1$) canonical correlations. A compressed feature is obtained by $\boldsymbol{r}_{CCA} = U_x^T(\boldsymbol{x} - \bar{\boldsymbol{x}})$. Therefore, the new distance metric obtained via CCA is given by:

$$dist_{CCA}(i,j) = \sqrt{(\boldsymbol{x}_i - \boldsymbol{x}_j)^T U_x U_x^T (\boldsymbol{x}_i - \boldsymbol{x}_j)}. \tag{3.22}$$

CCA has been successfully employed in some previous studies on image annotation [74; 209; 224]. The main focus in these studies, however, was to construct a strong regression model using CCA or KCCA, rather than dimensionality compression. Our objective is more similar to that of the correlational spectral clustering [18], in which CCA and KCCA are used for unsupervised clustering of weakly coupled image-text documents. The authors showed that the distance between instances can be estimated more accurately in the latent space, despite the dimensionality thereof being substantially reduced.

Multiple Linear Regression (MLR)

MLR is an intermediate method between PLS and CCA. It has an asymmetric structure in which one of two modals is whitened. Usually, this method is used for regression, as its name implies. In the case of image annotation, it is natural to take labels as objective variables. The problem is formulized as the following eigenvalue problems.

$$C_{xy}C_{yx}W_x = C_{xx}W_x\Omega \quad (W_x^T C_{xx} W_x = I_d), \tag{3.23}$$

$$C_{yx}C_{xx}^{-1}C_{xy}W_y = W_y\Omega \quad (W_y^T W_y = I_d). \tag{3.24}$$

where Ω is the diagonal matrix of the first d eigenvalues. A compressed feature is obtained as $\boldsymbol{r}_{MLR} = W_x^T(\boldsymbol{x} - \bar{\boldsymbol{x}})$. Thus, the distance metric obtained via MLR is given by:

$$dist_{MLR}(i, j) = \sqrt{(\boldsymbol{x}_i - \boldsymbol{x}_j)^T W_x W_x^T (\boldsymbol{x}_i - \boldsymbol{x}_j)}. \tag{3.25}$$

As in the case of PLS, the result of MLR is influenced by the variance of objective variables. Therefore, we also test performing MLR after normalizing the variance of label features. We call this normalized MLR (nMLR).

Relation between PLS, CCA, and MLR

PLS, CCA, and MLR are closely related methods [22]. Actually, CCA and MLR can be interpreted as performing PLS after a certain normalization of variables. We summarize this relationship in Table 3.2. Also, Table 3.3 gives the training complexity of each method. For fixed features, these methods scale to the number of training samples with linear complexity, a property beneficial to large scale problems.

Efficient implementation

Because only covariance matrices are necessary for solving the eigenvalue problems of the above mentioned methods, we do not have to preserve raw training data in memory.

Table 3.2: Relationship between dimensionality reduction methods. All methods can be interpreted as special cases of PLS.

Image features	Label features	Method
-	-	PLS
	$\Downarrow$	
variance normalization	variance normalization	nPLS
whitening	-	MLR
whitening	variance normalization	nMLR
whitening	whitening	CCA

Table 3.3: Computational complexity of PCA, PLS, and CCA based methods: (1) calculating covariances, (2) solving eigenvalue problems, and (3) projecting training samples using the learned metric.

	(1)	(2)	(3)
PCA	$O(Np^2)$	$O(p^3)$	$O(Npd)$
PLS	$O(Npq)$	$O(min\{p^2(p+q), (p+q)q^2\})$	$O(Npd)$
MLR	$O(N(p^2+pq))$	$O(p^3+p^2q)$	$O(Npd)$
CCA	$O(N(p^2+pq+q^2))$	$O(p^3+q^3+p^2q+pq^2)$	$O(Npd)$

For example, regarding the following covariance matrix:

$$C_{xx} = \frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{x}_i-\bar{\boldsymbol{x}})(\boldsymbol{x}_i-\bar{\boldsymbol{x}})^T, \tag{3.26}$$

$$= \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{x}_i\boldsymbol{x}_i^T-\bar{\boldsymbol{x}}\bar{\boldsymbol{x}}^T, \tag{3.27}$$

we can easily compute this by incrementally adding $\boldsymbol{x}_i$ one by one. Overall, we need to scan the training data only twice; once for computing covariances and once for projecting the data.

Chapter 4

Development of a Scalable Image Annotation Method

In this chapter, we focus on the dimensionality reduction methods discussed in Section 3.2.2. We develop a non-parametric image annotation method, which is computationally efficient in terms of both training and recognition [138; 140; 226]. The proposed method has the following advantages.

- Training complexity is linear in the number of training samples.
- It is not necessary to access data iteratively during training.
- Memory use for training is small and constant.
- During recognition, the cost of computing the sample distance is relatively small.

The core of our method is semantic dimensionality reduction together with similarity measures obtained via probabilistic canonical correlation analysis.

4.1 Non-parametric Image Annotation

Suppose N training pairs $T_i = \{I_i, L_i\}$ $(1 \leq i \leq N)$ are given. I is an image and L is its corresponding label. Given a query (new image) I_Q, we predict its labels using a sample based classifier. In this work, we consider two approaches, namely, k-nearest neighbor classification and MAP classification.

4.1.1 k-Nearest Neighbor Classification

The k-nearest neighbor algorithm is the most basic example of a non-parametric classification. Suppose a distance metric $DIST(I_Q, T_i)$ that defines the distance between

a query I_Q and a training sample T_i is given. According to this distance metric, the system outputs the most frequent labels in the k retrieved neighbors.

4.1.2 MAP Classification

As a more generic implementation, we also introduce a sample based MAP classifier, assuming that each sample constitutes a weak classifier. The posterior probability of a word w can be expressed as follows.

$$P(w|I_Q) = \sum_{i=1}^{N} P(w|T_i)P(T_i|I_Q). \tag{4.1}$$

Many previous works can be explained with this model.

In fact, k-nearest neighbor classification can be interpreted as a special case of Equation 4.1. That is, they become equivalent if we assume

$$P(T_i|I_Q) = \begin{cases} 1/k & \text{If } T_i \text{ is in the top } k \text{ nearest neighbors of } I_Q, \\ 0 & \text{otherwise.} \end{cases} \tag{4.2}$$

$$P(w|T_i) = \begin{cases} 1 & \text{If } w \text{ is given to sample } T_i, \\ 0 & \text{otherwise.} \end{cases} \tag{4.3}$$

Below, we describe the implementation of our method. $P(T_i|I_Q)$, which gives a weight to each training sample (a weak classifier), is described in detail in the following sections. For $P(w|T_i)$, we define a simple model in a top-down manner, like CRM [105]. We combine the labels of each sample and the inverse document frequency (IDF) of a word.

$$P(w|T_i) = \mu\delta_{w,T_i} + (1-\mu)\frac{\log(N/N_w)}{\log N}, \tag{4.4}$$

where N_w is the number of images that contain w in the training dataset, δ_{w,T_i} is one, if the label w is annotated in the training sample T_i, otherwise zero, and μ is a parameter between zero and one. Further, the posterior probability of multiple words $\boldsymbol{w}$ is expressed as follows.

$$P(\boldsymbol{w}|T_i) = \prod_{w \in \boldsymbol{w}} P(w|T_i). \tag{4.5}$$

As Equation 4.5 shows, each weak classifier formed by a sample treats word classes independently and does not consider their co-occurrence. However, as a result of the model definition of Equation 4.4, it gives large posterior probabilities for words that occur simultaneously in a sample. Therefore, it is expected that by averaging all weak classifiers as in Equation 4.1, our model can implicitly exploit the co-occurrence of labels in the training dataset.

4.2 Distance Metric Learning Using Probabilistic Canonical Correlation Analysis

4.2.1 Canonical Correlation Analysis

We extract p-dimensional image features $\boldsymbol{x}$, and q-dimensional label features $\boldsymbol{y}$ from the training dataset $\{T_i\}_{i=1}^N$. Then we have N training samples $\{\boldsymbol{x}_i, \boldsymbol{y}_i\}_{i=1}^N$. For a detailed explanation of CCA, refer to Section 3.2.2. Here, we define the terminology for our method.

CCA finds linear transformations $\boldsymbol{s} = U_x^T(\boldsymbol{x} - \bar{\boldsymbol{x}})$ and $\boldsymbol{t} = U_y^T(\boldsymbol{y} - \bar{\boldsymbol{y}})$ that maximize the correlation between the new values $\boldsymbol{s}$ and $\boldsymbol{t}$. Henceforth, we call $\boldsymbol{s}$ the image-side canonical variable, and $\boldsymbol{t}$ the label-side canonical variable. Furthermore, we call the corresponding feature spaces the image-side canonical space and label-side canonical space, respectively. Λ is the diagonal matrix of the first d ($\min\{p, q\} \geq d \geq 1$) canonical correlations in descending order. During the learning of the canonical spaces, image and label features work complementarily as the teaching signals for each other. As a result, we can obtain subspaces capturing essential features both in terms of appearance and semantics. It is expected that we can retrieve semantically similar samples using the structure of CCA.

The simplest way to exploit the structure of CCA is to compute the distance between samples in the image-side canonical space. We call this framework CCAsim [142; 227]. For the distance metric used in the k-nearest neighbor algorithm, we use the Euclidean distance.

$$DIST_{CCA}(I_Q, T_i) = \|U_x^T \boldsymbol{x}_Q - U_x^T \boldsymbol{x}_i\|. \tag{4.6}$$

This is equivalent to Equation 3.22.

For MAP classification, we define the posterior probability of each sample using a Gaussian distribution fitted to the query.

$$P_{CCA}(T_i|I_Q) = \frac{\exp\left(-\frac{1}{2}(\boldsymbol{s}_i - \boldsymbol{s}_Q)^T\Sigma^{-1}(\boldsymbol{s}_i - \boldsymbol{s}_Q)\right)}{\sum_{j=1}^N \exp\left(-\frac{1}{2}(\boldsymbol{s}_j - \boldsymbol{s}_Q)^T\Sigma^{-1}(\boldsymbol{s}_j - \boldsymbol{s}_Q)\right)}. \tag{4.7}$$

Here $\Sigma = \alpha I$ (where I is a unit matrix). The denominator is a regularization term such that $\sum_{i=1}^N P_{CCA}(T_i|I_Q) = 1$. α is a manually tuned parameter that determines the smoothness.

4.2.2 Probabilistic Canonical Correlation Analysis

CCA only gives linear transformations U_x, U_y and the corresponding two canonical spaces. It does not give any insight into the use of the canonical spaces or the distance

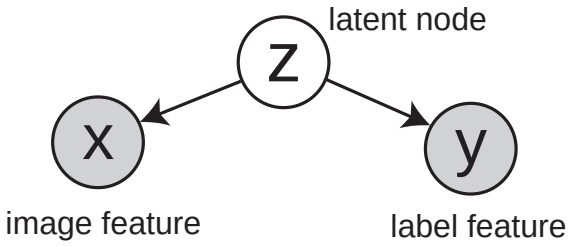


Figure 4.1: Graphical model of PCCA.

metric between samples. Therefore, we heuristically used the Euclidean distance in the image-side canonical space for $DIST_{CCA}$ and P_{CCA}.

In this respect, it has been proved that CCA has the following probabilistic structure [5] (Figure 4.1). This is called the probabilistic canonical correlation analysis (PCCA) model.

$$\begin{aligned} \boldsymbol{z} &\sim \mathcal{N}(0, I_d), \ \min\{p, q\} \geq d \geq 1, \\ \boldsymbol{x}|\boldsymbol{z} &\sim \mathcal{N}(W_x \boldsymbol{z} + \boldsymbol{\mu}_x, \Psi_x), W_x \in \mathcal{R}^{p \times d}, \Psi_x \geq 0, \\ \boldsymbol{y}|\boldsymbol{z} &\sim \mathcal{N}(W_y \boldsymbol{z} + \boldsymbol{\mu}_y, \Psi_y), W_y \in \mathcal{R}^{q \times d}, \ \Psi_y \geq 0. \end{aligned} \tag{4.8}$$

Here, $\mathcal{N}$ is a Gaussian. $\Psi_x \geq 0, \Psi_y \geq 0$ indicate that Ψ_x and Ψ_y are positive semi-definite matrices. $\boldsymbol{z}$ is an unobserved latent variable that generates $\boldsymbol{x}$ and $\boldsymbol{y}$ under the assumption of conditional independence. d is the dimension of $\boldsymbol{z}$ (the same value as in Equations 3.20 and 3.21). The maximum likelihood solution of this model basically corresponds to the solution of normal CCA. Specifically, $\boldsymbol{x}$ and $\boldsymbol{y}$ are first projected onto canonical variables $\boldsymbol{s}$ and $\boldsymbol{t}$ as in the normal CCA. PCCA further merges two canonical variables using canonical correlations as a mapping to $\boldsymbol{z}$. This mapping is performed in a probabilistic manner, giving a Gaussian as the posterior probability.

The details are given below. Let $M_x, M_y \in \mathcal{R}^{d \times d}$ denote arbitrary matrices such that $M_x M_y^T = \Lambda$ and the spectral norms of M_x and M_y are smaller than one.

If only an image feature $\boldsymbol{x}$ of the sample is given, $p(\boldsymbol{z}|\boldsymbol{x})$ becomes a Gaussian with mean $\dot{\boldsymbol{z}}$ and variance Φ_x defined as:

$$\dot{\boldsymbol{z}} = E(\boldsymbol{z} \mid \boldsymbol{x}) = M_x^T U_x^T (\boldsymbol{x} - \bar{\boldsymbol{x}}), \tag{4.9}$$

$$\Phi_x = var(\boldsymbol{z} \mid \boldsymbol{x}) = I - M_x M_x^T. \tag{4.10}$$

Similarly, if both an image feature $\boldsymbol{x}$ and a label feature $\boldsymbol{y}$ are given, we have

$$\ddot{\boldsymbol{z}} = E(\boldsymbol{z} \mid \boldsymbol{x}, \boldsymbol{y}) = \begin{pmatrix} M_x \\ M_y \end{pmatrix}^T \begin{pmatrix} (I - \Lambda^2)^{-1} & -(I - \Lambda^2)^{-1}\Lambda \\ -(I - \Lambda^2)^{-1}\Lambda & (I - \Lambda^2)^{-1} \end{pmatrix} \begin{pmatrix} U_x^T(\boldsymbol{x} - \bar{\boldsymbol{x}}) \\ U_y^T(\boldsymbol{y} - \bar{\boldsymbol{y}}) \end{pmatrix}, \tag{4.11}$$

$$\Phi_{xy} = var(\boldsymbol{z} \mid \boldsymbol{x}, \boldsymbol{y}) = I - \begin{pmatrix} M_x \\ M_y \end{pmatrix}^T \begin{pmatrix} (I-\Lambda^2)^{-1} & -(I-\Lambda^2)^{-1}\Lambda \\ -(I-\Lambda^2)^{-1}\Lambda & (I-\Lambda^2)^{-1} \end{pmatrix} \begin{pmatrix} M_x \\ M_y \end{pmatrix}. \quad (4.12)$$

M_x and M_y have arbitrary properties for scale and rotation. Here, we define them simply using the following diagonal matrices:

$$M_x = \Lambda^{\beta}, \; M_y = \Lambda^{1-\beta} \quad (0 < \beta < 1). \quad (4.13)$$

With this definition, Φ_x and Φ_{xy} are now diagonal. β is a parameter to balance the contributions of the image and label features in estimating the latent variable.

4.2.3 Proposed Method: Canonical Contextual Distance

As described above, a sample forms a Gaussian in the latent space. Using this structure, we can derive a probabilistically supported distance metric. We call this framework the canonical contextual distance (CCD) [138; 140; 226].

Since each training sample consists of an image and labels, there are two possible approaches. One considers only the image side in estimating the posterior probability distribution of the latent variable (1-view CCD), while the other considers both image and label sides (2-view CCD).

1-view CCD (CCD1)

As a distance metric for the k-nearest neighbor algorithm, let us consider the KL divergence between a query $\boldsymbol{x}_Q$ and a training sample $\{\boldsymbol{x}_i, \boldsymbol{y}_i\}$ in the latent space. When considering the x-view only (Figure 4.2(a)), this becomes:

$$\begin{aligned} DIST_{CCD1}(I_Q, T_i) &= KL\left(p(\boldsymbol{z}|\boldsymbol{x}_Q), p(\boldsymbol{z}|\boldsymbol{x}_i)\right) \\ &= (\dot{\boldsymbol{z}}_Q - \dot{\boldsymbol{z}}_i)^T \Phi_x^{-1} (\dot{\boldsymbol{z}}_Q - \dot{\boldsymbol{z}}_i). \end{aligned} \quad (4.14)$$

This can be computed as the Euclidean distance of

$$\boldsymbol{r}_{CCD1} = \Phi_x^{-1/2} \dot{\boldsymbol{z}}. \quad (4.15)$$

As the posterior probability used in MAP classification, we use the integration of

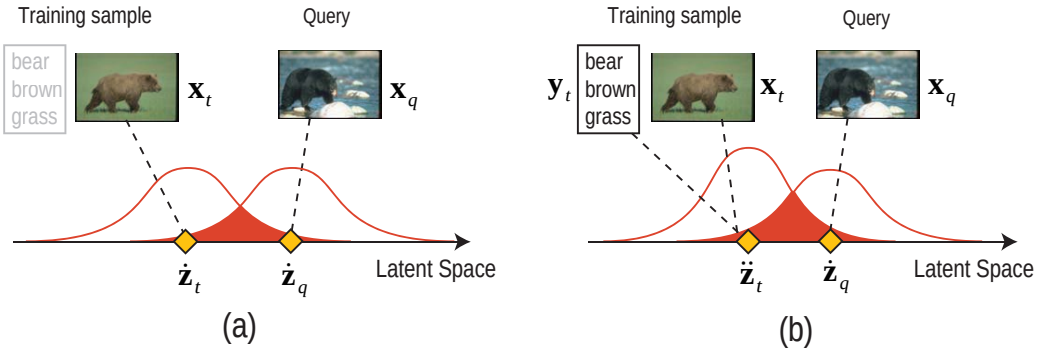


Figure 4.2: Illustration of canonical contextual distances. Estimation of distance between a query and training sample: (a) from the x-view only (CCD1); and (b) considering both the x- and y-views (CCD2).

the joint probability functions (the Bhattacharyya distance).

$$P_{CCD1}(T_i|I_Q) = \frac{\int \sqrt{p(z|\boldsymbol{x}_i)p(z|\boldsymbol{x}_Q)}dz}{\sum_{j=1}^{N}\int \sqrt{p(z|\boldsymbol{x}_j)p(z|\boldsymbol{x}_Q)}dz}$$
$$= \frac{\exp\left(-\frac{1}{8}(\dot{z}_Q - \dot{z}_i)^T\Phi_x^{-1}(\dot{z}_Q - \dot{z}_i)\right)}{\sum_{j=1}^{N}\exp\left(-\frac{1}{8}(\dot{z}_Q - \dot{z}_j)^T\Phi_x^{-1}(\dot{z}_Q - \dot{z}_j)\right)}, \tag{4.16}$$

The denominator is a regularization term, such that $\sum_{i=1}^{N} P_{CCD1}(T_i|I_Q) = 1$.

2-view CCD (CCD2)

Unlike CCD1, we explicitly consider the contribution of labels in each sample for the distance computation (Figure 4.2(b)). The KL divergence used in retrieving the k-nearest neighbors is:

$$\begin{aligned} DIST_{CCD2}(I_Q, T_i) &= KL\left(p(z|\boldsymbol{x}_Q), p(z|\boldsymbol{x}_i, \boldsymbol{y}_i)\right) \\ &= \frac{1}{2}\log\frac{|\Phi_{xy}|}{|\Phi_x|} - \frac{d}{2} + \frac{1}{2}\mathrm{Tr}(\Phi_{xy}^{-1}\Phi_x) + \\ &\quad (\ddot{z}_Q - \ddot{z}_i)^T\Phi_{xy}^{-1}(\ddot{z}_Q - \ddot{z}_i). \end{aligned} \tag{4.17}$$

Since the first three terms are constant, this can also be computed as the Euclidean distance, defining

$$\boldsymbol{r}^{Q}_{CCD2} = \Phi_{xy}^{-1/2}\dot{\boldsymbol{z}}_Q, \tag{4.18}$$

$$\boldsymbol{r}^{i}_{CCD2} = \Phi_{xy}^{-1/2}\ddot{\boldsymbol{z}}_i, \tag{4.19}$$

for a query and a training sample, respectively.

Similarly, the posterior probability used in MAP classification becomes:

$$\begin{aligned} P_{CCD2}(T_i|I_Q) &= \frac{\int \sqrt{p(z|\boldsymbol{x}_i,\boldsymbol{y}_i)p(z|\boldsymbol{x}_Q)}dz}{\sum_{j=1}^{N}\int \sqrt{p(z|\boldsymbol{x}_j,\boldsymbol{y}_j)p(z|\boldsymbol{x}_Q)}dz} \\ &= \frac{\exp\left(-\frac{1}{8}(\dot{\boldsymbol{z}}_Q-\ddot{\boldsymbol{z}}_i)^T\left(\frac{\Phi_x+\Phi_{xy}}{2}\right)^{-1}(\dot{\boldsymbol{z}}_Q-\ddot{\boldsymbol{z}}_i)\right)}{\sum_{j=1}^{N}\exp\left(-\frac{1}{8}(\dot{\boldsymbol{z}}_Q-\ddot{\boldsymbol{z}}_j)^T\left(\frac{\Phi_x+\Phi_{xy}}{2}\right)^{-1}(\dot{\boldsymbol{z}}_Q-\ddot{\boldsymbol{z}}_j)\right)}, \end{aligned} \tag{4.20}$$

4.3 Embedding Non-linear Metrics of Image Features

Although PLS, MLR, and CCA can perform semantic dimensionality reduction effectively, they have difficulty in dealing with specific features that have non-linear distance metrics. In fact, it is known that for many practically used image features, we should use a non-linear distance metric such as the χ^2 distance or L1 distance.

In this case, we first embed the non-linear metrics in a Euclidean space via kernel PCA (KPCA) [162]. Suppose a kernel function $K(\boldsymbol{x}_i,\boldsymbol{x}_j) = \langle\phi(\boldsymbol{x}_i),\phi(\boldsymbol{x}_j)\rangle$ is given, where $\phi : \boldsymbol{x} \rightarrow \phi(\boldsymbol{x})$ denotes the projection that maps an input vector onto a high-dimensional feature space. Using randomly sampled n_K ($n_K \leq N$) training samples, we compute the kernel base vector as

$$\boldsymbol{k}_x = (K(\boldsymbol{x},\boldsymbol{x}_1),...,K(\boldsymbol{x},\boldsymbol{x}_{n_K}))^T. \tag{4.21}$$

Using a kernel trick, the solution of KPCA becomes a linear problem on $\boldsymbol{k}_x$ coordinates. The embedded vector is obtained as $\tilde{\boldsymbol{x}} = B^T\boldsymbol{k}_x$, where B is the KPCA projection matrix. For the details, refer to Appendix B. We can use $\tilde{\boldsymbol{x}}$ as the new input for PLS, MLR, CCA, and CCD.

In our implementation, we use the exponentiated distance function. This is called the generalized RBF (GRBF) kernel, which has been reported to achieve good performance in many tasks [188; 219].

$$K(\boldsymbol{x}_i,\boldsymbol{x}_j) = \exp\left(-\frac{1}{2P}\text{dist}(\boldsymbol{x}_i,\boldsymbol{x}_j)\right). \tag{4.22}$$

Here, $\mathrm{dist}(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is a base distance, such as the χ^2 or L2 distance, and P is the mean of the base distances for the n_K training samples [219].

Theoretically, the larger n_K becomes, the better the performance is. In a standard approach, we may use all available training samples for a kernel trick ($n_K = N$). However, computing the kernel base of a query requires n_K raw training samples in memory. If n_K is large, this is computationally as expensive as a brute-force search in a raw feature space, thus destroying our objective. Moreover, the training phase requires solving an eigenvalue problem with dimension n_K[1], which is intractable when n_K is large.

Therefore, we randomly sample a small number of training samples ($n_K = 300$) for kernelization, and compute the eigenvalue decomposition of KPCA using all N samples. This approach is based on the idea of large-scale graph spectrum decomposition methods, such as the Nyström method [203] and column sampling [48].

Related to this topic, large-scale KPCA itself has been an active research area [176], since it is the most generic framework for manifold learning. For example, further improvements in the Nyström method [98; 112] and an efficient algorithm using additive kernels [151] have been proposed.

4.4 Label Features

Regarding label features, we use a binary vector indicating the presence of each word. Each element of the vector corresponds to one word. For example, if an image is annotated with "sky", "plane", and "cloud", the label feature becomes $(1, 0, 0, 1, 1)^T$, where the dictionary contains "sky", "sea", "mountain", "plane", and "cloud". The inner product of two label features is thus equal to the number of common words in the corresponding labels. Intuitively, this makes the Euclidean assumption on the feature space and application of linear methods reasonable. However, because of the sparsity of the label feature, the covariance matrix C_{yy} may become singular, which in turn may present problems with CCA. In this case, we can add regularization terms to make the eigenvalue problem stable. For example, C_{yy} can be replaced by $C_{yy} + \gamma I$, where γ is a small positive number.

4.5 Application to Keyword-based Image Retrieval

Keyword-based image retrieval is a promising application of image annotation. Having attached various keywords to unlabeled images through annotation, we can retrieve them in the same manner as the current text-based web search engines. However, for practical application, it is desirable to rank appropriate images higher. To do this,

[1] Generally, the computation complexity is $O(n_K^3)$.

simple keyword matching is not enough. We need to introduce a probabilistic framework to rank images according to their content. Here, we consider two approaches: maximum likelihood estimation and MAP estimation.

Let $\boldsymbol{w}_Q$ denote the query words. First, we describe the maximum likelihood estimation approach. Let g_l denote the likelihood of a candidate image I_c, then

$$g_l = P(\boldsymbol{w}_Q|I_c) \tag{4.23}$$

$$= \sum_{i=1}^{N} P(\boldsymbol{w}_Q|T_i)P(T_i|I_c). \tag{4.24}$$

This is also interpreted as the annotation score of $\boldsymbol{w}_Q$ for image I_C. We rank candidate images in descending order of g_l.

Similarly, we use the posterior probability for ranking in MAP estimation. Let g_{pp} denote the posterior probability of I_c for a given $\boldsymbol{w}_Q$, then

$$g_{pp} = p(I_c|\boldsymbol{w}_Q) \tag{4.25}$$

$$= \frac{P(\boldsymbol{w}_Q|I_c)p(I_c)}{P(\boldsymbol{w}_Q)} \tag{4.26}$$

$$= \frac{\left(\sum_{i=1}^{N} P(\boldsymbol{w}_Q|T_i)P(T_i|I_c)\right)p(I_c)}{P(\boldsymbol{w}_Q)} \tag{4.27}$$

$$\propto g_l p(I_c). \tag{4.28}$$

Note that $P(\boldsymbol{w}_Q)$ is constant for a given query $\boldsymbol{w}_Q$.

As shown, g_{pp} is the product of g_l and $p(I_c)$. The definition of $p(I_c)$ requires some prior knowledge and is a difficult problem. Since this problem is beyond the scope of this research, we assume it to be a constant value. In this case, retrieval based on g_{pp} corresponds to that based on g_l. In the remainder of this thesis, we perform image retrieval using the maximum likelihood estimation.

4.6 Discussion

4.6.1 Summary of Proposed Methods

The proposed methods (CCAsim, CCD1, and CCD2) all perform non-parametric image annotation using a subspace obtained via CCA. Compared to CCAsim, CCD (CCD1 and CCD2) exploits the PCCA scheme more strictly to obtain a better similarity measure. In CCAsim, projection of the input vector onto a subspace is done by a simple linear transformation. Then we use the Euclidean distance as the similarity measure in an ad-hoc manner. On the contrary, CCD performs projection in a probabilistic framework, and obtains a more discriminative similarity measure. For example, CCD

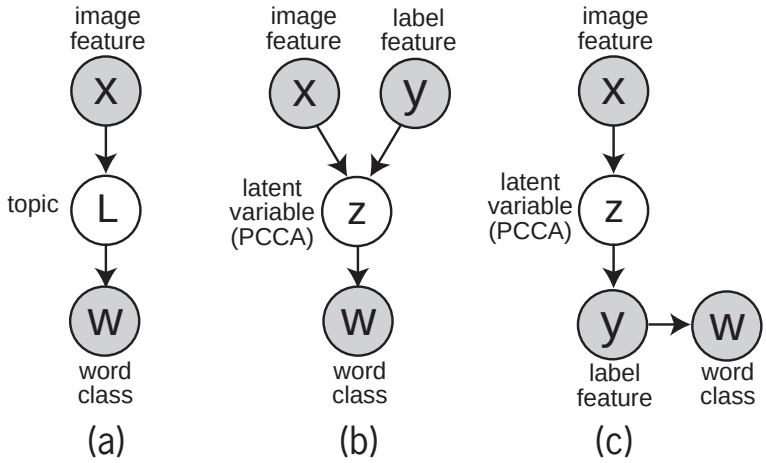


Figure 4.3: (a): Typical topic model approach. (b), (c): Approaches to the annotation problem using PCCA.

automatically weights each dimension of the latent variable according to its effectiveness, while CCAsim treats them the same.

Moreover, whereas CCAsim and CCD1 do not consider the contribution of labels in each sample during the distance computation, CCD2 explicitly utilizes both image and label features, resulting in a more powerful metric.

4.6.2 Relation to Other Methods Based on Topic Models

As is apparent from Figure 4.1, CCA (PCCA) has the same probabilistic structure as a topic model. In fact, PCCA can be interpreted as a special case of pLSA whose probability functions are defined by Gaussians. Generally, pLSA and LDA are formulized using multinomial distributions so that they can appropriately handle multiple words that provide symbolic information. Therefore, their topic model can be used directly for classification (Figure 4.3(a)). However, they need a sequential estimation, such as the EM algorithm or variational Bayes algorithm, for training, which is influenced by the initial parameters and often yields a local minimum. Moreover, since the training cost increases dramatically as the numbers of samples and words increase, it is barely tractable for the large web-scale datasets, which are the focus of this research. Meanwhile, because PCCA is based on a simple Gaussian model, we can obtain the global optimal solution in a short time. However, its topic model cannot be used directly as a classifier, because symbolic word information is not applicable to Gaussians.

In our method, as shown in Figure 4.3(b), we quantify label information to obtain the label feature beforehand, and then construct a topic model of PCCA using both image and label features. We roughly select essential features (dimensionality reduction)

in this stage. Next, we reuse the original word information in a sample-based approach, where each training sample is interpreted as a "topic." In this approach, each sample serves as a weak classifier. We model a Bayes optimal classifier by combining the weak classifiers according to their confidence values. Thus, our approach consists of two stages. First, semantic dimensionality reduction is performed using an intermediate representation (*i.e.* label features $\boldsymbol{y}$). Then, we build a non-parametric classifier in the latent space using $\boldsymbol{w}$.

As a straightforward approach to conducting annotation via CCA, it is possible to use the estimated label features output by the topic model directly (Figure 4.3(c)). Specifically, we can exploit $\hat{\boldsymbol{y}} = argmax \int p(\boldsymbol{y}|\boldsymbol{z})p(\boldsymbol{z}|\boldsymbol{x})d\boldsymbol{z}$ to design heuristic annotation rules. However, since CCA is a linear error minimization method, the above mentioned approach is basically equivalent to a simple linear regression. Although some previous works exploit linear CCA regression for the image annotation problem [74; 224], it is difficult for linear methods to explain adequately the complex relation between image and words in the generic problems targeted by image annotation. In many cases, kernelized methods are used to deal with non-linearity [74; 209]. However, naive kernelization can only embed the original generative distance metrics of features. Therefore, it is still difficult to model the correspondence of distributions.

On the contrary, our proposed method follows a practical approach. We can efficiently exploit the informative structure remaining in the latent space by a sample-based approach theoretically guaranteed by PCCA.

Chapter 5

Evaluation of Image Annotation Method

5.1 Datasets

We performed extensive experiments using four datasets. The Corel5K dataset [50] has been used as a benchmark for image annotation for a long time. In recent years, the IAPR-TC12 and ESP Game datasets have also frequently been used for evaluation [71; 125]. In addition to these, we also used the NUS-WIDE dataset [37], which is a relatively large-scale benchmark. Table 5.1 summarizes the statistics for each dataset.

Corel5K

The Corel5K dataset [50] has long been the de facto standard dataset for the problem of image annotation. This dataset contains 5000 pairs of an image and its labels. Each image has been manually annotated with an average of 3.4 keywords. 4500 samples are specified as the training data, while the remaining 500 samples are the test data. The dictionary contains 260 words.

IAPR-TC12

IAPR-TC12 was originally developed for the task of cross-lingual image retrieval. Makadia *et al.* [125] extracted common nouns from it and set up the current version for image annotation. Each sample is annotated with an average of 5.7 words from the 291 candidate words.

Table 5.1: Statistics of the training sets of the benchmarks.

	Corel5K	IAPR-TC12	ESP Game	NUS-WIDE
dictionary size	260	291	268	81
# of images	4,500	17,665	18,689	161,789
# of words per image (avg/max)	3.4/5	5.7/23	4.7/15	1.9/12
# of images per word (avg/max)	58.6/1004	347.7/4999	362.7/4553	3721.7/44255

ESP Game

The ESP Game dataset is a subset of an image-label database obtained from an online image labeling game [189]. It consists of 18,689 training samples and 2,081 test samples. This dataset includes not only real images, but also pictures and logos. We follow the same setup as in [71; 125].

NUS-WIDE

The NUS-WIDE dataset [37] is a comparatively large web image dataset, consisting of 161,789 training samples and 107,859 test samples downloaded from Flickr. All samples are supervised and labeled with 81 concepts. Note that many images in the dataset are "negative" and have no labels; that is, none of the 81 concepts appear within the images. We randomly sampled 2,000 "positive" images from the test samples and used these as our test data.

5.2 Basic Experiment

In this section, we discuss the effectiveness of CCD using the Corel5K, IAPR-TC12, and NUS-WIDE datasets. We compare CCD with other dimensionality reduction methods, and confirm its superiority in non-parametric image annotation. In addition, using various image features, we test both linear dimensionality reduction and that with KPCA embedding, to investigate whether these methods are effective.

5.2.1 Image Features

For the Corel5K and IAPR-TC12 datasets, we tested the following five image features.

1) Densely-sampled SIFT [120] bag-of-visual-words (BoVW) (1000 dim)

2) Densely-sampled Hue [185] BoVW (100 dim)

3) GIST [144] (512 dim)

4) HSV color histogram (4096 dim)

5) Higher-order local auto-correlation (HLAC) features (2956 dim)

Except for the HLAC features, all the features are employed in TagProp [71], and are available on the authors' web page[1]. For details of the HLAC features, refer to Appendix C.

For the NUS-WIDE dataset, we tested the following four image features.

1) Edge histogram (73 dim)

2) Color correlogram (144 dim)

3) Grid color moment (225 dim)

4) SIFT BoVW (500 dim)

These features are also provided by the authors of [37][2]. To provide baselines, we computed various base distances for each feature (*e.g.* χ^2 distance, L1 and L2 distances, histogram intersection).

5.2.2 Experimental Setup

Since our interest is in the performance of distance metrics for non-parametric image annotation, we simply use the k-nearest neighbor method with a brute-force search. The system outputs the most frequent labels in the k retrieved neighbors. We prioritize a rare label in the training dataset if the numbers of relevant neighbors are equal. With the Corel5K and IAPR-TC12 datasets, we tested $k = 1, 2, 4, 8, 16, 32$ and took the best performance. Similarly, we tested $k = 50, 100, 150, 200$ using the NUS-WIDE dataset.

As distance metrics, we evaluate both CCD1 (Equation 4.14) and CCD2 (Equation 4.17). Unlike CCD1, CCD2 explicitly considers the contribution of labels in the distance computation. To compare dimensionality reduction methods, we evaluated the following. In all these methods, the sample distance is computed in terms of Euclidean distance in the compressed subspace.

- PCA
- PCAW
- PLS
- nPLS

[1] http://lear.inrialpes.fr/data
[2] http://lms.comp.nus.edu.sg/research/NUS-WIDE.htm

- MLR
- nMLR
- CCA

Here, PCAW represents PCA with whitening, where the variance of the principal components is normalized. For the details of PLS (nPLS) and MLR (nMLR), refer to Section 3.2.2.

To evaluate annotation, we followed the methodology of a previous work [50]. The system annotates each test image with five words and computes the average of word-specific recall and precision. We finally used their F-measure as the annotation score. For more details, refer to Appendix A.

5.2.3 Experimental Results

We first report the results for the Corel5K and IAPR-TC12 datasets. In these small datasets, the dimension of image features is too large for MLR and CCA, both of which require the inverse of the covariance. Therefore, with the exception of Hue BoVW, we initially compressed visual features into 200-dimensional vectors using PCA, before computing MLR and CCA, CCD1, and CCD2. To embed non-linear metrics, we exploited the first 200 principal components of KPCA, and used these as the new image features. We placed the χ^2 distance into a kernel for SIFT BoVW and Hue BoVW (see Section 4.3). As for GIST, the L2 distance is empirically used as the base distance in many works. However, we placed the L1 distance in a kernel as it showed better performance in our experiment. Regarding HLAC, we did not perform kernelization since all possible base distances worked poorly.

Figures 5.1~5.10 show a comparison of annotation accuracy (F-measure). It is shown that bimodal dimensionality reduction methods such as PLS, MLR, and CCA improve the annotation score substantially compared with PCA. In many cases, nPLS and CCD exhibit superior performance, and achieve comparable or better performance than with the original L2 distance, using the first 10 or 20 dimensions only. If the Euclidean assumption of the feature space holds, we can expect both efficient compression and improvement of annotation accuracy. However, many practical image features are not embedded in a Euclidean space. In such cases, it is difficult for simple linear methods to compete with the original domain-specific metric, in terms of accuracy. This is especially true for the Hue BoVW and color histogram. In these cases, KPCA embedding works effectively and substantially improves the performance, although only a small fraction of training samples are used for kernelization (n_K=300). On the contrary, as has been proved in previous works, it is reasonable to assume a Euclidean space for GIST features. Thus, as illustrated in Figures 5.7 and 5.8, embedding

L1 distance does not substantially improve performance compared with normal linear methods.

Below, we summarize other knowledge and considerations.

- In linear methods, performance tends to be ordered CCD > nPLS > others. Sometimes nPLS outperforms CCD (for example, Figure 5.6), probably because PLS is a numerically stable method and works relatively well when ignoring the structure of the original non-linear manifold. In contrast, CCD always performs better when KPCA is applied. This result indicates that CCD is generally the best method in this framework when the Euclidean assumption holds.

- The performance of the CCA family is often ordered CCD2 > CCD1 > CCA. Because CCA assigns equal weights to all canonical features, performance sometimes declines rapidly as d becomes larger (for example, Figure 5.7). In contrast, CCD maintains good performance since it automatically weights latent features according to canonical correlations. Moreover, CCD2 generally outperforms CCD1, indicating the importance of considering the y-view at an instance level for distance computation.

- HLAC features show excellent performance compared to the other features. It should be noted that HLAC seems to be compatible with linear methods. It obtains high scores comparable with those of other features using KPCA embedding. Generally, HLAC works well with PCAW, MLR, and CCA (CCD). These methods all perform whitening of original features. However, when the original L2/L1 distance or variance preserving methods such as PCA and PLS are applied, performance is extremely low. This is due to the nature of HLAC where variances in the feature elements vary greatly. Although HLAC is a powerful feature, it should be used with care bearing this property in mind.

Next, we summarize the results for the NUS-WIDE dataset as illustrated in Figures 5.11~5.14. Since the provided features are normalized, we only investigate L1 and L2 as baseline distances, except for BoVW. Overall, the results are similar to those obtained with the Corel5K and IAPR-TC12 datasets. In particular, CCD shows a substantial improvement over the original distances in many cases, using only a dozen or so dimensions. However, unlike in the previous experiment, CCD1 and CCD2 perform almost equally. Although NUS-WIDE is a relatively large dataset, the label feature in this experiment consists of only 81 basic concepts. Our hypothesis is that, while this label feature is effective in the dimensionality reduction phase, it is too weak to contribute to the actual distance computation in the latent space.

Finally, we report actual computation times using the NUS-WIDE dataset. The training times for each method are summarized in Table 5.2. Target dimensionality was set at $d = 20$, and we used an 8-core Xeon 3.20 GHz processor for computation. PLS,

Table 5.2: Computation times for training the system on the NUS-WIDE dataset using each method[s]. We found that the differences in running times between PCA and PCAW, and between CCA and CCD are negligible for a small d.

	NUS-WIDE (161,789 samples, 81 words)			
	EDH (73 dim)	Cor. (144 dim)	C. mom. (225 dim)	BoVW (500 dim)
PCA (PCAW)	1.2	2.0	3.4	8.0
PLS	1.9	2.6	3.6	6.7
nPLS	3.5	5.2	7.4	14.6
MLR	2.0	2.7	4.0	8.3
nMLR	2.7	3.4	4.8	9.0
CCA (CCD)	2.1	3.0	4.5	10.1

MLR, and CCD can be computed with moderate additional time from PCA, although their annotation performance is improved. This is especially true when the dimension of the visual feature is much larger than the size of the vocabulary ($p \gg q$), which is explained in the analysis in Section 3.2.2. For example, PLS works faster than PCA in a 500-dimensional BoVW.

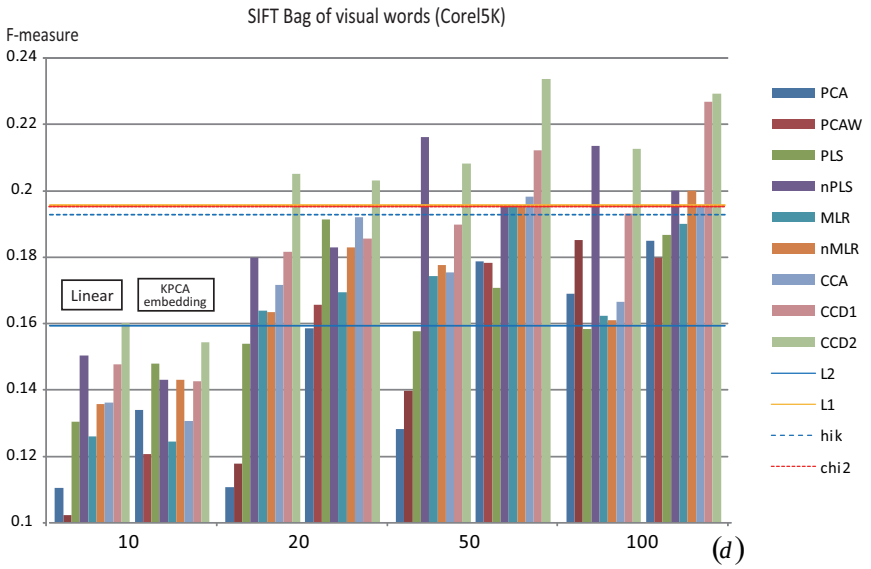


Figure 5.1: Results for the Corel5K dataset (1000-dimensional SIFT BoVW). Methods are compared using different features with designated dimensionality (d). For each entry, the left set of bars corresponds to normal linear methods, while the right set corresponds to those with KPCA embedding.

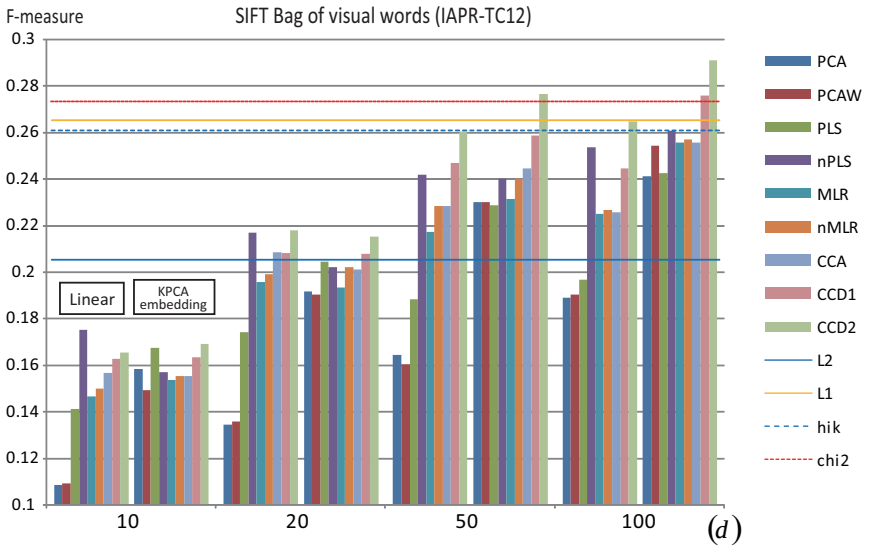


Figure 5.2: Results for the IAPR-TC12 dataset (1000-dimensional SIFT BoVW).

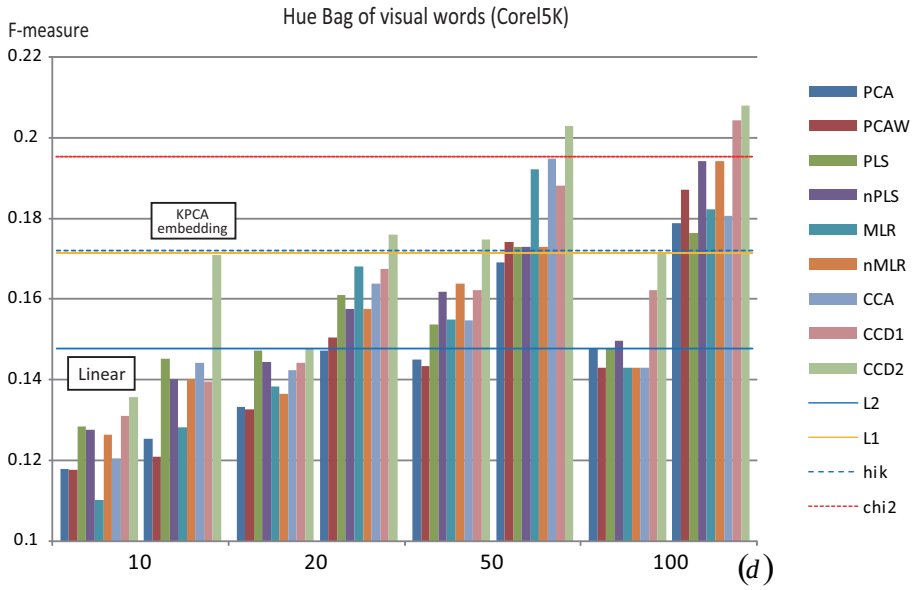


Figure 5.3: Results for the Corel5K dataset (100-dimensional hue BoVW).

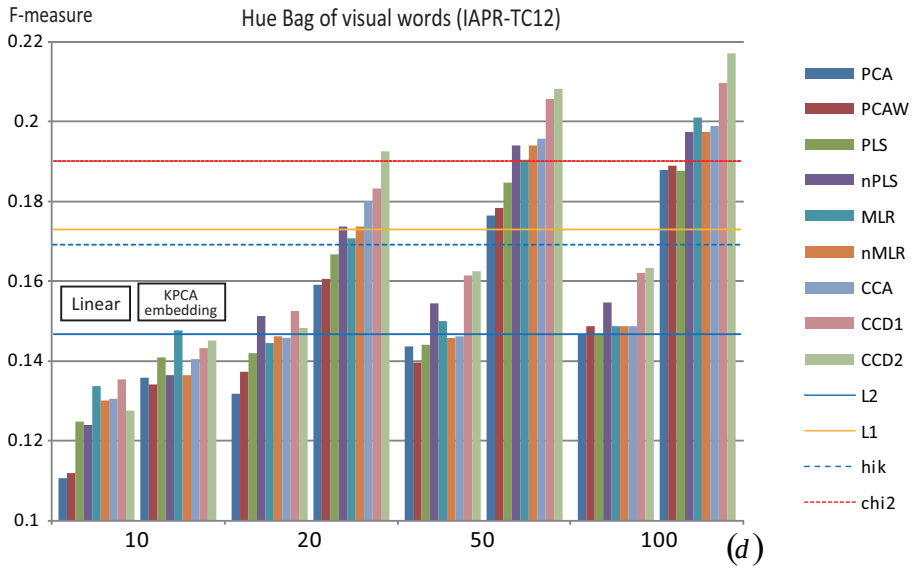


Figure 5.4: Results for the IAPR-TC12 dataset (100-dimensional hue BoVW).

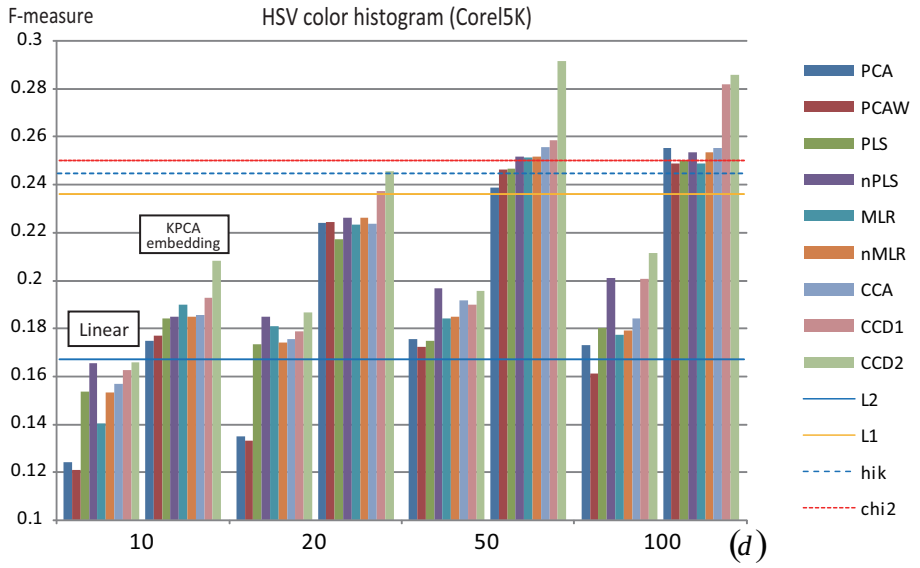


Figure 5.5: Results for the Corel5K dataset (4096-dimensional HSV color histogram).

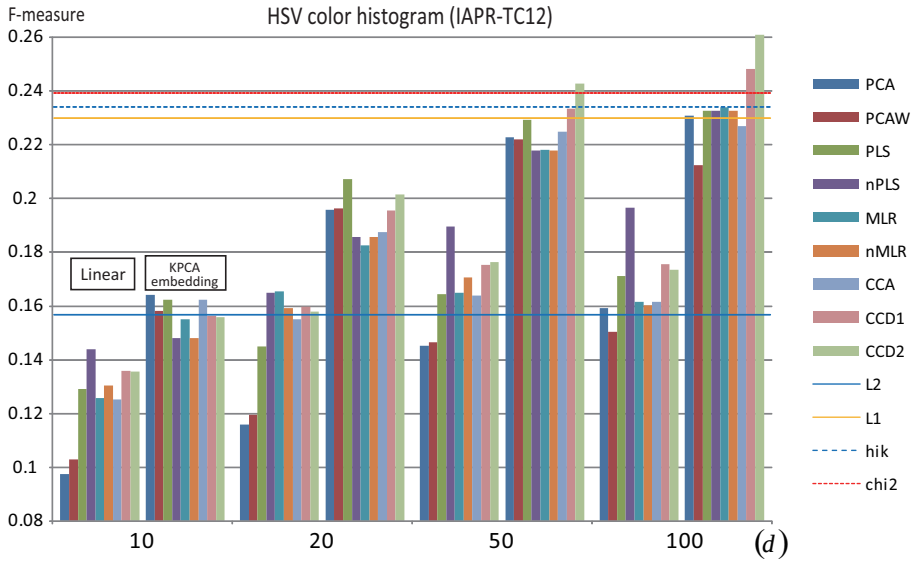


Figure 5.6: Results for the IAPR-TC12 dataset (4096-dimensional HSV color histogram).

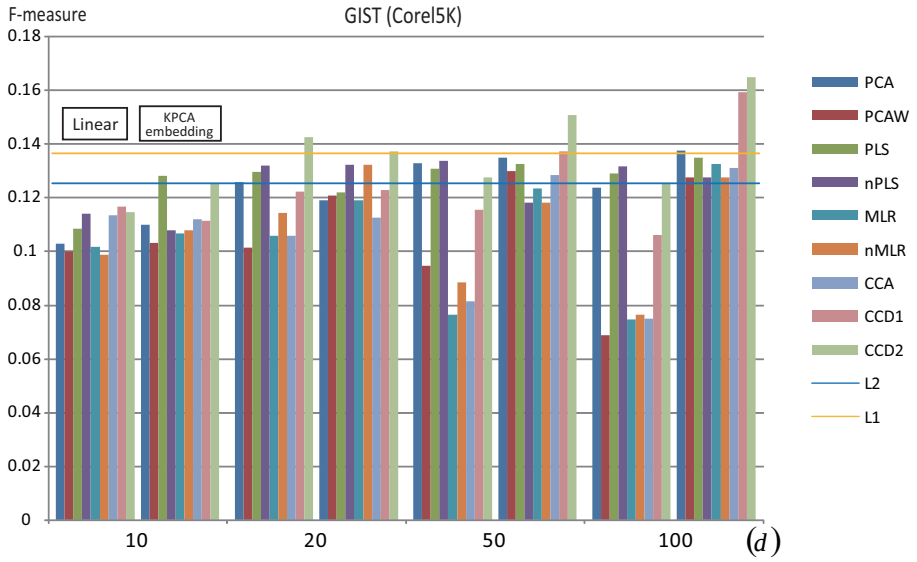


Figure 5.7: Results for the Corel5K dataset (512-dimensional GIST).

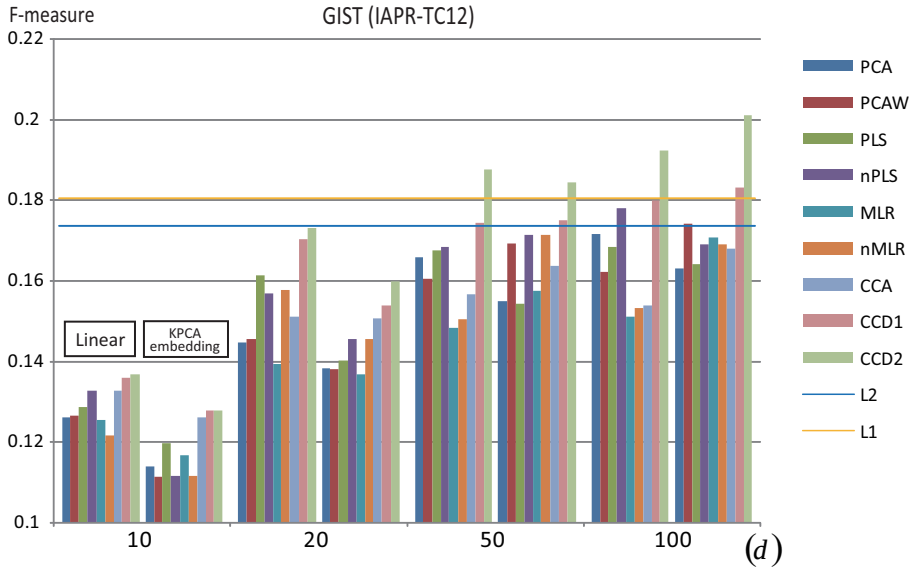


Figure 5.8: Results for the IAPR-TC12 dataset (512-dimensinal GIST).

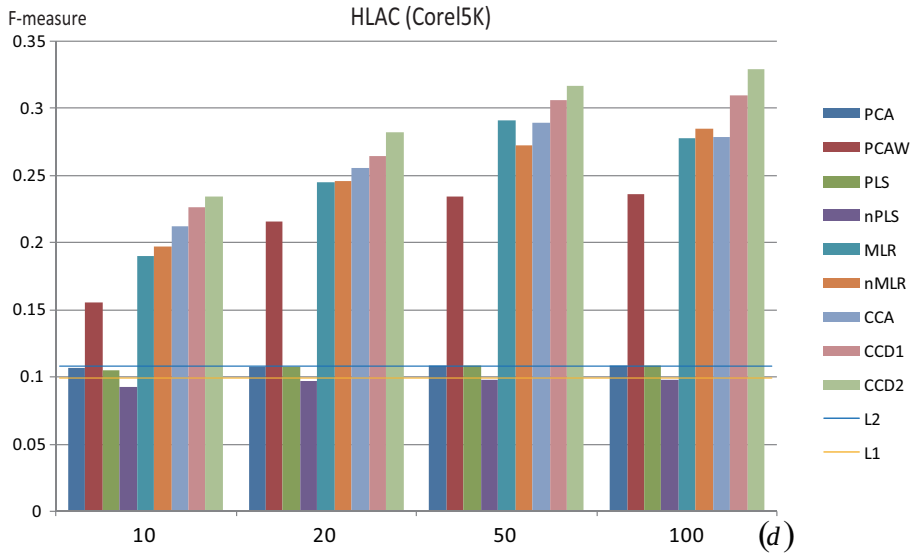


Figure 5.9: Results for the Corel5K dataset (2956-dimensional HLAC). Only linear methods are compared.

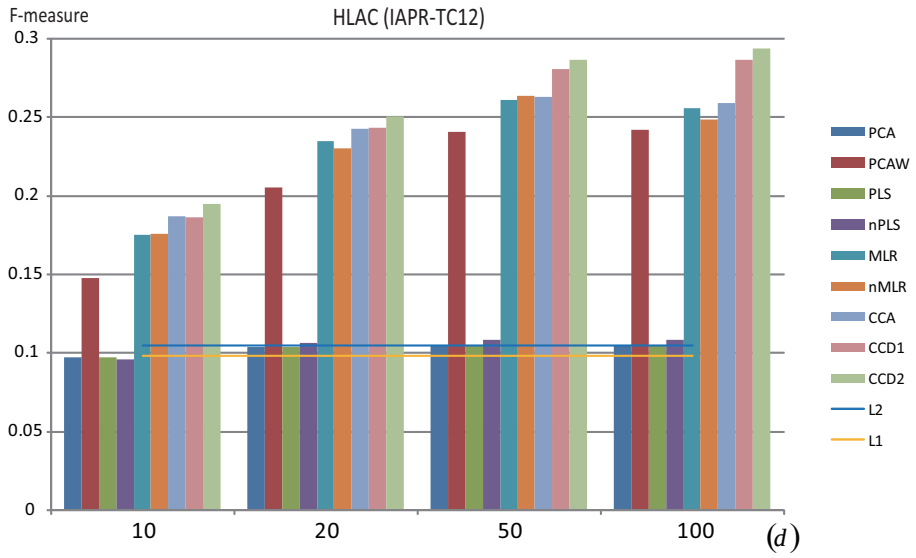


Figure 5.10: Results for the IAPR-TC12 dataset (2956-dimensional HLAC).

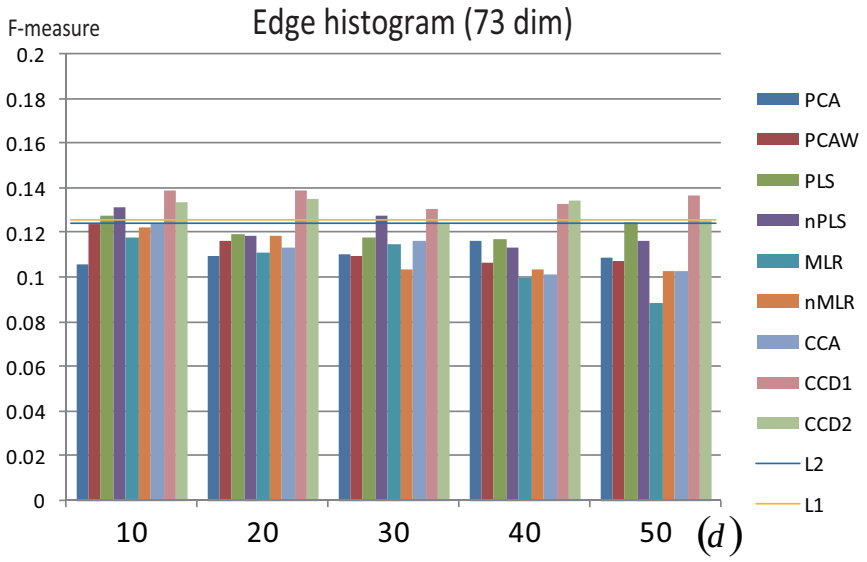


Figure 5.11: Results for the NUS-WIDE dataset (edge histogram). Methods are compared using different features with designated dimensionality (d).

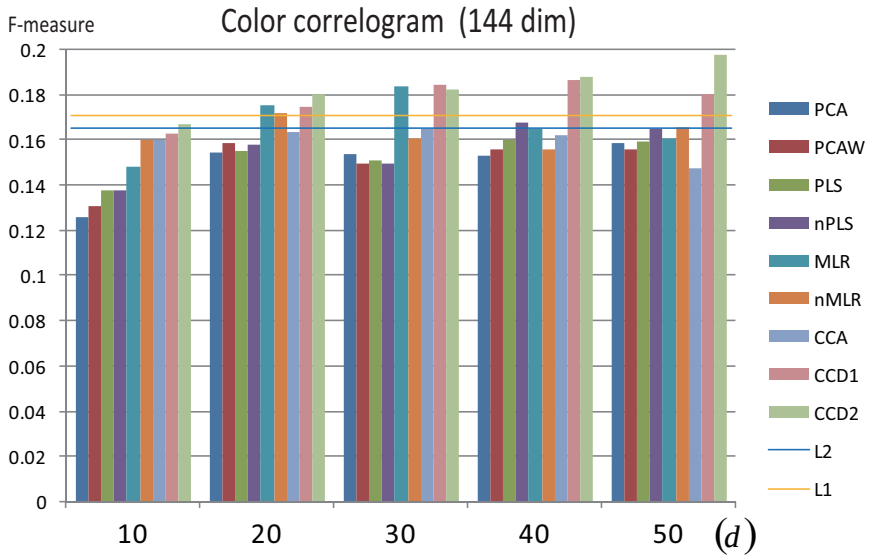


Figure 5.12: Results for the NUS-WIDE dataset (color correlogram).

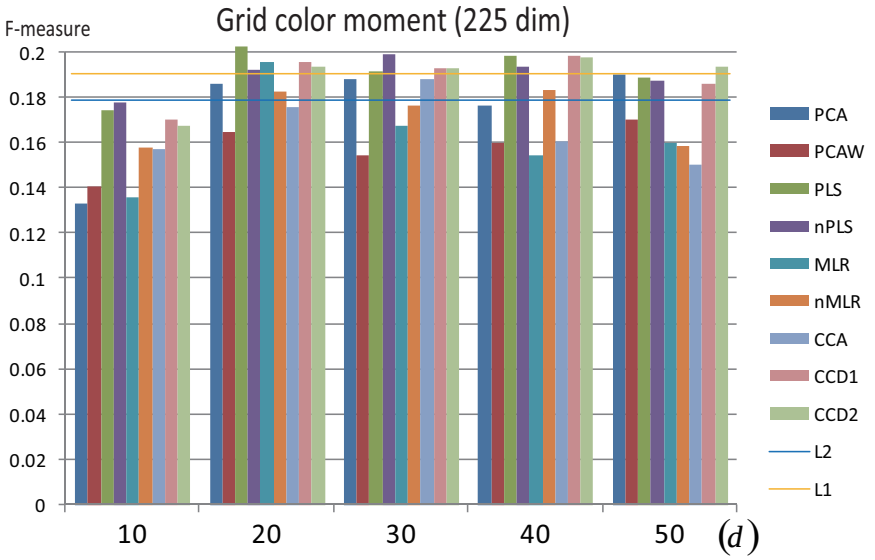


Figure 5.13: Results for the NUS-WIDE dataset (grid color moment).

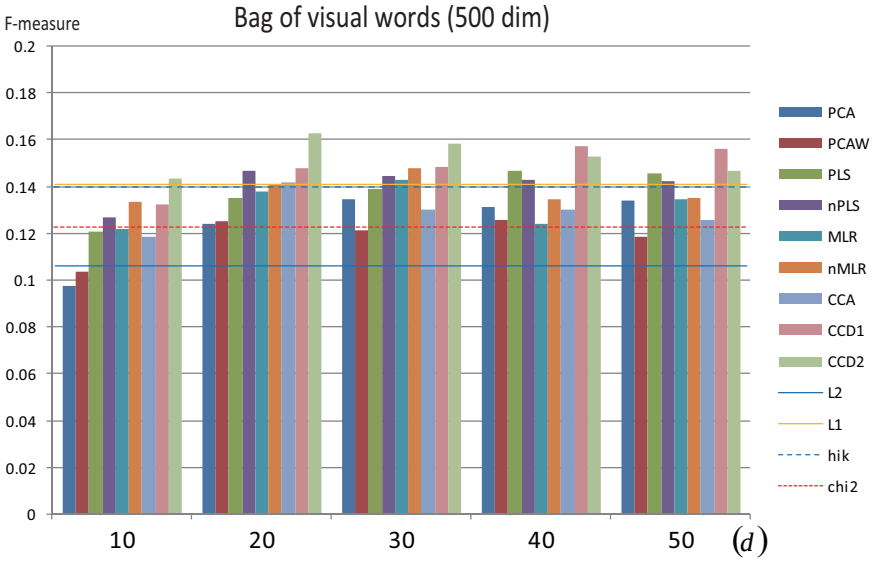


Figure 5.14: Results for the NUS-WIDE dataset (SIFT BoVW).

5.3 Comparison with Previous Research

Next, we compare our CCD-based image annotation method with previous research using the Corel5K, IAPR-TC12, and ESP game datasets. We implement our method with MAP classification because it can formulate the retrieval problem in a probabilistic manner.

5.3.1 Image Features

We investigate the following three cases.

(a) Only the HLAC feature is applied to the proposed method in a linear framework.

(b) Fifteen features from TagProp are concatenated with equal weights using kernels and embedded via KPCA.

(c) Fifteen features from TagProp are concatenated with different weights using kernels and embedded via KPCA.

The HLAC feature used in case (a) is the same as that used in the previous experiment (refer to Appendix C). Here, we explain cases (b) and (c). TagProp [71] uses the following image features.

1) SIFT bag-of-visual-words (dense sampling)

2) SIFT bag-of-visual-words (Harris detector)

3) Hue bag-of-visual-words (dense sampling)

4) Hue bag-of-visual-words (Harris detector)

5) RGB color histogram

6) HSV color histogram

7) LAB color histogram

8) GIST feature

Items 1, 3, 6, and 8 are the same as those used in the previous experiment. Further, for all features except item 8, TagProp also uses spatially partitioned versions. Therefore, it exploits 15 image features in total. In our method, we use the L1 distance for GIST

and the χ^2 distance for the other features to build GRBF kernels. Finally, these kernels are linearly combined for use in KPCA embedding.

$$K_{all} = \sum_{i=1}^{N_F} \alpha_i K_i, \tag{5.1}$$

where N_F is the number of features (here, $N_F = 15$), K_i indicates the kernel of the i-th feature, and α_i is its weight. Optimization of α_i is an important topic studied in the field of multiple kernel learning (MKL) [103]. In case (b), we give all kernels equal weights.

$$K_{all}^{average} = \frac{1}{N_F} \sum_{i=1}^{N_F} K_i. \tag{5.2}$$

In case (c), we optimize the weights in the MKL framework. Although MKL has been applied to KCCA [209], its computational cost is immense since it needs to solve an eigenvalue problem at each iteration. Therefore, we perform MKL through the task of label feature regression using support vector regression (SVR) [49]. Although the weights optimized by SVR are not directly related to annotation, we can roughly select important kernels. For an implementation of MKL SVR, we use the Shogun Library [172].

In the following sections, "Proposed (HLAC)" denotes case (a), "Proposed (15F: average+KPCA)" case (b), and "Proposed (15F: SVRMKL+KPCA)" case (c).

5.3.2 Experimental Results

Tables 5.3, 5.4, and 5.5 show the results for the Corel5K, IAPR-TC12, and ESP game datasets, respectively. For the definition of each score, refer to Appendix A. We use $n_K = 300$ base samples for kernelization. First, it is shown that Proposed (HLAC) achieves comparable performance with the state-of-the-art methods, except for TagProp. While these methods improve performance by using multiple features, our method obtains promising scores using only HLAC features. Moreover, our method further improves the annotation and retrieval performance when multiple features are utilized via KPCA.

Next, we show the relation between the number of base vectors n_K and performance in Figure 5.15. As illustrated, the more samples we use for kernelization, the better is the obtained score. Also, SVRMKL+KPCA (with optimized weights) generally outperforms average+KPCA (with equal weights). This is more evident when n_K is small. However, average+KPCA sometimes shows better performance when n_K is large. We observe that the key to improving the recognition accuracy is to use more base vectors. If we wanted to emphasize the computational cost for recognition, MKL with a small number of base vectors would be a good solution.

5.3. Comparison with Previous Research

Table 5.3: Performance comparison using Corel5K.

	MR	MP	F-m	N+	MAP	MAP (R+)
Co-occurrence [133]	0.02	0.03	0.02	19	-	-
Translation [50]	0.04	0.06	0.05	49	-	-
CMRM [88]	0.09	0.10	0.09	66	0.17	-
Maximum Entropy [89]	0.12	0.09	0.11	-	-	-
CRM [105]	0.19	0.16	0.17	107	0.24	-
NPDE [213]	0.18	0.21	0.19	114	-	-
InfNet [127]	0.24	0.17	0.20	112	0.26	-
CRM-Rectangles [59]	0.23	0.22	0.23	119	0.26	0.30
Independent SVMs [119]	0.22	0.25	0.23	-	-	-
MBRM [59]	0.25	0.24	0.25	122	0.30	0.35
AGAnn [117]	0.27	0.24	0.25	126	-	-
SML [29]	0.29	0.23	0.26	137	0.31	0.49
DCMRM [118]	0.28	0.23	0.26	135	-	-
TGLM [116]	0.29	0.25	0.27	131	-	-
MSC [191]	0.32	0.25	0.28	136	0.42	0.79
Matrix Factorization [119]	0.29	0.29	0.29	-	-	-
JEC [125]	0.32	0.27	0.29	139	0.33	0.52
JEC (15F) [71]	0.33	0.29	0.30	140	-	-
CBKP [122]	0.33	0.29	0.31	142	-	-
GS [220]	0.33	0.30	0.31	146	-	-
TagProp [71]	0.42	0.33	0.37	160	0.42	-
CCD (HLAC)	0.36	0.32	0.34	149	0.42	0.63
CCD (15F: average+KPCA, $n_K = 300$)	0.38	0.34	0.36	151	0.42	0.64
CCD (15F: SVRMKL+KPCA, $n_K = 300$)	**0.41**	**0.36**	**0.38**	**159**	**0.43**	**0.65**

Table 5.4: Performance comparison using IAPR-TC12.

	MR	MP	F-m	N+	MAP	MAP (R+)
MBRM [125]	0.23	0.24	0.23	223	0.24	0.30
JEC [125]	0.29	0.28	0.30	250	0.27	0.31
JEC (15F) [71]	0.19	0.29	0.23	211	-	-
TagProp [71]	0.35	0.46	0.40	266	0.40	-
GS [220]	0.29	0.32	0.30	252	-	-
CCD (HLAC)	0.26	0.35	0.30	249	0.32	0.38
CCD (15F: average+KPCA, $n_K = 300$)	0.28	0.43	0.34	**251**	0.37	0.43
CCD (15F: SVRMKL+KPCA, $n_K = 300$)	**0.29**	**0.44**	**0.35**	**251**	**0.39**	**0.44**

Table 5.5: Performance comparison using ESP game dataset.

	MR	MP	F-m	N	MAP	MAP (R+)
MBRM [125]	0.19	0.18	0.18	209	0.18	0.24
JEC [125]	0.25	0.22	0.23	224	0.21	0.25
JEC (15F) [71]	0.19	0.24	0.21	222	-	-
TagProp [71]	0.27	0.39	0.32	239	0.28	-
CCD (HLAC)	0.18	0.27	0.22	221	0.19	0.22
CCD (15F: average+KPCA, $n_K = 300$)	**0.24**	0.33	0.28	**236**	0.26	0.30
CCD (15F: SVRMKL+KPCA, $n_K = 300$)	**0.24**	**0.36**	**0.29**	232	**0.27**	**0.31**

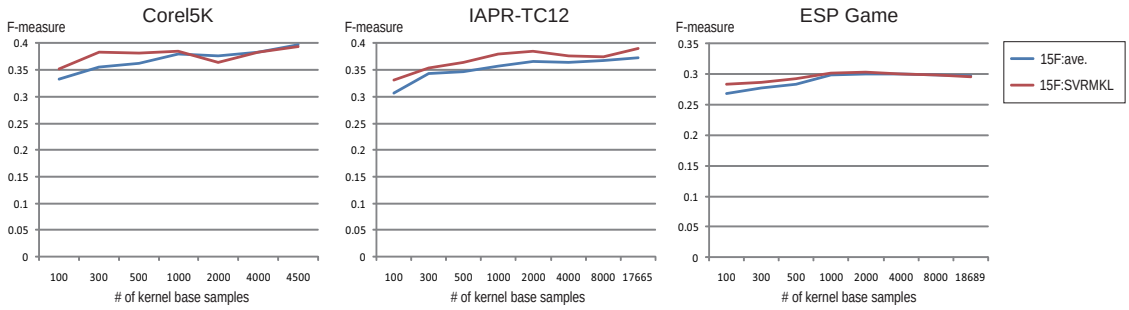


Figure 5.15: Annotation performance (F-measure) with a varying number of base samples for kernel PCA embedding.

Table 5.6: Comparison of annotation performance (F-measure) using TagProp.

	Corel5K	IAPR-TC12	ESP Game
TagProp ML	0.337	0.329	0.284
TagProp σML	0.369	**0.399**	**0.323**
CCD (HLAC)	0.341	0.297	0.217
CCD (15F: average+KPCA, $n_K = 300$)	0.355	0.342	0.277
CCD (15F: SVRMKL+KPCA, $n_K = 300$)	0.383	0.353	0.286
CCD (15F: SVRMKL+KPCA)	**0.394**	**0.391**	0.296

Finally, we give a detailed comparison of our method and TagProp [71]. TagProp owes its high recognition accuracy not only to the metric learning using multiple features, but also to the logistic discriminant model that relaxes the bias of training samples. As in [71], we let "TagProp ML" denote the case in which only metric learning is applied, while "TagProp σML" denotes the case in which the logistic discriminant model is added. We present the scores (F-measures) in Table 5.6. The proposed method, even with a small number of base vectors ($n_K = 300$), outperforms σML with Corel5K and TagProp ML with the other datasets. It is expected that we can further improve our method by incorporating a logistic discriminant model as in TagProp, although this is not within the current scope of this research.

5.3.3 Computational Costs

Table 5.7 summarizes the computational costs of our method and previous works, in terms of training and recognition. Considering that our research is aimed at large-scale applications, scalability of the number of training samples N is especially important. Here, we compare our method with JEC, GS, and TagProp, which are state-of-the-art methods.

Since these previous works all follow a sample based approach, their computational complexities for recognition are linear in the number of training samples. However, their actual computational costs differ because of the dimensionality of an instance. JEC and TagProp compute distances between the query and training samples in the image feature space, the cost of which is $O(pN)$, whereas the proposed method does this in the latent space of PCCA, the cost of which is $O(dN)$. Since $d \ll p$ in general, our method is faster than the other methods [1]. For example, TagProp uses 15 features,

[1] However, the cost is still prohibitive in large-scale problems, because we need to search all training samples in memory to perform non-parametric image annotation. Therefore, the sample representation should be as small as possible, while maintaining annotation accuracy. In Appendix E, we investigate this issue and develop an effective method using the technique of approximate nearest neighbor search.

Table 5.7: Comparison of computational costs against the number of samples. N is the number of whole training samples, while n_K is the number of those used for kernelization.

	Training	Annotation
JEC [125]	-	$O(pN)$
GS [220]	$O(N^2)$	$O(d_{GS}N)$
TagProp [71]	$O(N) \sim O(N^2)$	$O(pN)$
Proposed (linear)	$O(N)$	$O(dN)$
Proposed (KPCA embedding)	$O(N + n_K^3)$	$O(dN + pn_K)$

resulting in more than 37,000 dimensions. In contrast, the dimension of the latent space of our method is only 50∼100 in the current setup. However, when performing KPCA embedding, we need to compute distances in the image feature space to calculate kernel bases. Therefore, as n_K increases, the recognition speed of our method decreases substantially. Similarly, GS evaluates distances using d_{GS} features selected via group sparsity learning.

Next, we discuss the costs of training. JEC does not need a training phase because it is a simple k-NN based method. GS and TagProp need to compute all pairwise distances of training samples to optimize the weights of multiple features. Basically the computational complexity becomes $O(N^2)$, although in [71] a quasi-linear approximate method is implemented. CCA, which is the core of our method, consists of two steps, namely, computing covariance matrices, and solving the eigenvalue problem. For a fixed setup, the complexity is linear in the number of samples. Contrarily, solving KPCA requires $O(n_K^3)$ cost, which is intractable when n_K is large.

5.4 Discussion

We have confirmed that our method can achieve performance comparable with that of previous works using the standard benchmarks. Moreover, the computational cost of our method is substantially reduced for both training and recognition. This is because our method is primarily based on a linear learning method.

However, we also observed that annotation accuracy drops dramatically when our method is applied to certain image features that have non-linear distance metrics. One solution is to embed image features in a new Euclidean space using kernel methods. However, to realize good performance, we need to use many samples as bases for kernelization. The computational cost thereof increases significantly compared to the standard linear implementation, destroying the advantage of our method. This is a serious problem, since many practically used image features have non-linear metrics.

On the contrary, it is empirically shown that HLAC features are compatible with linear methods. In other words, merely applying HLAC features to the CCD framework, we can obtain performance comparable with that of other image features with kernelization. This is probably due to the nature of HLAC features having Euclidean properties to some extent.

Thus, it is extremely important to consider compatibility between learning methods and image features in developing image recognition systems. The investigation in this chapter suggests that we can realize scalable and accurate annotation methods using image features compatible with linear learning methods (*e.g.* HLAC features). In the next chapter, we develop a theoretical and generic framework to extract image features that satisfies the above mentioned condition.

Chapter 6

Development of Image Feature Extraction Scheme

In this chapter, we develop a new scheme for image feature extraction [139; 141]. The objective is to extract features compatible with linear methods such as CCD. More specifically, we can obtain high recognition accuracy merely by applying these features directly to linear methods.

6.1 Coding Global Image Features Using Local Feature Distributions

Image feature extraction can be roughly divided into the following two processes.

1. Extracting a number of local features.
2. Coding the extracted local features into a single global feature vector.

Both are important processes closely related to the performance of the final feature vector. Notably, recent works have shown that innovations in process 2 (coding) can substantially improve recognition performance [152; 195; 211; 222]. Moreover, coding methods generally determine the compatibility between features and classifiers. Therefore, we focus on the coding problem in this chapter. The key question for coding is how to efficiently exploit the statistical properties of the distribution of local features.

As discussed in detail in the following sections, the standard bag-of-visual-words (BoVW) can be interpreted as a sparse sampling of high-level statistics. In contrast, we propose the opposite approach: dense sampling of low-level statistics. We simply model a local feature distribution of each image as a Gaussian and introduce appropriate coding methods and distance metrics. Using the information geometry technique [4], we can derive a scalable and powerful linear approximation.

Table 6.1: Summary of previous work and our work from the viewpoint of local feature statistics.

	High-level	Low-level
Dense	Non-parametric [21] Gaussian mixture model [132; 221]	Covariance [183; 184] This work (single Gaussian)
Sparse	Bag-of-visual-words [40; 211]	

Despite its simplicity, our approach achieves satisfactory performance with several datasets. Furthermore, because our method and BoVW illustrate different statistical aspects, we can further improve classification performance by using both of these.

6.2 Related Work

In this section, we discuss previous research from the viewpoint of use of local feature distributions. Generally, local features employed in image recognition are high-dimensional. For example, the most well-known SIFT descriptor [120] has 128 dimensions. However, the number of statistically independent samples that can be extracted from one image is severely restricted. Therefore, estimating the distribution is an extremely difficult task. With this in mind, Table 6.1 summarizes the approaches of both previous work and this research. We classify each method by the complexity of models and coding sparsity.

6.2.1 Non-parametric Method

A straightforward approach is to use raw local features in a non-parametric manner without an explicit coding process. Boiman *et al.* [21] proposed the Naive-Bayes Nearest Neighbor (NBNN) classification algorithm, which finds the closest patch in the training corpus for all patches in the query image. This method showed excellent performance in 2008, probably because a non-parametric approach can handle the complex structure of real data in a relatively stable manner using a limited number of examples. However, the computational cost of this method is immense because all the raw local features in the training images must be preserved for use in classification. Therefore, although this method is scientifically significant, it is impractical for real problems.

6.2.2 Gaussian Mixtures

As an example of parametric estimation, Vasconcelos *et al.* exploited a Gaussian mixture model (GMM) to model the distribution [132; 187]. To apply their generative model to discrimination, they proposed a kernel function to define the similarity between two distributions, and used it on a support vector machine (SVM). These methods are interpreted as dense sampling of the high-level statistics of local feature distributions. Ideally, this gives an optimal representation of a distribution. However, as mentioned above, it is nearly impossible to estimate a large-scale GMM using local features sampled from each individual image. Therefore, in practice, the GMM is usually constructed using the entire training corpus and a distribution of each image is estimated as a deviation from it [149; 221; 223]. Zhou *et al.* [221] estimated a GMM for each image in this approach and used its parameters as appearance features. In addition, Perronnin *et al.* [149] represented each image using the Fisher score vector of a GMM of the dataset. While this approach shows superior performance, image representation depends heavily on the generative model of the training corpus. To deal with other tasks, we need to rebuild the GMM, which is computationally quite expensive.

6.2.3 Bag-of-Visual-Words

Bag-of-visual-words (BoVW) [40] is an application for image recognition of bag-of-words [126], which is a textual feature. It is the current de-facto standard approach in generic image recognition. The first step in this method is to perform a vector quantization of the local features of the training images using clustering algorithms to obtain centroids that represent the visual words. Usually, the k-means algorithm is used for clustering because of its computational efficiency. The resulting feature is a histogram of visual word occurrences in the image. This method is interpreted as exploiting only the mixing ratio of each Gaussian of the GMM. In this sense, it can be said that BoVW is a sparse sampling of high-level statistics.

Since BoVW can achieve promising recognition accuracy with relatively low computational costs, it has attracted much attention in the community. Despite its success, there are some major problems to be solved. The first of these is the codebook generation process, because the standard k-means algorithm tends to place its clusters around the densest regions in the training corpus. Many works have focused on this problem. For example, Jurie *et al.* [91] exploited a radius-based mean-shift clustering to generate a more appropriate codebook. Wu *et al.* [205] showed that a histogram intersection is generally a better metric for clustering local features. There are also many studies focused on improving BoVW related to other aspects. For example, the soft assignment strategy [150; 186], which assigns each local feature to several visual words, has been shown to create more descriptive visual word histograms. Recently, as a new breakthrough, it has been revealed that soft assignment using sparse coding

can achieve surprisingly high recognition accuracy [195; 211].

Of course, there are also many methods based on vector quantization other than those for clustering. For instance, Tuytelaars *et al.* [182] presented a lattice-based vector quantization instead of a data-driven approach. Shotton *et al.* [167] proposed a fast coding method using a random decision forest.

6.2.4 Covariance Descriptor

As an example of a method based on low-level statistics, Tuzel *et al.* proposed the covariance descriptor [183; 184], which is probably the closest to our method. They extracted a covariance matrix of the local features of an image, and described it as a point on a Riemannian manifold. Further, they performed LogitBoost learning using a tangent space of the manifold by means of differential geometry. They achieved excellent performance for a human detection task. This method can be interpreted as using the shape of a Gaussian to describe an image. Covariances are typical examples of low-level statistics and are expected to be relatively stable. However, since they are sampled from each image independently, an obvious problem is that they lose mean information. That is, two Gaussians at different points with similar shapes are indistinguishable. Moreover, because our method is based on a "flat" manifold, we can effectively exploit the structure of tangent spaces.

6.3 Proposed Method: Global Gaussian Approach

In our method, an image is represented as a Gaussian distribution of its local features. We call this the global Gaussian approach [141][1].

6.3.1 Coding Gaussian with Information Geometry

Suppose a bag of D-dimensional local features $\{\boldsymbol{v}_k\}$ are extracted from an image I_j. Then, I_j can be explained by the distribution $p_j(\boldsymbol{v}; \boldsymbol{\theta}(j))$ with $\boldsymbol{\theta}(j)$ as the parameters. We plot each sample on a flat Riemannian manifold using the information geometry technique. We derive some theoretically supported similarity metrics on the manifold and use these as kernel functions so that they are applicable to discrimination. As a natural result, it is shown that a theoretically optimal kernel is the one based on the Kullback-Leibler (KL) divergence. Basically, this kernel is the same as that used in [132], and is expected to provide the upper limit performance of our global Gaussian approach. However, the scalability of a KL divergence based method is low because it requires high-cost nonlinear computation. Therefore, we also derive a linear coding

[1] Here, "global" means that we fit a Gaussian over the entire local feature space. This is in contrast to the GMM and BoVW, which estimate local structures in the space.

that approximates the KL-divergence. This technique gives us an image feature vector compatible with linear methods, which is the final objective in this chapter.

6.3.2 Brief Summary of Information Geometry

Information geometry, which is based on differential geometry, began as the geometric study of statistical estimation [4]. It expresses the model space of a certain family of parametric probability functions as a Riemannian manifold. Each sample, which constitutes a probabilistic distribution, is represented as a point on the manifold. Let us consider the manifold S formed from a probabilistic model $p(\boldsymbol{v};\boldsymbol{\theta})$ with n-dimensional parameters $\boldsymbol{\theta} = (\theta^1, ..., \theta^n)$. An information geometry framework gives a statistically natural structure to the manifold. First, we exploit a Fisher information matrix as the Riemannian metric.

$$G^{\theta}_{lm}(\boldsymbol{\theta}) = E_{\boldsymbol{\theta}}\left[\frac{\partial \log p(\boldsymbol{v};\boldsymbol{\theta})}{\partial \theta^l}\frac{\partial \log p(\boldsymbol{v};\boldsymbol{\theta})}{\partial \theta^m}\right]. \tag{6.1}$$

The vicinity of each point on the manifold can be regarded as a Euclidean space. This is called a tangent space, where the inner product is defined by the Riemannian metric. Next, we apply a symmetric connection called an α-connection[1], with α as the parameter determining the structure of the manifold. For some specific probabilistic models, we find a flat manifold by taking an appropriate affine coordinate system ξ, in which tangent spaces are flatly connected. If such a coordinate ξ exists, the model space is defined as α-flat, and ξ is defined as the α-affine coordinate system. In an α-flat space, a geodesic is represented as a line on the α-coordinate system (α-geodesic). It is known that an α-flat space is always $-\alpha$-flat and that we can take another affine coordinate system that is dual to ξ. As discussed in more detail below, $\alpha = \pm 1$ becomes especially important in information geometry[2]. Actually, it is known that there exist ± 1-coordinate systems for many practical probabilistic models that are widely used in statistical learning. Therefore, information geometry has been successfully applied to the analysis and interpretation of many kinds of learning methods, such as the EM algorithm [3], boosting [137], and variational Bayes [84]. For further details, refer to [4].

The exponential family is among the most basic and important probabilistic models for practical applications. It also plays an important role in the information geometry framework. A distribution of the exponential family is represented as:

$$p(\boldsymbol{v};\boldsymbol{\theta}) = \exp\left(\sum_{i=1}^{n}\theta^i F_i(\boldsymbol{v}) - \psi(\boldsymbol{\theta}) + C(\boldsymbol{v})\right). \tag{6.2}$$

[1] $\alpha = 0$ corresponds to the Levi-Civita connection.

[2] In information geometry, terms such as 1-connection and 1-flat are specifically called e-connection and e-flat (e:exponential), respectively, while -1-connection and -1-flat are called m-connection and m-flat (m:mixture), respectively. However, we do not change the terminology in this paper for simplicity.

Here, $\boldsymbol{\theta}$ is the model parameter, F is a function of the observed variable $\boldsymbol{v}$, $\psi(\boldsymbol{\theta})$ is the potential function, and $C(\boldsymbol{v})$ is a constant function independent of $\boldsymbol{\theta}$. The exponential family is 1-flat, taking $\boldsymbol{\theta}$ as the corresponding affine coordinate system. We can take another affine coordinate system $\boldsymbol{\eta} = (\eta_1, ..., \eta_n)$, which is dual to $\boldsymbol{\theta}$ and is defined as $\eta_i = E_{\boldsymbol{\theta}}[F_i(\boldsymbol{v})]$. The $\boldsymbol{\eta}$-coordinate system is interpreted as the space of sufficient statistics and is -1-flat. The Riemannian metric of the $\boldsymbol{\eta}$-coordinate system becomes the inverse of that of the $\boldsymbol{\theta}$-coordinate system (G^{θ}, Equation 6.1). This can be explicitly described using the following conversion.

$$G^{\eta}_{lm} = \frac{\partial \theta^l}{\partial \eta_m}. \tag{6.3}$$

6.3.3 Gaussian Embedding Coordinates: Generalized Local Correlation (GLC)

A Gaussian also belongs to the exponential family and is described by $n = d+d(d+1)/2$ parameters. Let $\boldsymbol{\mu}$ and Σ denote the sample mean and covariance, respectively. Letting

$$\begin{gathered} C(\boldsymbol{v}) = 0, \quad F_i(\boldsymbol{v}) = v_i, \quad F_{ij}(\boldsymbol{v}) = v_i v_j \ (i \le j), \\ \theta^i = \sum_{j=1}^{D} (\Sigma^{-1})_{ij} \mu_j, \quad \theta^{ii} = -\frac{1}{2}(\Sigma^{-1})_{ii}, \quad \theta^{ij} = -(\Sigma^{-1})_{ij} \ (i < j), \end{gathered} \tag{6.4}$$

a Gaussian is represented as follows:

$$p(\boldsymbol{v}; \boldsymbol{\theta}) = \exp\left[\sum_{1 \le i \le D} \theta^i F_i(\boldsymbol{v}) + \sum_{1 \le i \le j \le D} \theta^{ij} F_{ij}(\boldsymbol{v}) - \psi(\boldsymbol{\theta}) \right]. \tag{6.5}$$

Here,

$$\psi(\boldsymbol{\theta}) = \frac{1}{2} \boldsymbol{\mu}^T \Sigma^{-1} \boldsymbol{\mu} + \frac{1}{2} \log(2\pi)^D |\Sigma|. \tag{6.6}$$

The $\boldsymbol{\eta}$-coordinates then become:

$$\eta_i = \mu_i, \quad \eta_{ij} = \Sigma_{ij} + \mu_i \mu_j \ (i \le j). \tag{6.7}$$

The $\boldsymbol{\theta}$-coordinates are based on the model parameters, while the $\boldsymbol{\eta}$-coordinates are based on sufficient statistics. In an ideal situation, where perfect information is obtained from the samples, we may take either of these as the image feature space. Usually, however, we have only a limited number of observations (local features) for each sample (an image). Therefore, we take the estimated sufficient statistics from the observations and plot each sample on the $\boldsymbol{\eta}$-coordinates. Let $\boldsymbol{e}_i, \boldsymbol{e}_{ij}$ denote the basis vectors

corresponding to η_i and η_{ij}, respectively. Then the $\boldsymbol{\eta}$-coordinate system is described as:

$$\begin{aligned}\boldsymbol{\eta} &= \sum_{1\leq i\leq D} \eta_i \boldsymbol{e}_i + \sum_{1\leq i\leq j\leq D} \eta_{ij}\boldsymbol{e}_{ij} \\ &= (\eta_1, ..., \eta_D, \eta_{11}, ..., \eta_{1D}, \eta_{22}, ...\eta_{2D}, ..., \eta_{DD})^T \\ &= (\hat{\mu}_1, ..., \hat{\mu}_D, \hat{\Sigma}_{11} + \hat{\mu}_1^2, ..., \hat{\Sigma}_{1D} + \hat{\mu}_1\hat{\mu}_D, \\ &\quad \hat{\Sigma}_{22} + \hat{\mu}_2^2, ..., \hat{\Sigma}_{DD} + \hat{\mu}_D^2)^T. \end{aligned} \tag{6.8}$$

As Equation 6.8 shows, the $\boldsymbol{\eta}$-coordinates consist of all the means and correlations of the elements of observed local features. Therefore, we call the $\boldsymbol{\eta}$-coordinate vector the generalized local correlation (GLC). The Riemannian metric of the $\boldsymbol{\eta}$-coordinate system is expressed as:

$$\begin{aligned} G^{\eta}_{ij} &= (\Sigma^{-1})_{ij}(1 + \boldsymbol{\mu}^T\Sigma^{-1}\boldsymbol{\mu}) + \\ &\qquad \sum_{k=1}^{D} \mu_k(\Sigma^{-1})_{ki} \sum_{k=1}^{D} \mu_k(\Sigma^{-1})_{kj}, \\ G^{\eta}_{i(pq)} &= -(\Sigma^{-1})_{pi} \sum_{k=1}^{D} \mu_k(\Sigma^{-1})_{kq} - \\ &\qquad (\Sigma^{-1})_{qi} \sum_{k=1}^{D} \mu_k(\Sigma^{-1})_{kp} \quad (p < q), \\ G^{\eta}_{i(pp)} &= -(\Sigma^{-1})_{pi} \sum_{k=1}^{D} \mu_k(\Sigma^{-1})_{kp} \\ G^{\eta}_{(pq)(rs)} &= (\Sigma^{-1})_{ps}(\Sigma^{-1})_{qr} + (\Sigma^{-1})_{qs}(\Sigma^{-1})_{pr} \\ &\qquad (p < q, r < s), \\ G^{\eta}_{(pq)(rr)} &= (\Sigma^{-1})_{pr}(\Sigma^{-1})_{rq} \quad (p < q), \\ G^{\eta}_{(pp)(rr)} &= \frac{1}{2}(\Sigma^{-1})^2_{pr}. \end{aligned} \tag{6.9}$$

In the above, the suffixes correspond to Equation 6.8. For example, $G^{\eta}_{i(pq)} = \langle \boldsymbol{e}_i, \boldsymbol{e}_{pq} \rangle$ and $G^{\eta}_{(pq)(rr)} = \langle \boldsymbol{e}_{pq}, \boldsymbol{e}_{rr} \rangle$.

6.3.4 Kernel Functions

KL divergence based kernel

Let P and Q denote the points on the manifold corresponding to distributions $f(\boldsymbol{v})$ and $g(\boldsymbol{v})$, respectively. In information geometry, the α-divergence between two points P and Q in a dually-flat space is defined as follows:

$$D^{(\alpha)}(P\|Q) = \psi(\boldsymbol{\theta}(P)) + \varphi(\boldsymbol{\eta}(Q)) - \sum_{i=1}^{n} \theta^i(P)\eta_i(Q). \tag{6.10}$$

Here, $\varphi(\boldsymbol{\eta})$ is the potential function of the $\boldsymbol{\eta}$-coordinate system. The α-divergence is an important metric for information geometry. Intuitively, it represents the dissimilarity between two points; strictly speaking, it is different from a mathematical distance, because a symmetric property does not hold unless P and Q are sufficiently close. Moreover, the dual $-\alpha$-divergence becomes $D^{(-\alpha)}(P\|Q) = D^{(\alpha)}(Q\|P)$. In the case of the exponential family, 1-divergence ($\alpha = 1$) is equal to the KL divergence between $f(\boldsymbol{v})$ and $g(\boldsymbol{v})$:

$$k(f\|g) = \int f(\boldsymbol{v}) \left[\log f(\boldsymbol{v}) - \log g(\boldsymbol{v})\right] d\boldsymbol{v}. \tag{6.11}$$

In addition, the dual -1-divergence ($\alpha = -1$) is equal to $k(g\|f)$. Since we take a -1-flat $\boldsymbol{\eta}$-coordinate system, we consider -1-divergence. However, since this is an asymmetric metric, we cannot use it directly as a kernel function. Therefore, we define a distance between two samples by symmetrizing the divergence following the approach of [132].

$$\begin{aligned} & dist(\boldsymbol{\eta}(P), \boldsymbol{\eta}(Q)) \\ = \; & D^{(-1)}(P\|Q) + D^{(-1)}(Q\|P) \\ = \; & k(g\|f) + k(f\|g) \\ = \; & tr(\Sigma_P \Sigma_Q^{-1}) + tr(\Sigma_Q \Sigma_P^{-1}) - 2D + \\ & tr\left((\Sigma_P^{-1} + \Sigma_Q^{-1})(\boldsymbol{\mu}_P - \boldsymbol{\mu}_Q)(\boldsymbol{\mu}_P - \boldsymbol{\mu}_Q)^T\right). \end{aligned} \tag{6.12}$$

To define a kernel that satisfies the Mercer conditions, we simply exponentiate the distance following [132]:

$$K_{kl}(P, Q) = \exp(-a \; dist(\boldsymbol{\eta}(P), \boldsymbol{\eta}(Q))), \tag{6.13}$$

where a is a smoothing parameter. KL divergence requires computing the inverse of a covariance matrix, which can be unstable when only a small number of local features are available. Therefore, we add a regularization matrix to the covariance matrices to improve numerical stability. That is, we let $\Sigma \to \Sigma + bI$. This process is equivalent to adding artificial white noise to local features.

Ad-hoc linear kernel

First, as the simplest baseline for linear approximation, we apply a linear kernel to the $\boldsymbol{\eta}$-coordinate system. This is a strong approximation that ignores the manifold metric, and is severely affected by the nature of local descriptors and scaling effects. We call this the ad-hoc linear (ad-linear) kernel.

$$K_{ad}(P, Q) = \boldsymbol{\eta}(P)^T \boldsymbol{\eta}(Q). \tag{6.14}$$

Center tangent linear kernel

For a stricter formulization, we need to exploit the Riemannian metric in Equation 6.9, which takes different values at each point of the $\boldsymbol{\eta}$-coordinates. Here, we simply use the metric for the mean, $\boldsymbol{\eta}_c = \frac{1}{N}\sum_i^N \boldsymbol{\eta}(i)$ for approximation. This is inspired by the initialization method of e(m)-PCA [2].

$$K_{ct}(P, Q) = \boldsymbol{\eta}(P)^T G^{\eta}(\boldsymbol{\eta}_c)\boldsymbol{\eta}(Q). \tag{6.15}$$

Here, $G^{\eta}(\boldsymbol{\eta}_c)$ is the metric for $\boldsymbol{\eta}_c$. This process is interpreted as approximating the manifold using the tangent space of $\boldsymbol{\eta}_c$. We call this the center tangent linear (ct-linear) kernel. The ct-linear kernel can be computed efficiently by applying a normal linear kernel to the transformed coordinate system:

$$\boldsymbol{\zeta} = (G^{\eta}(\boldsymbol{\eta}_c))^{1/2}\boldsymbol{\eta}. \tag{6.16}$$

As such, we can substantially improve the performance of the ad-hoc linear kernel without losing scalability.

6.4 Rigorous Evaluation using Kernel Machines

First, we confirm the effectiveness of our global Gaussian approach. Then, we compare the linear approximation derived from information geometry with theoretical upper bounds (KL divergence). For a fair comparison, we test each method using kernel machines in this section. Please note that learning with a linear kernel is basically equivalent to applying linear methods directly to the original feature space. This topic is discussed in the next section.

6.4.1 Datasets

We experimented with three challenging datasets: a 15 class scene dataset provided by Lazebnik *et al.* [106] (LSP15), an eight class sports events dataset provided by Li *et al.* [111] (8-sports), and a 67 class indoor scene dataset by Quattoni *et al.* [156] (Indoor67). Figure 6.1 illustrates various images from each dataset.

LSP15 is currently the standard benchmark for scene classification tasks. It consists of ten outdoor and five indoor classes. The 8-sports dataset has both scene recognition and object recognition aspects. Images in this dataset are characterized by background scenes with athletes in the foreground. Indoor67 is a new scene dataset published in 2009. It is characterized by a large number of classes and their high intra-class variations. Moreover, as pointed out in [156], indoor scene categorization is more difficult than natural scene categorization.

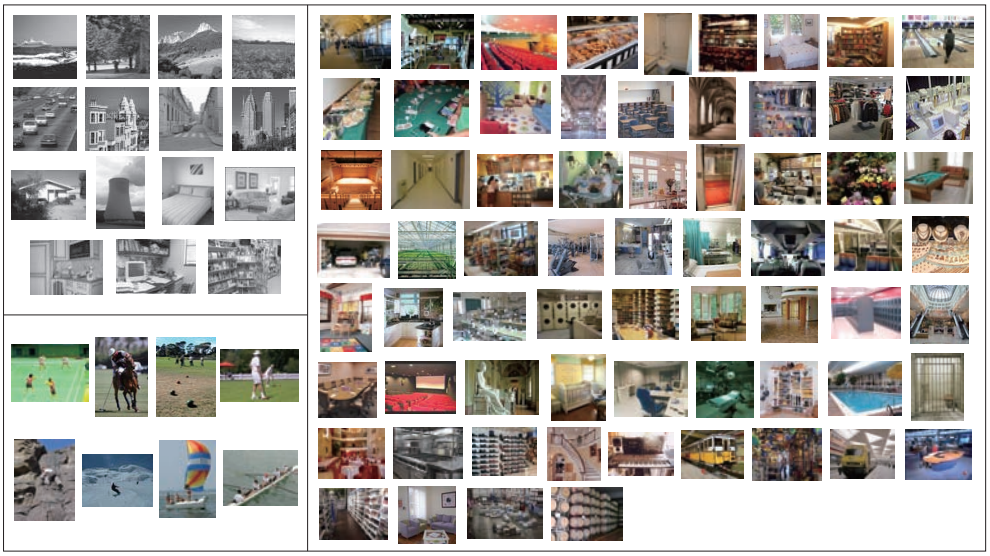


Figure 6.1: Images from benchmark datasets. Top left: LSP15 [106]. Bottom left: 8-sports [111]. Right: Indoor67 [156].

We followed the standard experimental protocols used in previous work. In LSP15, we randomly chose 100 training samples for each class and used the remaining samples for testing. Also, we randomly chose 70 training and 60 test samples from the 8-sports dataset, and 80 training and 20 test samples from the Indoor67 dataset. Performance was evaluated by the mean of the classification rate for each class[1]. This score was averaged over many trials, with the training and test samples randomly replaced. For all experiments in this section, we took the average over 10 trials.

6.4.2 Classification Methods

We employ two classification methods. The first is the SVM, which is a common tool for classification in recent work on generic image recognition. The other is probabilistic discriminant analysis (PDA) [86], a probabilistic interpretation of the classical linear DA. The benefit of PDA is that we can build a multiclass classifier by solving an eigenvalue problem only once, while SVM needs a fusion of binary classifiers. In both SVM and PDA, we apply kernel functions to cope with non-linearity. For SVM implementation, we use LIBSVM [33]. Below, we describe the implementation of the PDA classifier in detail.

[1] Average of diagonal elements in the confusion matrix.

Probabilistic discriminant analysis (PDA)

First, we introduce linear discriminant analysis (LDA)[1], which is the core of PDA. Let Σ_w denote the within-class covariance matrix, and Σ_b the between-class covariance matrix. LDA is formulated as the following generalized eigenvalue problem.

$$\Sigma_b W = \acute{\Sigma}_w W \Lambda \qquad (W^T \acute{\Sigma}_w W = I). \tag{6.17}$$

Here, $\acute{\Sigma}_w = \Sigma_w + \gamma I$. γ is a small positive number that decides the amplitude of the regularization matrix, which is used to prevent overfitting. W denotes the eigenvectors, while Λ is a diagonal matrix of corresponding eigenvalues (discriminant criterion) as the elements.

Let $t = N/K$ denote the number of samples in each class, and $\boldsymbol{\mu}_\eta$ denote the mean of an image feature over the entire dataset. The following projection maps an image feature $\boldsymbol{\eta}$ onto a point in the latent space:

$$\boldsymbol{u} = \left(\frac{t-1}{t}\right)^{1/2} W^T(\boldsymbol{\eta} - \boldsymbol{\mu}_\eta). \tag{6.18}$$

The covariance of the latent values is given by the following expression:

$$\Psi = \max\left(0, \frac{t-1}{t}\Lambda - \frac{1}{t}\right). \tag{6.19}$$

Using this structure, we classify a newly input sample $\boldsymbol{\eta}_s$ by maximum likelihood estimation. We assume that $\boldsymbol{u}_s$, the projected point of $\boldsymbol{\eta}_s$, is generated from a certain class C with probability:

$$p(\boldsymbol{u}_s | \boldsymbol{u}^C_{1\ldots t}) = \mathcal{N}\left(\boldsymbol{u}_s | \frac{t\Psi}{t\Psi + I}\bar{\boldsymbol{u}}^C,\ I + \frac{\Psi}{t\Psi + I}\right). \tag{6.20}$$

Here, $\boldsymbol{u}^C_{1\ldots t}$ are latent values of t independent training samples that belong to class C, and $\bar{\boldsymbol{u}}^C$ is their mean. We classify $\boldsymbol{\eta}_s$ as the class with the largest value for Equation 6.20. This is an extremely simple process similar to the nearest-centroid approach.

Kernelized PDA

Kernel discriminant analysis (KDA) is interpreted as performing linear DA on an implicit high-dimensional space using the kernel trick. Therefore, we can exploit the structure of PDA in the same manner. We call this KPDA.

Suppose a kernel function $K(\boldsymbol{\eta}(i), \boldsymbol{\eta}(j)) = \langle \phi(\boldsymbol{\eta}(i)), \phi(\boldsymbol{\eta}(j)) \rangle$ is given, where $\phi : \boldsymbol{\eta} \to \phi(\boldsymbol{\eta})$ denotes the projection that maps an input vector onto a high-dimensional feature

[1] In this chapter, LDA denotes linear discriminant analysis, and not latent Dirichlet allocation.

space. Let N denote the number of training samples, $\boldsymbol{\eta}^K = (K(\boldsymbol{\eta}, \boldsymbol{\eta}(1)), ..., K(\boldsymbol{\eta}, \boldsymbol{\eta}(N)))^T$ the kernel base vector, Σ_w^K the within-class covariance matrix of the kernel base vectors, and Σ_b^K the between-class covariance matrix. KDA is formulated as the following generalized eigenvalue problem.

$$\Sigma_b^K V = \acute{\Sigma}_w^K V \Lambda^K \qquad (V^T \acute{\Sigma}_w^K V = I). \tag{6.21}$$

$\acute{\Sigma}_w^K$, V, and Λ^K are defined in the same manner as in Equation 6.17.

Similarly, the projection is obtained as follows:

$$\boldsymbol{u}^K = \left(\frac{t-1}{t}\right)^{1/2} V^T(\boldsymbol{\eta}^K - \boldsymbol{\mu}_\eta^K). \tag{6.22}$$

$\boldsymbol{\mu}_\eta^K$ denotes the mean of kernel base vectors of training samples. Finally, we use the same classification rule as Equation 6.20 to classify the test samples.

6.4.3 Experimental Setup

Local feature sampling

Generally, local feature extraction involves two steps. The first is keypoint detection, while the second is feature description at the keypoints.

For keypoint detection, a visual saliency based approach, such as corner detection [75] and Difference of Gaussian filters [120], has been used for a long time. However, for image classification, keypoint detection based on filters does not always work effectively because salient points in terms of low-level image patterns are not necessarily related to semantic meanings. Nowak *et al.* [143] compared image classification performance achieved by various keypoint detection methods on several datasets. They showed that random keypoint detection achieved the best performance and identified that the most important factor for discrimination is the number of local features extracted from images. Fei-Fei *et al.* [57] performed classification on a 13 scene image dataset, and showed that grid-based keypoint detection gave the best performance. Considering these results, we perform keypoint detection based on a grid (possibly for every pixel). This strategy is called dense sampling and is used widely in the field of generic image recognition [25; 57; 143; 205; 221]. Specifically, we space the keypoints five pixels apart, and extract local features from each patch of 16×16 pixels with the keypoint at the center. Note that we extract local features from gray images in all experiments in this section, even if color images are available.

As for local feature descriptors, we use a SIFT [120] descriptor (128-dim) and a SURF [10] descriptor (64-dim). Mikolajczyk *et al.* [128] showed that the SIFT descriptor has the best performance on average of all local feature descriptors. Despite its computational cost being substantially reduced, SURF is known as a powerful descriptor comparable with SIFT.

Use of Spatial Information

We incorporate the spatial information of images into our kernels following the standard spatial pyramid kernel [106]. We hierarchically partition images into grids using the zeroth layer (original image) to the L-th layer. Each l-th layer ($0 \leq l \leq L$) is partitioned into a $2^l \times 2^l$ grid. Then, we generate the local $\boldsymbol{\eta}$-coordinate system independently for each region and compute kernels such as K_{kl} or K_{ct}. Finally, these are merged as follows:

$$K^{GG}(P, Q) = \frac{1}{\sum_{i=0}^{L} \beta^i} \sum_{l=0}^{L} \frac{\beta^l}{2^{2l}} \sum_{k=1}^{2^{2l}} K^{(l,k)}(P, Q). \tag{6.23}$$

Here, $\beta \in \mathcal{R}$ is the relative weight parameter of the layers. The suffix (l, k) indicates that the element belongs to the k-th region of the l-th layer.

As for the implementation of the K_{ct} kernel, since computing the metric for each region is expensive, we simply use the one from $L = 0$ for all regions.

Bag-of-Visual-Words implementation

To provide a quantitative baseline, we implement the BoVW method using the same local features sampled for the proposed method. We use the standard k-means method to generate a codebook and set the number of visual words to 200 and 1000. To train classifiers, we use a histogram intersection kernel and apply spatial pyramid matching [106]. Henceforth, K^{BoVW} denotes this kernel function.

In some experiments, we merge our proposed kernels (Equation 6.23) and those for BoVW to further improve the performance. Here, we simply exploit a linear combination.

$$K^{GG+BoVW} = \frac{1}{1+\kappa} K^{GG} + \frac{\kappa}{1+\kappa} K^{BoVW}, \tag{6.24}$$

where κ is a weight parameter. Note that the value of κ may not intuitively quantify the importance of each kernel because K^{GG} is not normalized, while the upper limit of K^{BoVW} is one.

6.4.4 Experimental Results

In depth study using LSP15 and the 8-sports datasets

First, we investigate the effectiveness of our global Gaussian approach using the LSP15 and 8-sports datasets. Table 6.2 gives the basic performance without the use of spatial information. We tested both SIFT and SURF descriptors. The notation "ad-linear" denotes the ad-hoc linear kernel, "ct-linear" the center tangent linear kernel, and "KL div." the KL divergence based kernel. As shown, the KL divergence based kernel

Table 6.2: Basic results of the global Gaussian approach with the LSP15 and 8-sports datasets using different kernels (%). No spatial information is used here.

		LSP15		8-sports	
		SIFT	SURF	SIFT	SURF
KPDA	ad-linear	77.3	75.9	77.9	72.4
	ct-linear	78.8	78.5	79.7	78.1
	KL div.	80.4	**81.5**	**81.7**	79.6
SVM	ad-linear	69.9	72.1	70.6	70.2
	ct-linear	75.7	77.7	75.5	73.3
	KL div.	76.3	**78.3**	**78.3**	74.9

Table 6.3: Performance comparison with spatial information for LSP15 (%). The SURF descriptor is used.

		L=0	L=1	L=2
GG	KPDA (ad-linear)	75.9	78.8	79.8
	KPDA (ct-linear)	78.5	81.6	82.3
	KPDA (KL div.)	**81.5**	**84.8**	**86.1**
	SVM (ad-linear)	72.1	73.2	74.3
	SVM (ct-linear)	77.7	80.1	80.7
	SVM (KL div.)	78.3	82.2	83.1
BoVW200	KPDA	71.9	78.5	81.1
	SVM	70.6	76.3	78.6
BoVW1000	KPDA	77.1	80.7	82.5
	SVM	74.9	78.0	79.4

achieves the best performance, followed by ct-linear and ad-linear. The ct-linear kernel substantially improves performance compared to the ad-linear kernel, while PDA achieves better performance than the SVM (LIBSVM). In addition, the results show that SURF is superior for LSP15, while SIFT is superior for the 8-sports dataset.

Next, we investigate the effect of spatial information on our method. Here, we implement BoVW to provide a baseline. In both our method and BoVW, we use spatial pyramids up to $L = 2$. Based on the results of Table 6.2, we use SURF for LSP15 and SIFT for the 8-sports dataset. Tables 6.3 and 6.4 give the results for LSP15 and the 8-sports dataset, respectively. Our method yields satisfactory results that compare well with the BoVW using 1000 visual words. Furthermore, the results show that spatial information can reasonably improve the performance of our method.

Finally, we attempt to merge our global Gaussian approach with BoVW. Although

Table 6.4: Performance comparison with spatial information for the 8-sports dataset (%). The SIFT descriptor is used.

		L=0	L=1	L=2
GG	KPDA (ad-linear)	77.9	79.3	80.2
	KPDA (ct-linear)	79.7	81.5	82.9
	KPDA (KL div.)	**81.7**	**83.2**	**84.4**
	SVM (ad-linear)	70.6	71.6	71.7
	SVM (ct-linear)	75.5	77.2	78.8
	SVM (KL div.)	78.3	80.2	81.4
BoVW200	KPDA	72.0	76.9	79.6
	SVM	71.7	76.3	77.7
BoVW1000	KPDA	77.8	80.6	81.5
	SVM	76.2	78.1	79.1

Table 6.5: Performance of the global Gaussian, BoVW, and combined approach (%). An $L = 2$ spatial pyramid is implemented. Kernel PDA is used for classification. The SURF descriptor is used for LSP15, while the SIFT descriptor is used for the 8-sports dataset.

	LSP15	8-sports
GG (KL)	**86.1**±0.5	**84.4**±1.4
GG (ct-linear)	82.3±0.4	82.9±1.0
BoVW200	81.1±0.7	79.6±1.1
BoVW1000	82.5±0.7	81.5±1.7
GG (ct-linear) + BoVW200	85.0±0.5	83.2±0.9
GG (ct-linear) + BoVW1000	85.3±0.5	83.4±0.7

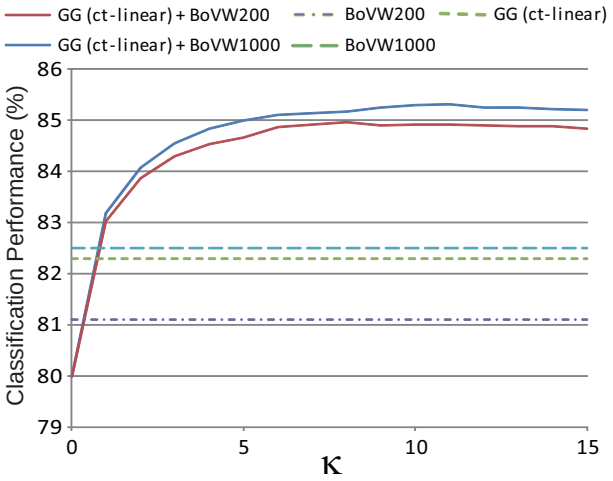


Figure 6.2: Merging the global Gaussian and BoVW approaches for use with the LSP15 dataset. κ is the parameter for weighting the kernels (Eq. 6.24).

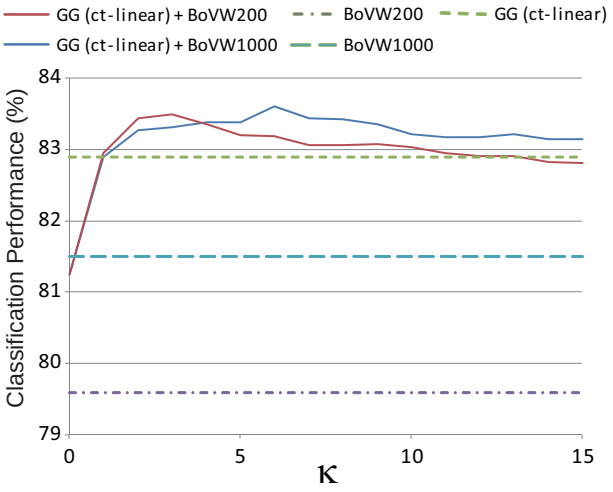


Figure 6.3: Merging the global Gaussian and BoVW approaches for use with the 8-sports dataset. κ is the parameter for weighting the kernels (Eq. 6.24).

Table 6.6: Performance comparison with previous work (%). For our method, an $L = 2$ spatial pyramid is implemented, and kernel PDA is used for classification. We used the SURF descriptor for LSP15 and Indoor67, and the SIFT descriptor for the 8-sports dataset.

Method	LSP15	8-sports	Indoor67
GG (KL-div.)	**86.1**±0.5	**84.4**±1.4	**45.5**±1.1
GG (ct-linear) + BoVW1000	85.3±0.5	83.4±0.7	44.9±1.3
Previous	85.2 [221]	84.2 [205]	39.6 [134]
	85.2 [135]	73.4 [111]	25.0 [156]
	84.1 [205]		
	83.7 [25]		
	83.4 [134]		

the KL divergence based kernel achieves high performance, it is not suitable for practical systems because of its low scalability. Therefore, here we combine the ct-linear kernel of our method and the histogram intersection kernel of the BoVW method as Equation 6.24. Table 6.5 shows that we can further improve the performance by concatenating different statistics of local features provided by the Gaussian and BoVW. Figures 6.2 and 6.3 show the effect of the weighting parameter κ. Since the classification accuracy seems to shift in a stable manner, it is expected that we can optimize κ in a multiple kernel learning framework. Moreover, this approach is expected to be feasible in a perfectly linear framework by further incorporating the linear approximation techniques of the histogram intersection kernel [123; 188].

Comparison with previous work

We compare the performance of our approach for LSP15, 8-sports, and Indoor67 with that of previous work. Recent state-of-the-art work achieves remarkably high performance by concatenating various image features [24; 207; 210]. However, this is beyond the scope of this research. Therefore, we summarize the results of previous studies using single feature description. Table 6.6 summarizes the best performance of our method and that of previous work. For LSP15, hierarchical Gaussianization [221], which is a GMM-based method, and the directional local pairwise bases (DLPB) method achieved the previous best score of 85.2%. Our best score using the KL divergence based kernel is 86.1%. The performance of a more scalable ct-linear + BoVW technique is reasonably close at 85.3%. For 8-sports, the HIK-codebook [205] achieved 84.2% and we obtained a slightly better score of 84.4%. Note that [205] improved the performance by sampling local features from an original image and Sobel image at five different scales, while we only extract features from a single scale orig-

inal image[1]. For the Indoor67 dataset, the original work [156] achieved an accuracy of 25.0% by concatenating the global description with a GIST descriptor [144] and ROI detection using BoVW. Also, the local pairwise codebook (LPC) method [134] achieved 39.6%. LPC is the previous version of DLPB and utilizes coupled local features in a BoVW approach. We use the SURF descriptor for the Indoor67 dataset, motivated by its promising performance with the LSP15 scene dataset. Our best scores are 45.5% (KL div.) and 44.9% (ct-liner+BoVW), which are both superior to those of the LPC method.

Thus, our approach obtained promising results for all three benchmarks.

6.4.5 Discussion

The objective in this chapter is to develop a coding method for image features compatible with linear methods such as CCD. As we have shown, GLC is effective when used with the ct-linear kernel. This means that we can apply linear learning methods directly to $\boldsymbol{\zeta}$-coordinates in Equation 6.16. Here, remember that $\boldsymbol{\zeta}$-coordinates are obtained by applying an affine transformation to GLC ($\boldsymbol{\eta}$-coordinates). This fact indicates that we can use GLC directly as the input feature vector for learning methods invariant to affine transformations of the feature space, such as CCA and LDA. Since CCD is a CCA based method, GLC is an ideal representation for CCD. Moreover, since this idea is equivalent to regarding the $\boldsymbol{\eta}$-coordinate system as a Euclidean space, we can possibly reduce computational costs using subspaces. We investigate this issue in the next section.

6.5 Scalable Approach Using GLC and Linear Methods

6.5.1 Compressing GLC

Here, we review the implementation of GLC (Equation 6.8) in detail. In addition, we consider some efficient variations.

Let there be N training images. Suppose there are $p^{(j)}$ D-dimensional local features $\boldsymbol{v}_k^{(j)} (k \leq p^{(j)})$ in an image $I^{(j)} (j \leq N)$. Further, let $\boldsymbol{\mu}^{(j)} = \frac{1}{p^{(j)}} \sum_k^{p^{(j)}} \boldsymbol{v}_k^{(j)}$ denote their mean. This can also be interpreted as the zeroth-order auto-correlation. Also, let $R^{(j)} = \frac{1}{p^{(j)}} \sum_k^{p^{(j)}} \boldsymbol{v}_k^{(j)} \boldsymbol{v}_k^{(j)T}$ denote the auto-correlation matrix of $\boldsymbol{v}_k^{(j)}$. GLC in Equation 6.8 is the concatenation of the zeroth- and first-order auto-correlations. Namely,

$$\boldsymbol{\eta}_{0th+1st}^{(j)} = \begin{pmatrix} \boldsymbol{\mu}^{(j)} \\ upper(R^{(j)}) \end{pmatrix}. \tag{6.25}$$

[1]Without the Sobel images, [205] achieved 81.9%.

This is the most basic coding of GLC. Here, $upper()$ is a function that enumerates the components in the upper triangular part of a symmetric matrix. For example, $upper(R^{(j)})$ becomes a $D(D+1)/2$ dimensional vector.

Next, we introduce some simple coding methods using subspaces. The simplest is to use certain parts of the original GLC features. For instance, if we use only the zeroth-order correlations, we obtain

$$\boldsymbol{\eta}_{0th}^{(j)} = \boldsymbol{\mu}^{(j)}. \tag{6.26}$$

This is just the mean vector. In fact, many typical global features, including edge and color histograms, are included in this framework. Similarly, if we only use first-order correlations, we obtain

$$\boldsymbol{\eta}_{1st}^{(j)} = upper(R^{(j)}). \tag{6.27}$$

A drawback of GLC is that its dimensionality tends to be large. For example, the dimensionality of the standard GLC is $D + D(D+1)/2$. If D is large, it is difficult to train a classifier. To address this problem we perform dimensionality reduction on local features using PCA. Let R denote the auto-correlation matrix of local features extracted from all training images, then

$$R = \frac{1}{\sum_j^N p^{(j)}} \sum_j^N p^{(j)} R^{(j)}. \tag{6.28}$$

We can obtain the projection matrix U by solving the following eigenvalue problem:

$$RU = U\Omega \quad (U^T U = I). \tag{6.29}$$

Here, Ω is a diagonal matrix with eigenvalues as its elements. We cut off the principal component space at an experimentally determined optimal dimension m, and use the first m eigenvectors as the projection matrix U_m. The resultant feature vector using first-order correlations of principal components can be obtained as follows:

$$\tilde{\boldsymbol{\eta}}_{1st}^{(j)} = upper({U_m}^T R^{(j)} U_m). \tag{6.30}$$

In addition, when the mean (zeroth-order correlations) vector is added,

$$\tilde{\boldsymbol{\eta}}_{0th+1st}^{(j)} = \begin{pmatrix} \boldsymbol{\mu}^{(j)} \\ upper({U_m}^T R^{(j)} U_m) \end{pmatrix}. \tag{6.31}$$

We see that these variations are all linear transformations of the original GLC (Equation 6.25).

Figure 6.4: Sample images from the OT8 dataset.

6.5.2 Datasets

In addition to LSP15 [106] used in the previous section, we experimented with a commonly used scene classification benchmark dataset called OT8 [144]. OT8 consists of 2,688 color images of eight classes shown in Figure 6.4. Each class has 260 to 410 sample images[1].

In addition, we used the Caltech-101 dataset [55] for evaluation. This is currently the most widely used benchmark for object recognition tasks. Caltech-101 contains 101 target objects and a background class. Each class has about 31 to 800 images. In total, we performed the classification task with 102 classes.

We randomly chose 100 training images for each class in OT8 and LSP15, and 30 in Caltech-101. We used the remaining samples as test data, and calculated the mean of the classification rate for each class. This score was averaged over many trials, in which the training and test samples were replaced randomly. We used the average over 10 trials.

6.5.3 Experimental Setup

Local feature sampling

In this section, we experiment with the following four local descriptors.

1) SIFT [120]

2) RGB-SIFT [25]

[1]LSP15 is the updated version of OT8, in which seven classes were added by the authors of [57; 106].

3) Local edge histogram

4) Local HSV color histogram

RGB-SIFT is a descriptor used for color images. We extract SIFT descriptions independently from each RGB component and concatenate them to get a $128 \times 3 = 384$ dimensional vector. To provide a baseline, we also investigate the performance of our method using edge histograms and color histograms as local descriptors. For the edge histogram, we extract 72-dimensional gradient direction histograms from gray-scale images. For the color histogram, we use the standard 84-dimensional HSV color histogram from color images[1]. All features are scaled between zero and one.

Local features are extracted according to the dense sampling strategy presented in the previous section. Here, we space the keypoints M pixels apart and extract a local feature from each region of $P \times P$ pixels with the keypoint at the center. We investigate the effect of M and P through experiments.

Classification method

We mainly use a linear PDA classifier (Section 6.4.2). Since LDA, which is the core of PDA, is affine invariant, we can apply GLC features directly. In this sense, a PDA classifier is suitable for the investigation in this section. In some experiments, we also use SVM (LIBSVM) for comparison.

Theoretically, spatial pyramid matching (SPM) (Equation 6.23) can be implemented in a simple manner with PDA. That is, we can use the concatenation of GLC features from each region as a long input vector. Then we can train a PDA classifier with SPM including the weight optimization for each region. However, a major drawback of this approach is that the dimensionality of the feature vector becomes quite large. Since the computational cost of LDA increases with the cube of the number of dimensions, the training cost of this method would be immense.

Therefore, we develop an approximation method using a weighted log-likelihood instead of spatial pyramid matching. We call this SP-PDA. As is the case in Section 6.4.3, we first hierarchically partition images into grids. We use the zeroth layer (original image) to the L-th layer, and partition each l-th layer ($0 \leq l \leq L$) into $(l+1) \times (l+1)$ grids[2]. We extract image features and fit PLDA in all regions independently. Classification is conducted through the maximization of the weighted log likelihood as follows:

$$\mathcal{L} = \sum_{l=0}^{L} \alpha^l \sum_{i=1}^{(l+1)^2} \log p(\boldsymbol{u}_s^{(l,i)} | \boldsymbol{u}_{1\ldots t}^{(l,i)C}). \tag{6.32}$$

[1] We use 36 dimensions for H, 32 dimensions for S, and 16 dimensions for V.

[2] Note that this partitioning is different from the one in the previous section.

The suffix (l, i) means that the element belongs to the i-th region of the l-th layer. α^l is the weight parameter for the l-th layer, and is decided experimentally in a validation phase. This classification rule is equivalent to the minimization of the weighted distance. We classify a new sample as the class $\hat{C}$ that has the minimum value for the following distance:

$$\hat{C} = \underset{C}{\operatorname{argmin}} \sum_{l=0}^{L} \alpha^l \sum_{i=1}^{(l+1)^2} (\tilde{\boldsymbol{u}}_s^{(l,i)C})^T (\Theta^{(l,i)})^{-1} (\tilde{\boldsymbol{u}}_s^{(l,i)C}), \tag{6.33}$$

where,

$$\tilde{\boldsymbol{u}}_s^{(l,i)C} = \boldsymbol{u}_s^{(l,i)} - \frac{t\Psi^{(l,i)}}{t\Psi^{(l,i)} + I} \bar{\boldsymbol{u}}^{(l,i)C}, \tag{6.34}$$

$$\Theta^{(l,i)} = I + \frac{\Psi^{(l,i)}}{t\Psi^{(l,i)} + I}. \tag{6.35}$$

Because our method learns models independently from each region, it does not consider the co-occurrence of regions. In this sense, it is a somewhat approximate approach. However, this approach also brings a major benefit. Once learning is complete, we can tune the weight parameters freely without learning again. Therefore, the validation phase of this method is easy.

6.5.4 Experimental Results

First, we investigate the effectiveness of GLC using the OT8 scene dataset. We apply the GLC scheme to different local descriptors. We also investigate the effect of each parameter for recognition performance. Then we compare out method with state-of-the-art works. All experiments were conducted on an 8-core desktop PC (dual Xeon 3.20 GHz).

Baseline performance

Here, we extract the GLC with four different local descriptors and examine its performance. For the edge/color histogram, we fix the parameters of the sliding window as $P = 10$ and $M = 5$ (see Section 6.5.3). We extract the basic GLC as (Equation 6.25). Regarding SIFT and RGB-SIFT, we fix the parameters at $P = 16$ and $M = 5$. Because the dimensions of these descriptors are large, we perform dimensionality reduction using PCA beforehand, and then extract the first-order GLC as shown in (Equation 6.31). We use $m = 30$ PCA vectors. We also compare the classification performance of PLDA and SVM. We use experimentally decided optimal parameters for training classifiers.

Table 6.7 shows the performance of each local feature descriptor. First, we compare the classification performance of GLC using different statistical moments of local

Table 6.7: Baseline performance for OT8 (%) using GLC in different types. Classification is conducted via PDA and SVM. Regarding the results for the SVM, the plain number indicates the classification score using a linear kernel, while the italic number in parenthesis indicates that using the RBF kernel. The best score for each descriptor is shown in bold.

	0th (Mean)		1st (Cor.)		0th+1st	
	PDA	SVM	PDA	SVM	PDA	SVM
Edge Hist	66.5	70.3 (*71.0*)	**74.5**	73.6 (*72.7*)	**74.5**	73.6 (*72.8*)
Color Hist	45.2	47.4 (*50.8*)	54.1	55.3 (*55.9*)	54.2	55.3 (***56.3***)
Gray-SIFT	73.1	72.5 (*73.5*)	84.8	80.9 (*81.1*)	**85.0**	80.9 (*81.0*)
RGB-SIFT	77.7	75.2 (*76.2*)	86.4	81.4 (*81.6*)	**86.8**	81.7 (*81.9*)

features. "0th" is the case in which only the mean of the local feature is used (Equation 6.26), and is similar to the normal global feature. "1st" is the case in which only the first-order correlation of the local feature is used. We use Equation 6.27 for edge/color histograms, and Equation 6.30 for SIFT and RGB-SIFT. "0th+1st" is the case in which we incorporate both the mean and first-order correlation. We use Equation 6.25 for edge/color histograms, and Equation 6.31 for SIFT and RGB-SIFT. Regarding the results for the SVM, the plain number indicates the classification score for the SVM with a linear kernel, while the italic number in parenthesis indicates that for the RBF kernel.

As shown by these results, the performance improves considerably for each local descriptor when a first-order GLC is used. Also, using both the zeroth-order and first-order GLC improves the performance slightly, except when an edge histogram is used as the descriptor. Generally, however, we do not observe a major difference between these two cases. Theoretically, it is reasonable to exploit both mean and correlation information because they point out different statistics in a distribution of local features. However, whether it is actually effective depends on the task and the nature of the descriptors, since the mean and auto-correlations (diagonal elements of $R^{(j)}$, see Section 6.5.1) are thought to be more or less similar. Moreover, it is shown that PDA outperforms SVM for all descriptors except the color histogram. In particular, it obtains high scores when first-order correlations of SIFT/RGB-SIFT are used. This is because the affine-invariant property of PDA can absorb the effect of scale change caused by PCA.

Table 6.8: Classification performance of GLC and bag-of-visual-words (BoVW) for OT8 (%). We implement BoVW with 200, 500, 1000, and 1500 visual words.

	GLC (0th+1st)	BoVW 200	BoVW 500	BoVW 1000	BoVW 1500
PDA	**85.0**	78.9	79.9	80.7	80.8
SVM (linear)	80.9	77.2	78.1	78.6	78.6
SVM (RBF)	81.0	77.5	78.3	78.8	78.7
SVM (HIK)	N/A	80.0	82.0	82.7	83.0
SVM (χ^2)	N/A	80.8	82.5	83.2	83.7

Comparison with bag-of-visual-words

We compare the performance of GLC and BoVW using the same local features (Gray-SIFT) sampled from images. We fix the descriptor size $P = 16$ and sampling step $M = 5$ here. For an implementation of GLC, we follow Equation 6.31 using $m = 30$ PCA vectors. For an implementation of BoVW, we use the standard k-means clustering to obtain visual words. We employ PLDA and SVM for classification and consider their compatibility with features. As non-linear SVM classifiers for BoVW, we also implement a histogram intersection kernel (HIK) and χ^2 kernel [218]. Note that these kernels are not directly applicable to GLC because they are designed for histogram features.

Table 6.8 gives the results. It is clearly shown that GLC employed with PLDA achieves superior performance. These results once again show the effectiveness of the combination of GLC and PLDA. As for BoVW, although the score is relatively low with linear classifiers, we can improve the performance considerably using non-linear kernels such as the HIK and χ^2 kernel. These results correspond to those obtained in the previous chapter. In general, the performance of BoVW is comparable with that of GLC when used with a larger number of visual words and a non-linear SVM. However, this implies greater computational costs. (A more detailed discussion is given in the final section of this chapter.) Moreover, the scalability of kernelized methods is seriously poor as already discussed. Since the objective in this chapter is to develop image features compatible with linear methods, it can be said that the GLC+PDA combination ideally satisfies this requirement.

Effect of parameters

Next, we investigate the effect of various parameters using the Gray-SIFT descriptor. We use PLDA for classification.

First, we show the effect of the sampling step M in Figure 6.5. We use $P = 16$ and

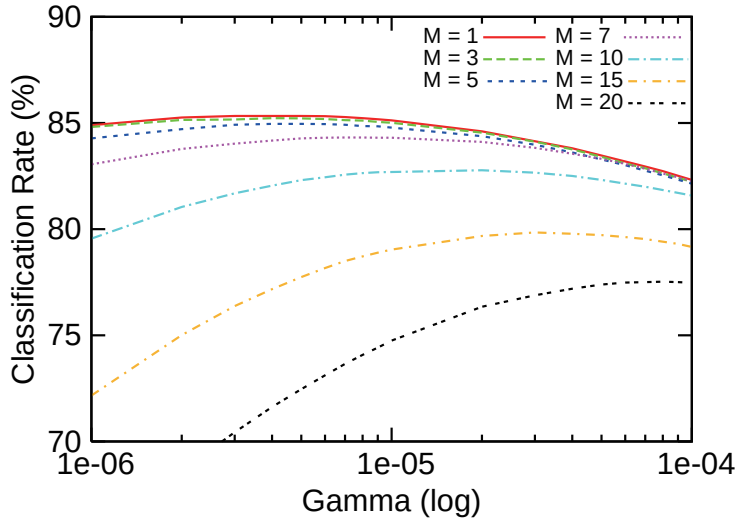


Figure 6.5: Effect of sampling density on performance ($P = 16$, $m = 30$).

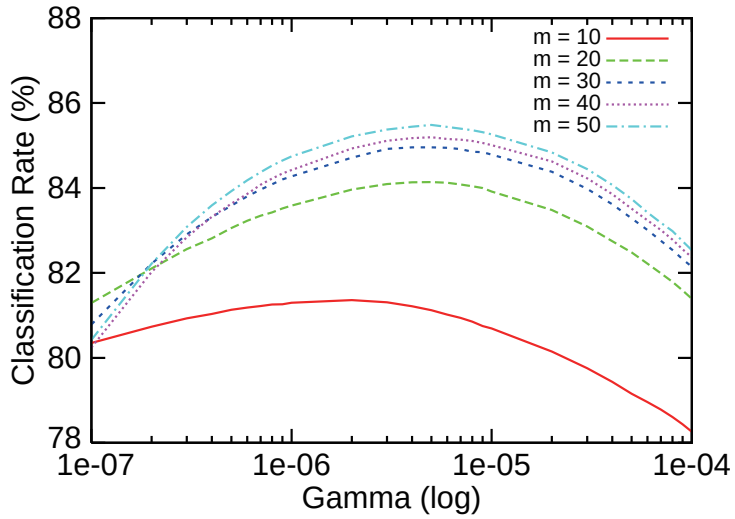


Figure 6.6: Effect of the dimensionality of PCA compression ($P = 16$, $M = 5$).

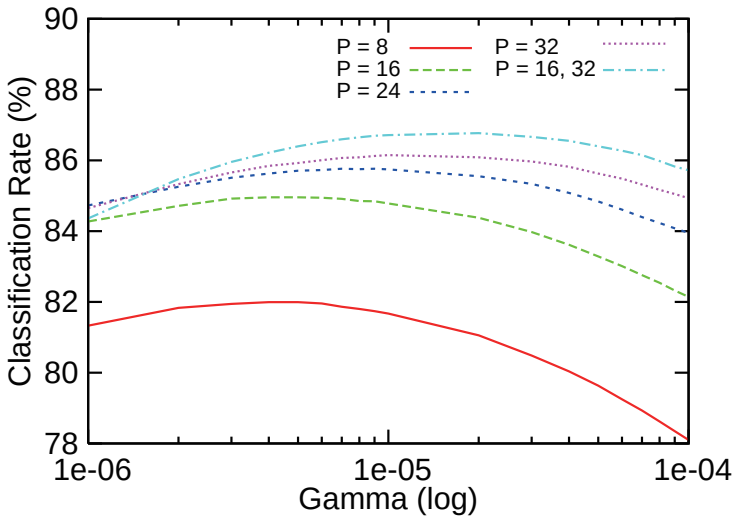


Figure 6.7: Effect of the scale parameter of the SIFT-descriptor ($m = 30$, $M = 5$).

$m = 30$ here. The horizontal axis depicts the log-scale of γ (generalization term of the PLDA classifier), while the vertical axis gives the mean performance over 100 trials. Figure 6.5 shows that the more densely we sample local features, the better the performance is. The best performance was obtained with $M = 1$, that is, a feature description in every pixel. This result corresponds to the study by Nowak *et al.* [143], who pointed out that the number of local features extracted from images is the most important factor for classification. However, the feature extraction cost increases dramatically with a smaller M. This is the trade-off between accuracy and speed.

Next, we show the effect of the dimensionality compression parameter m. We use $P = 16$ and $M = 5$ here. In general, dimensionality reduction by PCA is not related to the semantic meanings of images. Thus, there is a possibility of losing important information for discrimination if m is too small. Figure 6.6 shows the results. Not surprisingly, the larger m becomes, the higher is the performance. However, as m increases, so too does the computation time for fitting PLDA. This is another trade-off. Without feature extraction time, our system takes about 0.1 s with $m = 10$, 1 s with $m = 30$, and 10 s with $m = 50$ for fitting PLDA[1].

Further, we investigate the scale parameter of the SIFT descriptor. Figure 6.7 shows the performance using four different scale parameters. We use $m = 30$ and $M = 5$ here. Bosch *et al.* [25] showed that multiscale SIFT feature description improves the classification performance. Therefore, we also test multiscale feature description ($P = 16$

[1]As for classification of the test samples, it takes less than 0.05 s to classify all the test samples (1,888 in OT8) in all cases mentioned above.

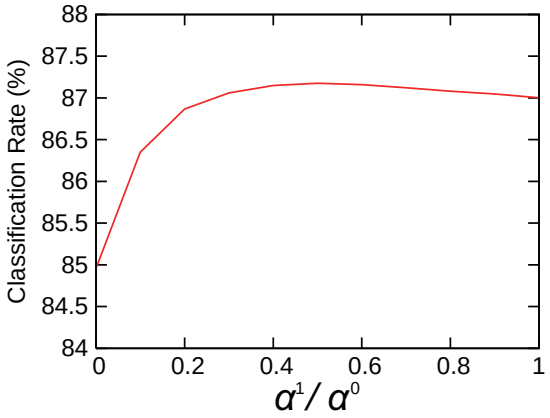


Figure 6.8: Effect of the weight parameter using at most the 2nd layer ($P = 16$, $m = 30$, $M = 5$, $\gamma = 5.0e-06$).

and 32). We extract GLC from two different scales independently, and then concatenate them to obtain the final image feature. The results show that the scale of the descriptor is an important parameter to achieve good performance. Performance can be further improved by concatenating features of two different scales.

Contribution of spatial information

We verify the performance of SP-PDA, which utilizes spatial information. Figure 6.8 shows the performance against relative weight parameters using at most the first layer (L1), while Figure 6.9 shows the performance using at most the second layer (L2). In both cases, our method improves performance. Our best results are 87.2% for L1, and 88.0% for L2. Our results show that the proposed method is not very sensitive to weight parameters, and merely by providing equal weights for each layer we can achieve good results. That is, $\alpha^1/\alpha^0 = \alpha^2/\alpha^0 = 1$[1]. In this case, we obtained 87.0% for L1 and 87.8% for L2. This stability is important for practical use.

Dimensionality reduction issues

As preprocessing for compressing local features, we need to compute the projection matrix of PCA within the training samples. Although this computational cost is substantially lower than that of building visual words by clustering, it can be a problem in really large applications. To perform dimensionality reduction without preprocessing, we investigate two ideas. The first is simply to use randomly sampled elements of the original correlations (elements of $upper(R^{(j)})$ in Equation 6.27) as the first-order GLC.

[1]This is equivalent to the naive Bayes fusion of classifiers for each region.

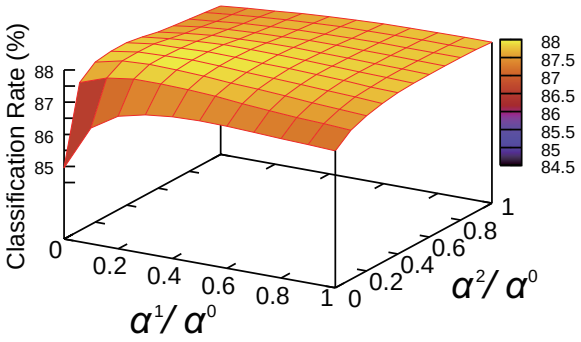


Figure 6.9: Effect of the weight parameter using at most the 3rd layer ($P = 16$, $m = 30$, $M = 5$, $\gamma = 5.0e-06$).

We call this R-GLC. The second is to share the standard projection matrix obtained from a large repository of generic images, in the same manner as PCA-SIFT [94]. To discuss the latter, we compare two GLC features taken from OT8. One uses the projection matrix obtained from the training dataset in OT8, which is the standard methodology. The other uses the matrix obtained from 3,000 random images of the Caltech-101 dataset. The objective is to consider how projection matrices obtained from Caltech-101 can be generalized to an entirely different dataset, OT8. As for R-GLC, we sample $m(m+1)/2$ elements of first-order correlations such that the dimension of the resultant feature will be the same as that of the others. We shuffle the elements for each of the 100 trials.

Figure 6.10 shows the evaluation results. We set $m = 30$, $M = 5$ here. Naturally, the performance of R-GLC is not as good as that of the other two cases using PCA projections (about 0.7% decrease) because it ignores the structure of the feature space. However, considering that R-GLC exploits a simple random approach and needs no preparation for dimensionality compression, its performance is good in comparison. This result is possibly due to the nature of the SIFT-descriptor. The SIFT-descriptor basically consists of edge histograms. Therefore, correlations of feature elements directly express the strength of certain shape patterns and are considered to be less redundant. In this sense, simply using some original feature elements is a reasonable approach, although the performance will be affected by randomness.

Furthermore, it is remarkable that the projection matrix from Caltech-101 achieves almost the same performance as the original method (about 0.05% decrease). This encouraging result suggests the possibility of sharing a common projection matrix. This issue needs to be further investigated in our future work.

In summary, a task-independent dimensionality reduction method would be the best way to precompute a PCA projection from a large image database. If such a database were not available, we could use the R-GLC strategy at the expense of clas-

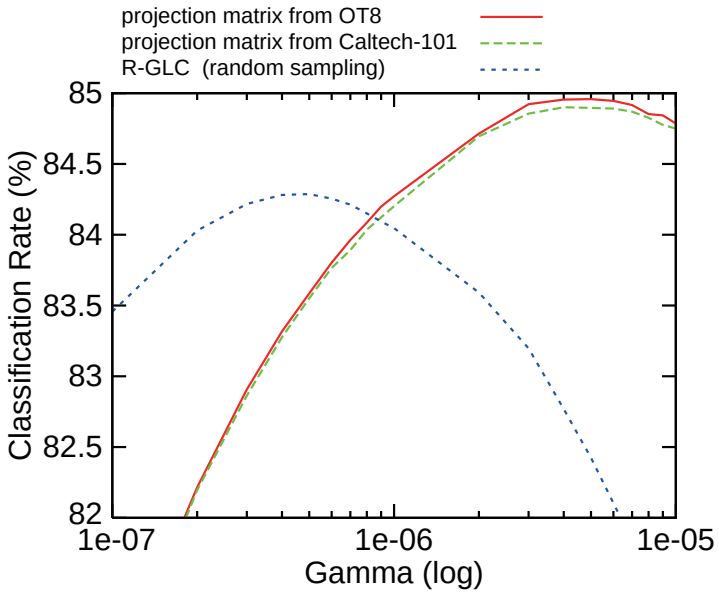


Figure 6.10: Results using different dimensionality compression methods ($P = 16$, $m = 30$, $M = 5$). We used two different projection matrices (one from OT8 and the other from Caltech-101), and random sampling.

sification performance.

Comparison with previous studies

We compare the performance of our method with previous studies using the OT8, LSP15, and Caltech-101 datasets. We basically summarize the results of previous studies using a single feature description. We use RGB-SIFT as the descriptor for OT8 and Caltech-101, and Gray-SIFT for LSP15. Moreover, we extract GLC from two different scales, $P = 16$ and 32, and set $m = 50$ for dimensionality compression. We summarize both the previous work and our current study, with spatial information (with SI) and without (no SI).

Table 6.9 gives the results of the performance comparison. We first consider the results for OT8 and LSP15. In [25; 106], the authors extract BoVW using the SIFT descriptor, and perform classification via SVM. The method in [198] estimates a part-based generative model of images using the conditional random field (CRF), and performs classification and segmentation of an image simultaneously. However, its computational cost is even higher than that of BoVW. Our method obtains relatively high performance when spatial information is not used (L0), indicating the effectiveness of GLC as a global description of images. When spatial information is included (with

SI), Perina *et al.* [148] and our Global Gaussian method [141] (KL-divergence based kernel) achieved the best performance for the OT8 and LSP15 datasets, respectively. Although the GLC+PDA method follows a simple linear framework, the difference in performance is just 2%. Instead, it enables fast training and recognition.

For the Caltech-101 dataset, the "no SI" case of [106] is the standard BoVW baseline, achieving a 41.2% classification rate. In addition, the global approach in [77] achieved 39.6% by concatenating various global features. Our L0 result substantially outperforms these standard methods.

Many state-of-the-art methods are based on modifications of BoVW or GMM, and achieve remarkably high scores [135; 195; 211; 221]. For example, comparisons with [135] show that the difference in performance is more emphasized in Caltech-101 than in LSP15. This is probably due to two main factors. The first is the manner in which spatial information is used. In Caltech-101, objects are scaled to roughly the same size and are facing the same direction. Also, they are placed in the center of images. Therefore, spatial information is thought to be especially useful in Caltech-101. Because our SP-PDA method is an approximate approach, it may lose some discriminative information. The second is the nature of the object recognition task. Compared to abstract scene images, object images often have specific local patterns. The key to object recognition is to exploit such distinctive information. BoVW and GMM, which can model local structures in the local feature space, are thought to be more suited to doing this than GLC. However, as discussed in the previous section, BoVW and GLC are not conflicting concepts. They can be combined to improve performance. We further investigate this issue using a large dataset in Chapter 7.

Computational costs

Here we estimate the computational cost of GLC in terms of both final feature extraction and preprocessing. Let p denote the number of local features in an image, D denote the dimension of the local features, and V denote the number of visual words in the BoVW scheme.

The computational cost of the final image feature extraction per image is $O(pD^2)$ for our method when PCA compression is not used (Equation 6.25). When PCA compression is exploited (Equation 6.31), this becomes $O(pm(D + m))$, where $m < D$. In contrast, the BoVW method costs $O(pVD)$ to extract a visual word histogram per image. In most studies, V is larger than D[1]. This time could be shortened using an approximate nearest-neighbor search method such as the kd-tree [12] and locality sensitive hashing (LSH) [42]. However, this creates another trade-off between accuracy and speed.

Next, we describe the computational costs of preprocessing. The basic GLC (Equa-

[1]For example, while the dimension of SIFT is $D = 128$, V is usually set to a few thousand.

Table 6.9: Comparison of the performance using two scene datasets and Caltech-101 (%).

Dataset	GLC + PDA			Previous	
	L0	L1	L2	no SI	with SI
OT8	88.8	90.5	91.1		**92.8** [148]
				82.3 [198]	90.2 [198]
				82.5 [25]	87.8 [25]
LSP15	80.0	83.2	84.1	81.5 [141]	**86.1** [141]
					85.2 [221]
					85.2 [135]
					84.1 [205]
				72.7 [25]	83.7 [25]
				74.8 [106]	81.4 [106]
Caltech-101	55.0	63.3	64.8		**77.3** [135]
					73.4 [195]
					73.2 [211]
					73.1 [221]
					67.7 [25]
					66.2 [217]
				41.2 [106]	64.6 [106]
				58.2 [68]	
				39.6 [77]	

tion 6.25) does not need preprocessing. Furthermore, we can compress GLC without any cost using a random subspace or shared projection. Similarly, it has been reported that randomly chosen visual words can achieve performance reasonably close to that obtained from clustering [143]. However, to obtain the best performance, both methods need to go through a preprocessing step. The BoVW method usually requires a preprocessing step in which the local features are clustered using the k-means algorithm [1].

The computational cost of the batch process of building visual words is $O(pNVDI)$, where I is the number of iterations. This could be greatly increased with the scale of the task, because both N and V would increase. In addition, convergence gets slower (I becomes greater) in a larger setup. Moreover, a massive amount of memory is used to store the local features of the training samples for efficient computation, with the order of memory use $O(pND)$. Our method does not require a substantial preprocessing step. The only preparation necessary is to find the PCA matrix, the complexity of which is $O(D^3 + pND^2)$, which is linear in the number of training samples. In addition, this operation requires a small amount of memory $O(D^2)$ as it needs to preserve only the covariance matrix in memory.

Finally, we report the actual computation times for the OT8 dataset using a standard computational resource[2]. Here, we use the Gray-SIFT descriptor, and set the sampling rate as $M = 10$. Also, we set the number of visual words as $V = 1500$[3]. Overall, the parameters are: $N = 800$, $D = 128$, $p = 600$, and $V = 1500$. We used a Xeon 3.20 GHz CPU with single-thread implementation. Our method takes just 90 seconds to fit PCA, while building the visual words by k-means clustering takes 18 hours. As for the final image feature description, GLC takes 60 ms per image, while BoVW takes 3.8 s.

Thus, GLC is not only informative, but also quite fast and highly scalable.

[1] In practice, a portion of local features are used for clustering.
[2] Not including local feature extraction.
[3] This is reported as the best parameter by Bosch *et al.* [25]

Chapter 7

Evaluation of Large-scale Image Annotation

In this chapter we implement a scalable and accurate image annotation system by combining our annotation method (Chapter 4) and image features (Chapter 6). Using a large dataset of twelve million images, we show the effectiveness of our method.

7.1 Dataset Construction (Flickr12M)

We used images from Flickr [1] to build a large-scale training dataset. Flickr is the largest photo sharing site, where many users upload their photos. Images are publicly available and are tagged with certain keywords (social tags) by Internet users. Currently, thousands of images are uploaded every minute. In 2010, more than 4 billion images were already stored on the site [197].

Figure 7.1 shows some examples of Flickr images and their social tags. In this section, we use the social tags as the ground truth labels. Basically, tags are expected to describe the content of images. However, some tags are totally senseless or unrelated to the content. Therefore, compared to traditional supervised datasets such as Corel5K, our dataset will be a difficult one to use with noisy and miscellaneous data.

7.1.1 Downloading Samples

As we need some keywords to retrieve images from Flickr, we used the words listed in the "All time most popular tags"[2] on Flickr as triggers[3] (Table 7.1). Note that these triggers are not used directly as the ground truth.

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

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

[3] This procedure should be replaced by a random crawling in the future, as it could bias the data.

Rooster Days
Parade
Broken Arrow
Oklahoma
Old Car
Police
Police Car
Highway Patrol
Vintage
Retro

south africa
cape town
camps bay
cameraphone
mobile
k750i
beach
tidal pool
mountain
reflection
sky
landscape
panorama
autostitch
stitched
25
twenty five
mycapetown
BestofTableMountain

California
San Diego County
El Cajon
tree
kapok
Ceiba pentandra
leaf
leaves
flower
flowers
pink
bud
buds
flowering trees
Malvaceae
Ceiba speciosa
Ceiba
rosa
flor
bello
el arbol
arbol
Silk floss tree

Figure 7.1: Examples of Flickr data: images and corresponding social tags.

Table 7.1: The most popular 145 tags on Flickr. These tags were used for the initial download.

animals architecture art august australia autumn baby band barcelona beach
berlin bird birthday black blackandwhite blue boston bw california cameraphone
camping canada canon car cat chicago china christmas church city
clouds color concert cute dance day de dog england europe
fall family festival film florida flower flowers food football france
friends fun garden geotagged germany girl girls graffiti green halloween
hawaii hiking holiday home house india ireland island italia italy
japan july june kids la lake landscape light live london
macro may me mexico mountain mountains museum music nature new
newyork newyorkcity night nikon nyc ocean paris park party people
photo photography photos portrait red river rock rome san sanfrancisco
scotland sea seattle show sky snow spain spring street summer
sun sunset taiwan texas thailand tokyo toronto tour travel tree
trees trip uk urban usa vacation vancouver washington water wedding
white winter yellow york zoo

We downloaded 18,176,861 images containing 1,486,869 unique tags for the initial collection. We filtered out minor tags occurring fewer than 2,000 times, and then removed images with no tags. Consequently, we obtained a dataset consisting of 12,283,296 images with 4,130 labels (Flickr12M). The size of each image is about 512×384.

For the test dataset, we crawled 10,000 test images from Flickr in the same way as for training dataset. It should be noted that there are many near-duplicated images on Flickr (Figure 7.2). These images are uploaded by the same user in the same situation, and annotated with the same social tags. Since it is undesirable to have these images both in the training and test datasets, we carefully separated the test set from the training dataset in terms of timestamp, so that the test set would not contain near-duplicated images of those in the training dataset.

7.1.2 Statistics of Flickr12M Dataset

Table 7.2 summarizes the statistics of Flickr12M. Each image is annotated with an average of 3.47 words. This is similar to other datasets such as Corel5K. Table 7.3 and Figure 7.3 show the word frequencies. As shown, word occurrence is highly biased and a small number of words are dominant. This is a common problem in web image mining, making it difficult to ensure diversity in annotation. Table 7.4 lists the 10 most frequently used words.

7.1. Dataset Construction (Flickr12M)

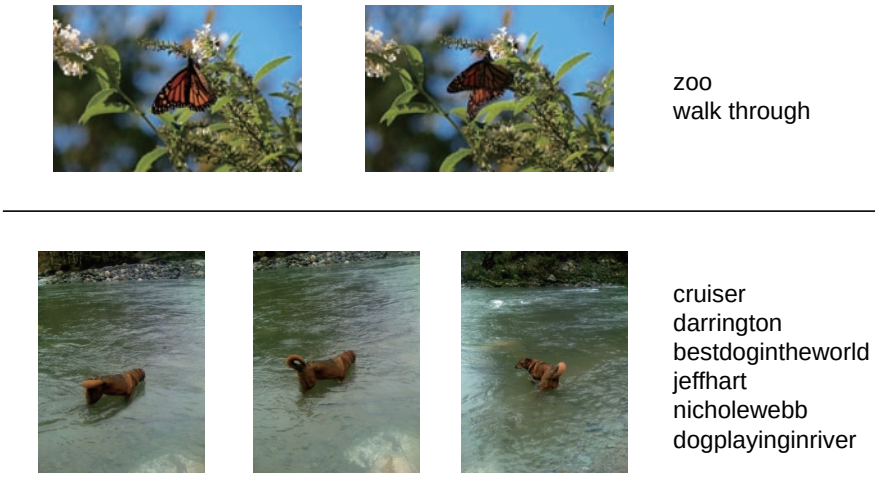


Figure 7.2: Examples of near-duplicate images in the Flickr dataset. Each row corresponds to a duplicate set. These images are annotated with the same social tags.

Table 7.2: Statistics of the Flickr12M dataset.

dictionary size	4130
# of images	12,283,296
# of words per image (avg/max)	3.47/75
# of images per word (avg/max)	10325/491595

Table 7.3: Word frequencies in Flickr12M.

Frequency			# of words
200,001	-		16
100,001	-	200,000	53
50,001	-	100,000	75
30,001	-	50,000	80
20,001	-	30,000	134
10,001	-	20,000	414
5,001	-	10,000	844
2,000	-	5,000	2514

Table 7.4: Most frequently used words in Flickr12M.

	Frequency
wedding	491595
vacation	355111
travel	350101
party	274706
japan	273445
family	263835
beach	260641
summer	251521
italy	243073
trip	239890

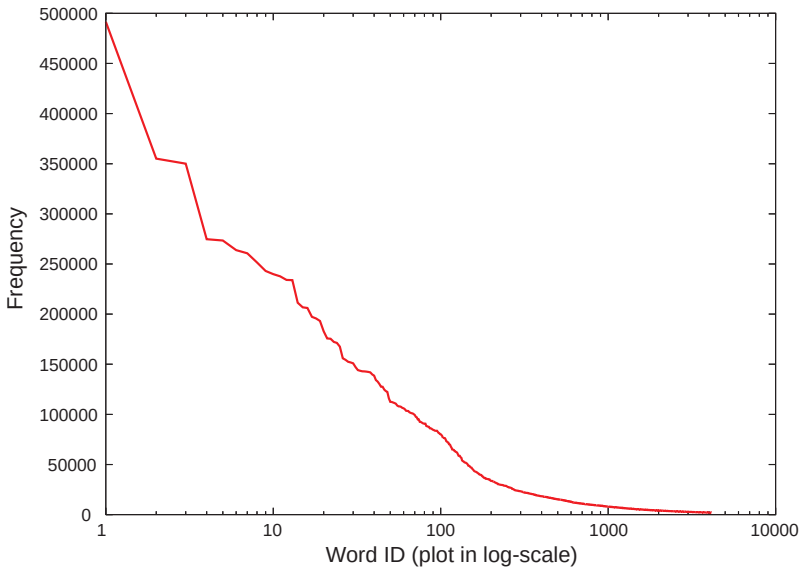


Figure 7.3: Word frequencies in the Flickr12M dataset.

7.2 Preliminary Experiments

First, using various subsets containing up to 1.6M training images we performed the same experiments carried out in Section 5.2 to confirm the effectiveness of our method for web-scale problems. Furthermore, by comparing several image features, we show that GLC based ones are highly effective.

7.2.1 Image Features

We used the following image features.

1) Tiny image [177] (3072dim)

2) RGB color histogram (4096dim)

3) GIST [144] (960dim)

4) HLAC (2956dim)

5) SURF GLC (2144dim)

6) SURF BoVW (1000dim)

7) SURF BoVW-sqrt (1000dim)

8) RGB-SURF GLC (3432dim)

The tiny image feature vector consists of pixel values of downsized (32×32) images. In the case of three-channel images, the dimensions are $32 \times 32 \times 3 = 3072$. The RGB color histogram is a feature employed in TagProp [71]. Each color component is divided into 16 bins. HLAC is the same as that used in Chapter 5. We used the SURF descriptor to implement GLC and BoVW. We extracted local features in a dense sampling approach[1]. We constructed 1000 visual words using k-means clustering. SURF BoVW is the standard implementation of a visual words histogram. Moreover, as pointed out in [151] a Bhattacharyya kernel of a BoVW is equivalent to a linear kernel of the square root of BoVW (BoVW-sqrt). This suggests that BoVW-sqrt is a more appropriate representation for linear methods. Also, RGB-SURF is the concatenation of SURF features extracted from each color component. Therefore, its dimensionality is $64 \times 3 = 192$. We extracted RGB-SURF GLC according to Equation 6.31 using 80 principal components.

[1]Here, we spaced the keypoints eight pixels apart, and extracted local features from each patch of 16×16 pixels with the keypoint at the center. We extracted $1,200 \sim 1,300$ local features per image.

7.2.2 Evaluation Protocol

The system annotated five words per image in the same manner as in Chapter 5. We then evaluated the performance with two different F-measures. The first is the F-measure of the means of word-specific recall and precision, the same metric as used in Chapter 5. For further details, refer to Appendix A. We call this the word-centric F-measure (F_W). F_W increases if the system succeeds in annotating more words. Therefore, it is appropriate to evaluate the diversity of annotation.

In addition to F_W we introduced an image-centric F-measure (F_I). For a test image I_j, we let x denote the number of words that can be correctly annotated, y denote the number of ground truth tags of I_j, and z denote the number of words that the recognition system outputs (in this experiment, z is always five).

Then the recall and the precision of I_j are defined as:

$$\text{Recall}(I_j) = x/y, \tag{7.1}$$

$$\text{Precision}(I_j) = x/z. \tag{7.2}$$

They are averaged over all the test images to obtain the image-centric mean recall (MRi) and mean precision (MPi). Finally, we use the F-measure thereof.

$$F_I = \frac{2 \times \text{MRi} \times \text{MPi}}{\text{MRi} + \text{MPi}}. \tag{7.3}$$

F_I directly reflects the accuracy of annotation for each test image. Therefore, contrary to F_W, F_I indicates the annotation accuracy of some basic words, rather than their diversity.

7.2.3 Experimental Results

We set the dimensionality of the latent space d to 20, 50, 100, 200, and 300, respectively. We then performed k-NN annotation with $k = 50, 100, 200, 400, 800, 1600$ and took the best performance. Note that we did not test MLR here, since no difference was observed in its performance from that of PLS and CCA in Section 5.2.

We compared the methods for each feature doubling the number of training images. For convenience of reference, the results are summarized in Appendix D. We see that for all methods, the more samples that are used, the better is the annotation accuracy. Compared to the scores for the 100K dataset, F_I increases by 20% and F_W increases by $100 \sim 200\%$ when using the 1.6M dataset. As shown, the improvement in F_W is substantial. This result indicates that dataset size is important to realize diversity in annotation.

As for HLAC features, CCD2 always obtains the best score. This result corresponds to that shown in Section 5.2. Moreover, we observe similar results with SURF

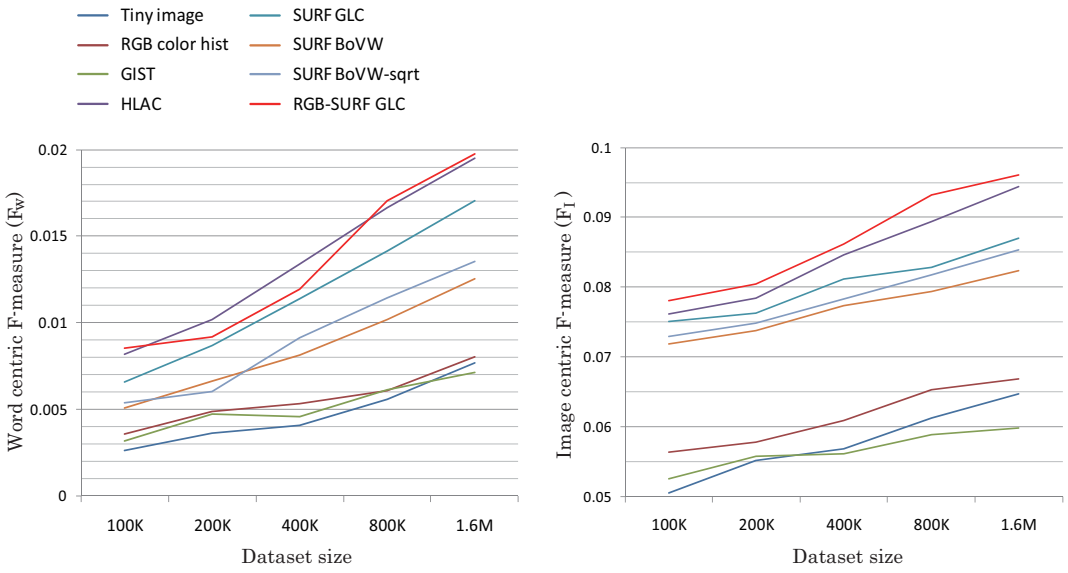


Figure 7.4: Annotation performance of each feature with CCD2 (<1.6M samples).

GLC and RGB-SURF GLC. As for the other features, CCD is not always superior, as nPLS sometimes outperforms CCD. As the example of the RGB color histogram shows, although the scores of CCA and CCD are worse when a small number of training samples are used, they outperform other methods when the number of training samples is increased. This is probably because CCA becomes more stable with a larger dataset. This result indicates the superiority of our method in a large-scale problem.

Figure 7.4 summarizes the scores for each feature with CCD2. For each feature, we selected the best dimensionality d. It is notable that HLAC and GLCs substantially outperform other features. Considering that HLAC can be interpreted as a kind of GLC, we see that the combination of CCD and GLC based features works quite well. Overall, RGB-SURF GLC obtains the best results in terms of F_I score. Also, it achieves the best F_W score using the 800K and 1.6M datasets.

In addition, we observe that BoVW-sqrt always outperforms BoVW, although they are both based on the same local features and visual words. As expected, BoVW-sqrt is a better representation for linear problems.

Based on these results, HLAC, SURF GLC, and SURF BoVW-sqrt features are used with the full Flickr12M dataset in the experiments in the next section[1].

[1]We did not use RGB-SURF GLC in the subsequent experiments since it is computationally too expensive.

7.3 Large-scale Experiments

7.3.1 Quantitative Evaluation

First, we examine the performance of each individual feature. Figure 7.5 shows the annotation performance of SURF BoVW-sqrt, SURF GLC, and HLAC features. We compare PCAW (PCA for BoVW-sqrt), CCA, and CCD2. In addition, Figure 7.6 shows the results when concatenated features are used. Since the scale of each feature is different, we only test CCA and CCD2 for concatenated features. As in the preliminary experiment, the annotation accuracy improves in a logarithmic scale with the number of training samples. In all cases, we observe that CCD2 outperforms CCA.

Moreover, it is shown that annotation accuracy substantially improves when multiple features are used. The best result is obtained when all the features are used. These results support the conclusion in Section 2.2.2, which states that using as many features as possible is the key to bridging the semantic gap. Figure 7.7 shows the superimposed results of CCD2.

7.3.2 Qualitative Effect of Large-scale Data

Here, we describe the advantage of using large-scale datasets with some qualitative examples. Since our method is an example-based method, the quality of the nearest neighbors determines the annotation performance. The more semantically similar the retrieved neighbors are, the better is the annotation. We illustrate three examples in Figures 7.8, 7.9, and 7.10, respectively. For each query image, we give the top 10 annotations and 25 nearest training samples. Here, we omit ambiguous annotations related to time and place.

Figure 7.8 (a stained glass) is the most illustrative example. When using the 100K dataset, only one stained glass image is included in the 25 nearest samples, while more visually similar sports images are retrieved. Consequently, the system outputs irrelevant words such as “football”. When the dataset grows to 1.6M, the quality of neighbors seems to improve, although the annotation results are still poor. When the full dataset is used, all 25 neighbors are stained glass images. As a result, the annotation result is greatly improved.

Similar results are observed in Figures 7.9 and 7.10. The query image in Figure 7.9 (a dolphin) is confused with abstract sea images when the dataset is small, while more dolphin images are retrieved as the dataset grows. The query image in Figure 7.10 shows a roller coaster at Disneyland called “space mountain”. The system recognizes the content of this image correctly as the dataset grows. When the full dataset is used, it even knows that this is an image from Disneyland.

Thus, as the number of training samples increases, the semantic gap seems to be relaxed and more appropriate neighbors are retrieved. The annotation accuracy improves accordingly.

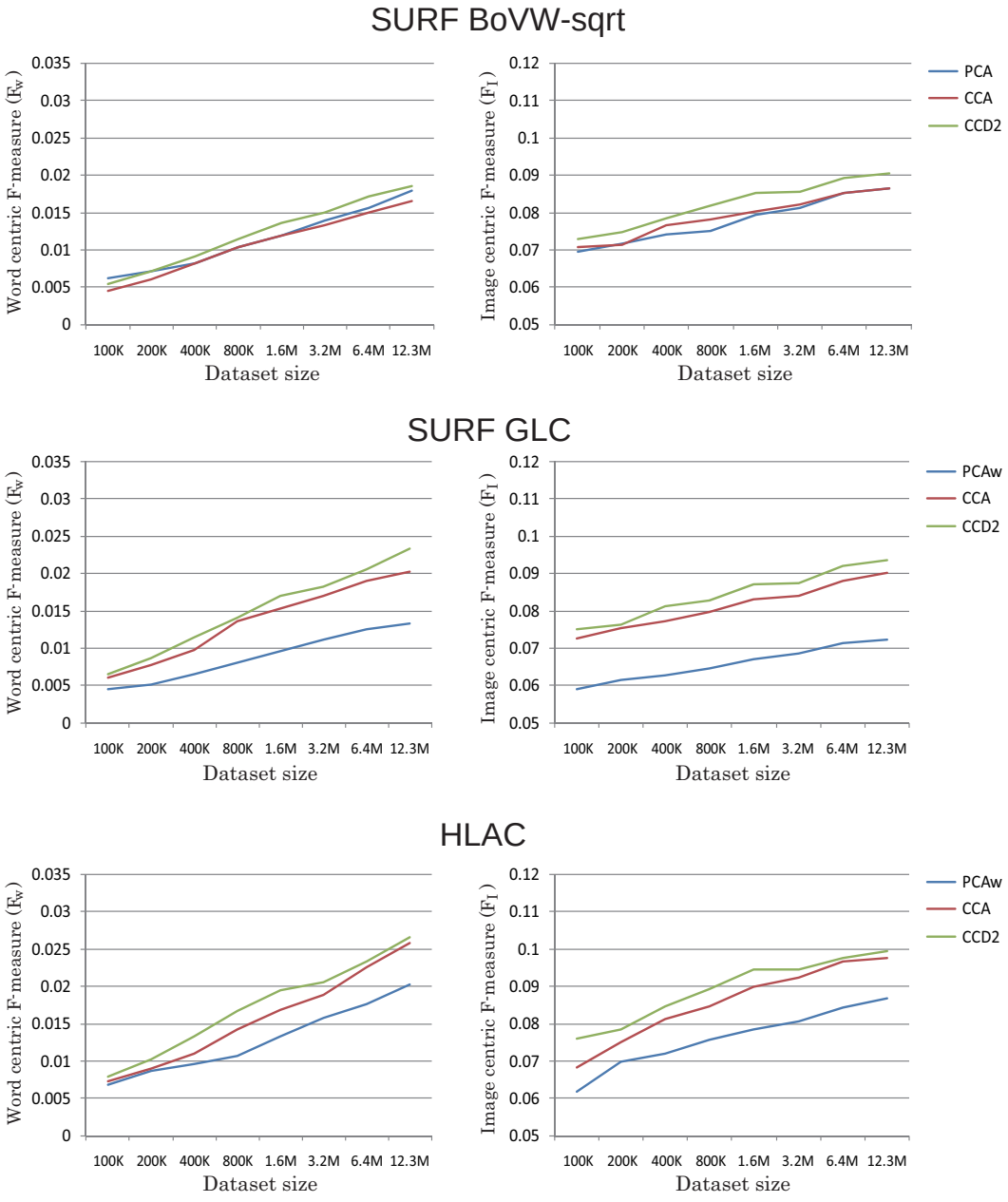


Figure 7.5: Annotation performance of SURF BoVW-sqrt, SURF GLC, and HLAC features (<12.3M samples).

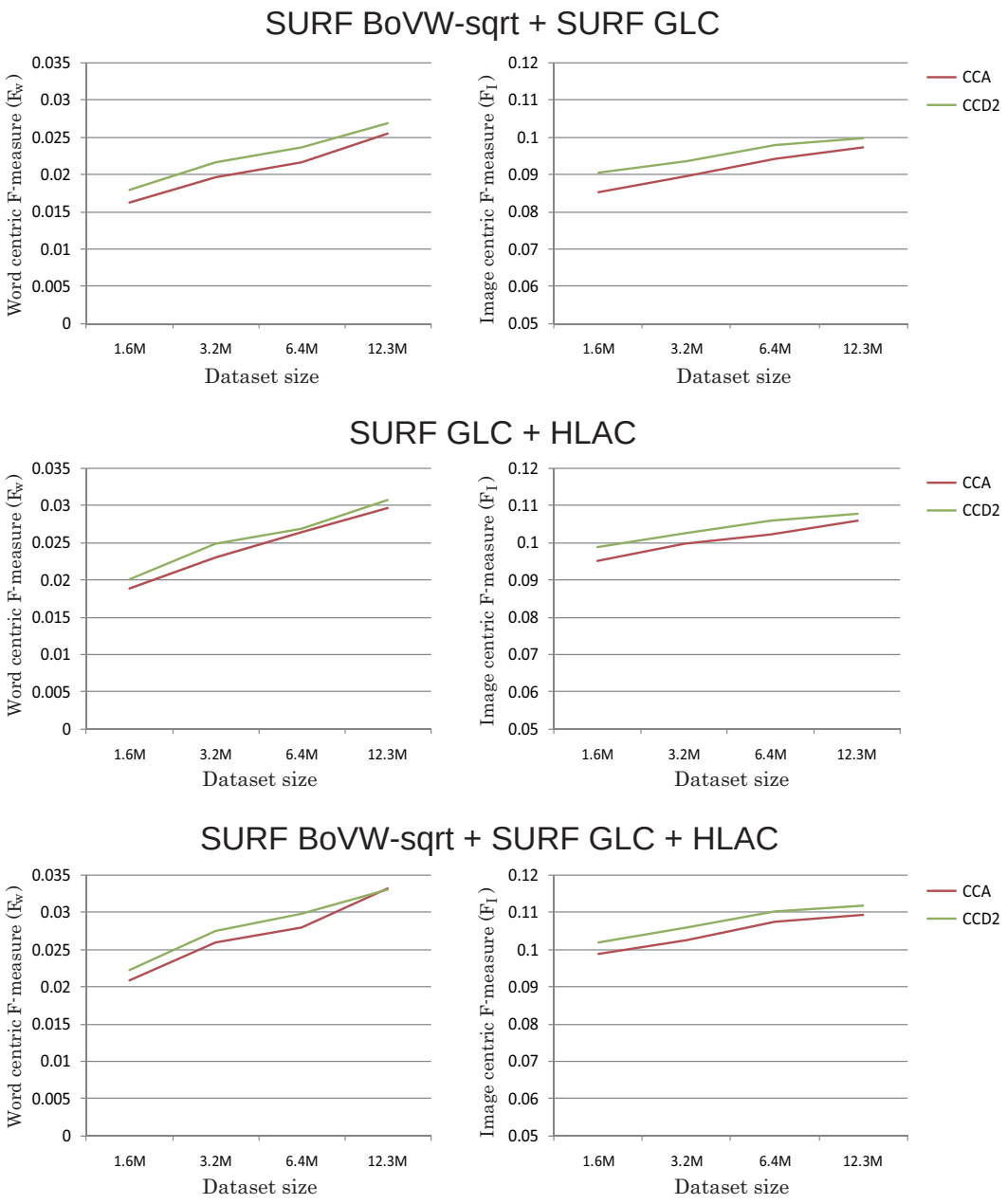


Figure 7.6: Annotation performance of combinations of SURF BoVW-sqrt, SURF GLC, and HLAC features (<12.3M samples).

7.3. Large-scale Experiments

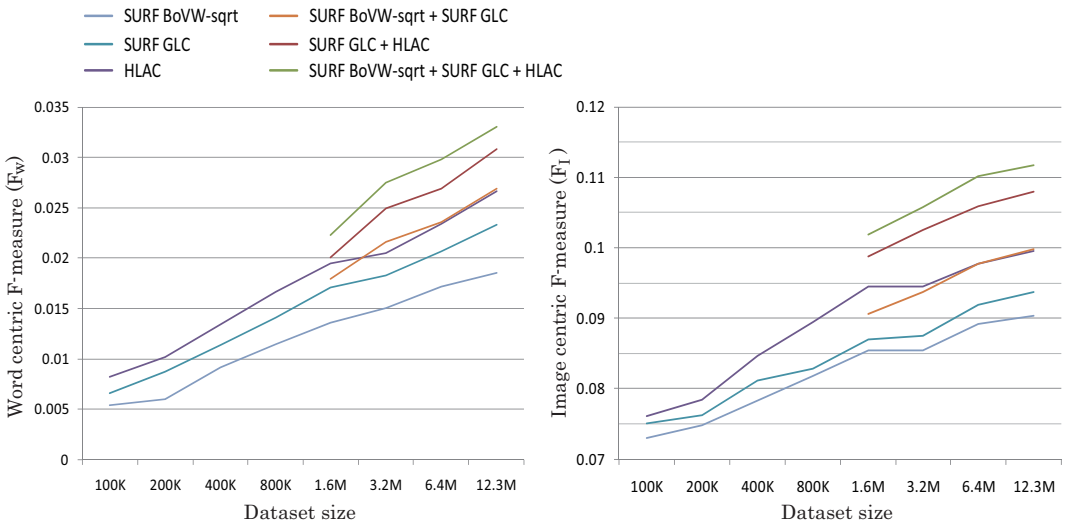


Figure 7.7: Comparison of annotation performance with CCD2 (<12.3M samples).

Query image

	Annotation results	Nearest training samples
100K	football soccer varsity girls boys travel party family school high	
1.6M	football soccer festival college futbol park people cycling marchingband vacation	
12.3M	church stainedglass football bath city vacation travel cathedral window glass	

Figure 7.8: (1/3) Example of annotation using a varying number of training samples. Correct annotations are written in red. For each case, the 25 nearest images are shown.

Query image

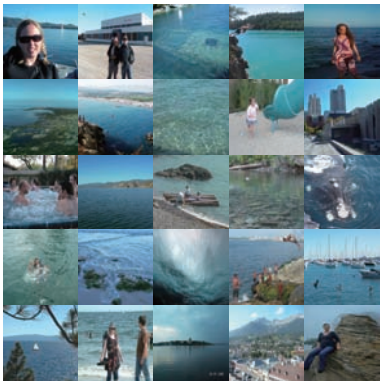
	Annotation results	Nearest training samples
100K	vacation beach travel trip nature sea island dale dalemorton mftmon	
1.6M	pool seaworld vacation swimming aquarium travel beach trip water sea	
12.3M	seaworld dolphin water sept zoo animals vacation beach orangecountry trip	

Figure 7.9: (2/3) Example of annotation using a varying number of training samples. Correct annotations are written in red. For each case, the 25 nearest images are shown.

Query image

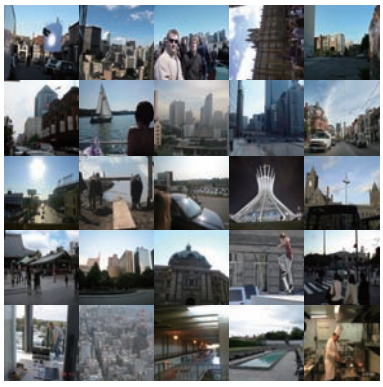
	Annotation results	Nearest training samples
100K	city trip buildings travel nikon road van birthday house car	
1.6M	travel rollercoaster freizeitpark vergngungspark achterbahn amusementparc vacation park city architecture	
12.3M	vacation festival travel trip disney city roadtrip waltdisneyworld rollercoaster matsuri	

Figure 7.10: (3/3) Example of annotation using a varying number of training samples. Correct annotations are written in red. For each case, the 25 nearest images are shown.

Chapter 8

Conclusion and Future Works

8.1 Conclusion

The key to realizing versatile and high performance generic image recognition is statistical machine learning using a large number of examples. However, since previous methods lack scalability with respect to the number of training samples, it has been practically impossible to utilize a large-scale image corpus for training and recognition. Therefore, in this thesis, we have tackled this problem and developed a scalable and accurate generic image recognition (image annotation) algorithm. This is realized by two technologies: discriminative distance metric learning between samples, and a new framework for image feature extraction. Because these technologies are mutually dependent, it is critically important that they are designed taking into account their compatibility. Finally, having applied our method to a large-scale dataset of twelve million images, we show its effectiveness. Below, we summarize some of the contributions of this thesis.

Discriminative Distance Metric Learning for Image Annotation (Chapters 4 and 5)

For image annotation, where the system outputs multiple words for an image, a non-parametric example-based approach is effective. This is probably because this approach can implicitly utilize co-occurrence information of labels in a dataset. Also, a non-parametric method can accept qualitatively new samples more stably compared to parametric models. However, the following two problems must be considered.

- How to define a distance metric between samples that relaxes the semantic gap.
- How to reduce the dimensionality of samples.

To address these problems, we need to exploit a statistical machine learning method. For large-scale problems, it is desirable for the training complexity to be linear in the number of training samples. Therefore, we focused on canonical correlation analysis (CCA), which is a technique for bimodal dimensionality compression, to learn a discriminative distance metric between samples. This approach has the following benefits.

- Training complexity is linear in the number of training samples.
- It is not necessary to access data repeatedly during training.
- Memory use for training is small and constant.
- During recognition, the cost of computing the sample distance is relatively small.

Classical CCA, however, only performs dimensionality reduction and does not give any information about the distance metric. Therefore, by exploiting the probabilistic structure of CCA, we derived a theoretically optimal distance metric, which we call the canonical contextual distance (CCD). Through experiments, non-parametric image annotation based on CCD is shown to achieve comparable performance to state-of-the-art methods with smaller computational costs for learning and recognition.

We compared CCD with related methods, PLS and MLR, which are both bimodal dimensionality reduction methods similar to CCA. With certain image features, PLS sometimes outperformed CCA and CCD in recognition accuracy. This is probably because PLS is a numerically stable method and works relatively well when non-linear image features are used. In such a case, however, the linear assumption itself is inappropriate and severely deteriorates annotation accuracy, compared to the original generative distance metric.

When image features were originally embedded in a Euclidean space, or implicitly embedded within a kernel method, CCD always achieved the best result. This indicates that CCD is generally the best method in this framework when the Euclidean assumption holds. As shown, to use CCD effectively, we must also pay attention to input image features. This is discussed in the next subsection.

Framework for Image Feature Extraction (Chapter 6)

CCD assumes that image features are embedded in a Euclidean space. In other words, the inner product in the feature space should reflect the similarity of features in terms of a generative process. However, this assumption does not hold for many practically used image features. If we apply CCD directly to them, the performance may drop substantially. In general, a kernel method is used to avoid this problem. However, to obtain good recognition accuracy, a number of samples must be used as bases for

kernelization. As a result, scalability of the method is lost, making a large-scale application impossible. This is a common problem in previous image recognition methods.

To address this problem, we need to design image features that are originally embedded in a Euclidean space. In this thesis, we proposed the global Gaussian approach, in which we model a distribution of local features in an image with a single Gaussian. Further, using the technique of information geometry, we approximately code a Gaussian into a global feature vector called the generalized local correlation (GLC).

The objective of the global Gaussian approach is to exploit low-level statistical properties of local feature distributions, which historically, have not attracted much attention. Our approach achieved the best performance with three scene recognition benchmarks. Characteristics of the global Gaussian are listed below.

- Supports an arbitrary local feature descriptor.
- Is an image-specific representation.
- Even after linear approximation, it achieves promising performance comparable to the standard bag-of-visual-words (BoVW).
- By using both global Gaussian and BoVW, we can further improve performance because they are mutually complementary.

GLC, our final feature vector, consists of the affine coordinates of the manifold of Gaussian distributions. GLC has the following properties.

- It is directly applicable to linear methods invariant to affine transformations of the input feature space, such as CCD.
- It is extracted faster that the standard BoVW.

Thus, GLC is an ideal representation for CCD based image annotation methods.

Moreover, the classical HLAC feature can be interpreted as a specific example of GLC. In other words, it is basically equivalent to GLC using pixel values as the local feature descriptor. While the compatibility of the HLAC feature and linear methods is empirically known, our analysis theoretically supports this fact.

Effect of Large-scale Image Datasets (Chapter 7)

The combination of CCD and GLC enables a scalable and accurate image annotation system, our final goal. We tested our system on a large-scale dataset of twelve million web images and obtained the following results.

- The more training samples that are used, the higher is the probability of finding semantically similar samples. As a result, both diversity and accuracy of annotation improves.

- Compared to other dimensionality reduction methods, CCD always obtains the best score.

- Performance improves when multiple image features are used. In particular, GLC based features (including HLAC) are effective.

As described, CCD is theoretically the best distance metric that can be trained with linear complexity. With small subsets, where only hundreds of thousands of samples are used, other methods such as PLS sometimes outperform CCD. On the contrary, CCD is always superior as the number of samples increases. This is because the eigenvalue decomposition of CCA, the core of CCD, becomes more stable with an increased number of samples. This result indicates that CCD shows its true power in large-scale problems. Moreover, since GLC is approximately embedded in a Euclidean space and directly applicable to CCD, it is reasonable that a combination of GLC and CCD achieves the best result. Our experimental results strongly support the effectiveness of large-scale generic image recognition using our method.

Moreover, it is shown that annotation accuracy substantially improves when multiple features are concatenated. This fact supports our discussion in Section 2.2.2 that using as many features as possible is the key to bridging the semantic gap. It is expected that by using many GLC features based on different descriptors, we can consistently improve annotation performance.

8.2 Unsolved Problems

In this thesis, we established a mathematical framework for large-scale image annotation, which has previously been an extremely difficult task. Still, we need to solve some other problems before practical annotation systems can become a reality.

Building a High-quality Training Corpus

In addition to learning methods, the quality of the training corpus also determines the performance of annotation systems. Due to the advances in crowd sourcing frameworks, we can now build large-scale datasets with a labor-intensive approach [46; 173]. However, since many anonymous people are included, it is difficult to maintain quality and consistency of image labeling. Moreover, in generic image recognition, the ground truth itself is not obvious in many cases. Many current works use external ontologies, such as WordNet [58], although its applicability to image recognition has not been thoroughly investigated. We need to consider and develop more appropriate ground truths for generic image recognition.

Incremental Learning

Irrespective of its size, a pre-defined corpus can only support general visual knowledge and does not cover the entire world. It cannot deal with objects or scenes that only exist in local environments, or newly discovered concepts. Moreover, even in known categories, recognition is difficult when the appearance of a query is very different from the training samples. In such cases, the system should incrementally learn the new visual knowledge.

To realize this, the system must be able to discover unknown categories. This is an essentially difficult challenge, since it is contradictory to usual pattern recognition that aims to generalize knowledge from experience. To balance both factors, a semi-supervised framework could be considered. Also, we need to design a framework in which the system can ask human users questions about unknown objects.

8.3 Future Works

Integration with Region Labeling Methods

We believe that global image labeling is currently the most fundamental and important topic in generic image recognition, and have thus focused on the image annotation problem. In the future, we would like to integrate region labeling (object detection) algorithms with our annotation method. First, a rough scene of an image could be sketched via image annotation. Then, we could run detectors of objects that are likely to appear in the scene. This two stage approach would lead to an efficient recognition system. Moreover, as a more advanced problem, we could improve performance by simultaneously optimizing the image annotation and detection processes.

Multimodal Extension

While generic image recognition considers a still image as input, we could integrate other resources in some applications. For example, current smart phones are usually equipped with GPS and inertial sensors, which could provide additional information for recognition. Also, in robot and car systems, we could use many other information sources such as audio and video to develop practical applications. To integrate the varied information provided by multiple modals, we could possibly use a multimodal extension of canonical correlation analysis [95].

Towards Real-world Image Recognition

The Internet is thought to provide sufficient training data for recognizing personal photos and online images. In fact, most currently used datasets consist of images down-

loaded from the Internet. However, almost all of the online images are taken by human photographers and are uploaded for some reason. In other words, online images are guaranteed to have specific clear meanings from the beginning. In contrast, the nature of real-world images that human and robots observe is totally different. They are highly arbitrary and do not always have clear meanings. The true goal of generic image recognition should be recognizing such real-world images. To realize this, in training a system we may need to exploit real-world images closely related to the objective, such as the ones taken by lifelog systems.

Appendix A: Evaluation Protocol for Image Annotation and Retrieval

In this thesis, we follow the standard evaluation protocol for image annotation and retrieval [50; 88]. We describe the details here.

A.1. Evaluation Protocol for Annotation

The recognition system annotates each test image with five words. These words are then compared with the ground truth labels. For a single word w_i, let a denote the number of images that can be correctly annotated, b denote the number of images that originally include the label w_i, and c denote the number of images that the recognition system annotates with the word w_i (correctly or not). Then recall and precision are defined as:

$$\text{Recall}(w_i) = a/b, \tag{1}$$

$$\text{Precision}(w_i) = a/c. \tag{2}$$

These values are averaged over all the test words to obtain Mean-Recall (MR) and Mean-Precision (MP), respectively. Because these metrics are a trade-off, we need to evaluate the total performance using the F-measure:

$$\text{F-measure} = \frac{2 \times \text{MR} \times \text{MP}}{\text{MR} + \text{MP}}. \tag{3}$$

Note that these scores change according to the number of output annotations by the system, although this is fixed at five in the standard protocol. For example, Figure 1 shows the scores when our annotation method is applied to the Corel5K dataset with a varying number of annotations. Naturally, the more annotations that are output, the higher is the recall and the lower is the precision obtained. However, when the number of annotations is too few compared to the number of ground truth labels, both these metrics are remarkably low, because the variety of annotations is lost. Since each image in Corel5K has 3.4 labels on average, precision drops when more than

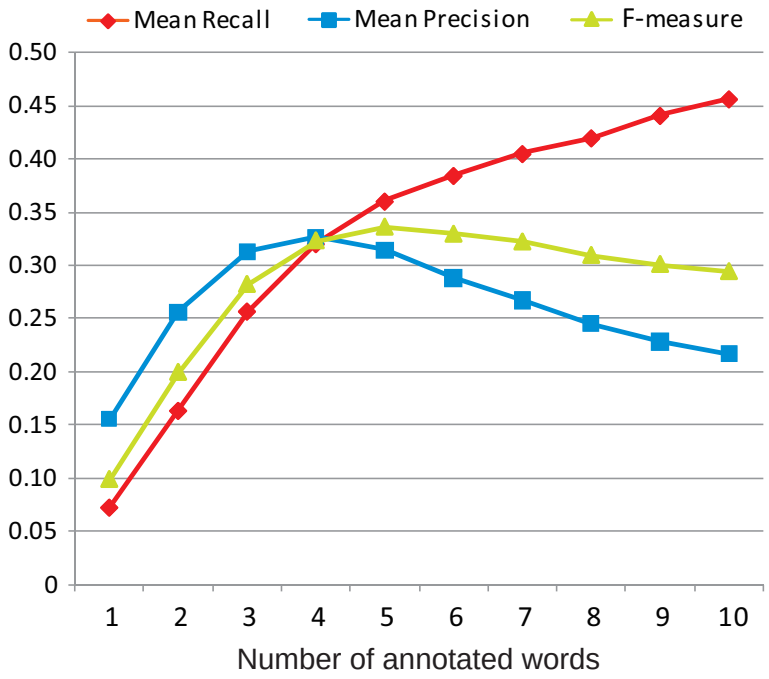


Figure 1: Annotation scores for the Corel5K dataset with varying numbers of output words. The proposed method (linear) + HLAC feature is used.

three words are output, while recall still increases. Practically, we should select an appropriate number of annotations to be output according to the task setup. Moreover, note that the theoretical upper limits for MR and MP do not reach one in many cases, because images have a different number of ground truths in general.

In addition to the above metrics, we also evaluate the number of words with positive recall (N+).

A.2. Evaluation Protocol for Retrieval

During retrieval, the system ranks the test images for each word. The system achieves better performance for retrieval if relevant images are ranked higher. We evaluate this with the Mean Average Precision (MAP).

Let N_t denote the number of candidate images for retrieval. The average precision (AP) for a query word w is defined as follows.

$$\mathrm{AP}(w) = \frac{1}{\sum_{i=1}^{N_t} y_i^w} \sum_{i=1}^{N_t} \frac{y_i^w}{i} \sum_{k=1}^{i} y_k^w, \tag{4}$$

Figure 2: Illustration of "car" retrieval results. Correct images are ranked 2nd, 5th, and 7th, respectively.

where i is the rank of each image and y_i^w is a flag that is set to one if the i-th image is related to w, otherwise zero. For example, we perform "car" image retrieval with a dataset of ten images (Figure 2), of which three images are actually related to "car". If the retrieval system ranks these images as 2nd, 5th, and 7th, we get AP(car)= $\frac{1}{3}\left(\frac{1}{2}+\frac{2}{5}+\frac{3}{7}\right) = 0.44$.

MAP is the average of AP and is a standard evaluation metric for information retrieval. We use two types of AP; one is the average over all testing words (MAP), while the other is the average over the words that provide positive recall in annotations (MAP R+).

Appendix B: Kernel Principal Component Analysis

B.1. Standard Implementation

First, we explain the standard implementation of KPCA, in which all training samples are used for kernelization. Let N denote the number of training samples. Let us consider a non-linear projection $\phi(\boldsymbol{x})$ that maps an input vector $\boldsymbol{x}$ onto a high-dimensional feature space. Usually, ϕ is implicitly given by defining the inner product with a kernel function. Specifically, using $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j) \rangle$, we can compute inner products in the original feature space without actually deriving the projection ϕ.

Let C denote the covariance matrix in the high-dimensional space, then

$$C = \frac{1}{N} \sum_{i=1}^{N} \left(\phi(\boldsymbol{x}_i) - \frac{1}{N} \sum_{j=1}^{N} \phi(\boldsymbol{x}_j) \right) \left(\phi(\boldsymbol{x}_i) - \frac{1}{N} \sum_{j=1}^{N} \phi(\boldsymbol{x}_j) \right)^T . \tag{5}$$

The solution of PCA in $\phi(\boldsymbol{x})$ space is obtained by solving the following eigenvalue problem:

$$C\boldsymbol{v} = \lambda \boldsymbol{v}. \tag{6}$$

Here, from the definition of C, Equation 6 is rewritten as follows:

$$\frac{1}{N} \sum_{i=1}^{N} \left(\phi(\boldsymbol{x}_i) - \frac{1}{N} \sum_{j=1}^{N} \phi(\boldsymbol{x}_j) \right) \left(\left(\phi(\boldsymbol{x}_i) - \frac{1}{N} \sum_{j=1}^{N} \phi(\boldsymbol{x}_j) \right)^T \boldsymbol{v} \right) = \lambda \boldsymbol{v}. \tag{7}$$

Thus, $\boldsymbol{v}$ can be represented as a linear combination of $\phi(\boldsymbol{x}_i)$.

$$\boldsymbol{v} = \sum_{i=1}^{N} \alpha_i \left(\phi(\boldsymbol{x}_i) - \frac{1}{N} \sum_{j=1}^{N} \phi(\boldsymbol{x}_j) \right) \tag{8}$$

$$= (\Phi - \Phi 1_N)\boldsymbol{\alpha}, \tag{9}$$

where,

$$\Phi = \left(\phi(\boldsymbol{x}_1) \quad \phi(\boldsymbol{x}_2) \quad \ldots \quad \phi(\boldsymbol{x}_N) \right). \tag{10}$$

$1_N \in \mathcal{R}^{N\times N}$ is a matrix whose elements are all $1/N$. Inserting this back into Equation 6, we get

$$\frac{1}{N}(\Phi - \Phi 1_N)(\Phi - \Phi 1_N)^T(\Phi - \Phi 1_N)\boldsymbol{\alpha} = \lambda(\Phi - \Phi 1_N)\boldsymbol{\alpha}. \tag{11}$$

Multiplying both sides by $(\Phi - \Phi 1_N)^T$ from the left, gives

$$\frac{1}{N}(\Phi - \Phi 1_N)^T(\Phi - \Phi 1_N)(\Phi - \Phi 1_N)^T(\Phi - \Phi 1_N)\boldsymbol{\alpha} = \lambda(\Phi - \Phi 1_N)^T(\Phi - \Phi 1_N)\boldsymbol{\alpha}. \tag{12}$$

Here, we can omit Φ using a kernel trick as follows:

$$\begin{aligned}(\Phi - \Phi 1_N)^T(\Phi - \Phi 1_N) &= \Phi^T\Phi - \Phi^T\Phi 1_N - 1_N\Phi^T\Phi + 1_N\Phi^T\Phi 1_N &(13)\\ &= K - K1_N - 1_N K + 1_N K 1_N, &(14)\end{aligned}$$

where $K_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$. Using a kernel function, we can compute a Gram matrix $\tilde{K} = K - K1_N - 1_N K + 1_N K 1_N$. Consequently, the eigenvalue problem is:

$$\tilde{K}^2\boldsymbol{\alpha} = \lambda N\tilde{K}\boldsymbol{\alpha}. \tag{15}$$

Removing $\tilde{K}$ from both sides gives

$$\tilde{K}\boldsymbol{\alpha} = \lambda N\boldsymbol{\alpha}. \tag{16}$$

Eigenvectors have a constraint such that $\boldsymbol{v}^T\boldsymbol{v} = 1$. We can rewrite this using Equations 9 and 16, to obtain

$$\begin{aligned}1 &= \boldsymbol{\alpha}^T(\Phi - \Phi 1_N)^T(\Phi - \Phi 1_N)\boldsymbol{\alpha} &(17)\\ &= \boldsymbol{\alpha}^T\tilde{K}\boldsymbol{\alpha} &(18)\\ &= \lambda N\boldsymbol{\alpha}^T\boldsymbol{\alpha}. &(19)\end{aligned}$$

The KPCA projection of input vector $\boldsymbol{x}_s$ is computed as follows:

$$\begin{aligned}\boldsymbol{v}^T\left(\phi(\boldsymbol{x}_s) - \frac{1}{N}\sum_{j=1}^{N}\phi(\boldsymbol{x}_j)\right) &= \boldsymbol{\alpha}^T(\Phi - \Phi 1_N)^T(\phi(\boldsymbol{x}_s) - \Phi 1_N^c) &(20)\\ &= \boldsymbol{\alpha}^T(K_s - 1_N K_s - K1_N^c + 1_N K 1_N). &(21)\end{aligned}$$

In the above, $1_N^c \in \mathcal{R}^N$ is a column vector whose elements are all $1/N$, and K_s is the kernel base vector of $\boldsymbol{x}_s$, that is

$$K_s = (k(\boldsymbol{x}_s, \boldsymbol{x}_1) \;\; k(\boldsymbol{x}_s, \boldsymbol{x}_2) \;\; \ldots \;\; k(\boldsymbol{x}_s, \boldsymbol{x}_N))^T. \tag{22}$$

B.2. Approximate Implementation Using a Small Number of Base Samples

The core idea of KPCA is to represent eigenvectors with a linear combination of training samples. These samples are called base samples. It is known that the more base samples that are used, the better is the performance. However, since KPCA requires computing an eigenvalue problem whose dimension is the number of base samples, it is practically impossible to use all training samples as bases.

Here, we propose an approximate, yet efficient implementation using a small number of samples for kernelization. Let n_K denote the number of base samples. As is the case in Equation 9, we represent eigenvectors with a linear combination of n_K samples.

$$\begin{aligned} \boldsymbol{v} &= \sum_{m=1}^{n^K} \beta_m \phi(\boldsymbol{x}_m) && (23) \\ &= \Phi_B \boldsymbol{\beta}. && (24) \end{aligned}$$

Although in Equation 9 we subtract the mean in the high-dimensional space, we do not do this here because this is a simple offset. Φ_B is a matrix of n_K samples, expressed as

$$\Phi_B = (\phi(\boldsymbol{x}_1) \quad \phi(\boldsymbol{x}_2) \quad ... \quad \phi(\boldsymbol{x}_{n^K}))\,. \tag{25}$$

Inserting this back into the eigenvalue problem gives

$$\frac{1}{N}(\Phi - \Phi 1_N)(\Phi - \Phi 1_N)^T \Phi_B \boldsymbol{\beta} = \lambda \Phi_B \boldsymbol{\beta}. \tag{26}$$

Multiplying both sides by Φ_B^T from the left, we get

$$\frac{1}{N}\Phi_B^T(\Phi - \Phi 1_N)(\Phi - \Phi 1_N)^T \Phi_B \boldsymbol{\beta} = \lambda \Phi_B^T \Phi_B \boldsymbol{\beta}. \tag{27}$$

Here, we introduce the following replacements through a kernel trick.

$$\begin{aligned} \Phi_B^T(\Phi - \Phi 1_N) &= \Phi_B^T \Phi - \Phi_B^T \Phi 1_N \\ &= K' - K' 1_N, && (28) \\ \Phi_B^T \Phi_B &= K_B. && (29) \end{aligned}$$

In the above, $K' \in \mathcal{R}^{n \times N}$ is a matrix of kernel base vectors of training samples, and $K_B \in \mathcal{R}^{n \times n}$ is a Gram matrix of base samples.

Consequently, the objective eigenvalue problem is

$$(K' - K' 1_N)(K' - K' 1_N)^T \boldsymbol{\beta} = \lambda N K_B \boldsymbol{\beta}. \tag{30}$$

The regularization condition for eigenvectors is as follows:

$$\boldsymbol{\beta}^T K_B \boldsymbol{\beta} = 1. \tag{31}$$

Appendix C: Details of HLAC Features

Here, we describe the color higher-order local auto-correlation (Color-HLAC) features [93], used in many of our experiments. Color-HLAC is a color extension of HLAC features [145] defined for gray images. Since this is an image-specific representation, it does not require a preprocessing step as is the case with bag-of-visual-words [40] (building visual words). Furthermore, as it can be extracted fairly quickly, it is suitable for realizing scalable systems.

The Color-HLAC features enumerate all possible mask patterns that define auto-correlations of neighboring points and include both color information and texture information. Figure 3 illustrates the mask patterns of at most the first order Color-HLAC features. In this thesis we use at most 2nd order correlations, whose dimension is 739.

We extract Color-HLAC features from two scales (original size and half size) to obtain robustness against scale change. In addition, we extract these features from edge images obtained by a Sobel filter as well as from the raw images. Let $\boldsymbol{x}_{o_1}$ denote the color-HLAC features extracted from an original-size raw image, $\boldsymbol{x}_{e_1}$ denote those from an original-size edge image, $\boldsymbol{x}_{o_{1/2}}$ denote those from a half-size raw image, and $\boldsymbol{x}_{e_{1/2}}$ denote those from a half-size edge image. We use $\boldsymbol{x} = (\boldsymbol{x}_{o_1}^T, \boldsymbol{x}_{o_{1/2}}^T, \boldsymbol{x}_{e_1}^T, \boldsymbol{x}_{e_{1/2}}^T)^T$ as the resultant image feature vector. Thus, the dimensionality thereof is $739 \times 2 \times 2 = 2956$. In this paper, we use the term "HLAC feature" to indicate this feature vector.

APPENDIX C: DETAILS OF HLAC FEATURES

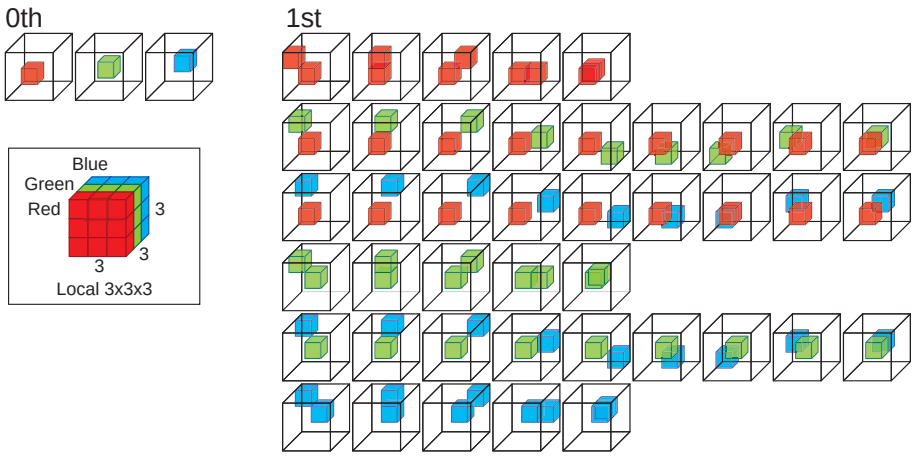


Figure 3: Mask patterns of at most the first order Color-HLAC features.

Appendix D: Experimental Results for Subsets of Flickr12M

Here, we summarize the experimental results presented in Section 7.2.3. We show the annotation scores of each image feature for various subsets of Flickr12M.

- Figures 4, 5: Tiny image
- Figures 6, 7: RGB color histogram
- Figures 8, 9: GIST
- Figures 10, 11: HLAC
- Figures 12, 13: SURF GLC
- Figures 14, 15: SURF BoVW
- Figures 16, 17: SURF BoVW-sqrt
- Figures 18, 19: RGB-SURF GLC

APPENDIX D: EXPERIMENTAL RESULTS FOR SUBSETS OF FLICKR12M

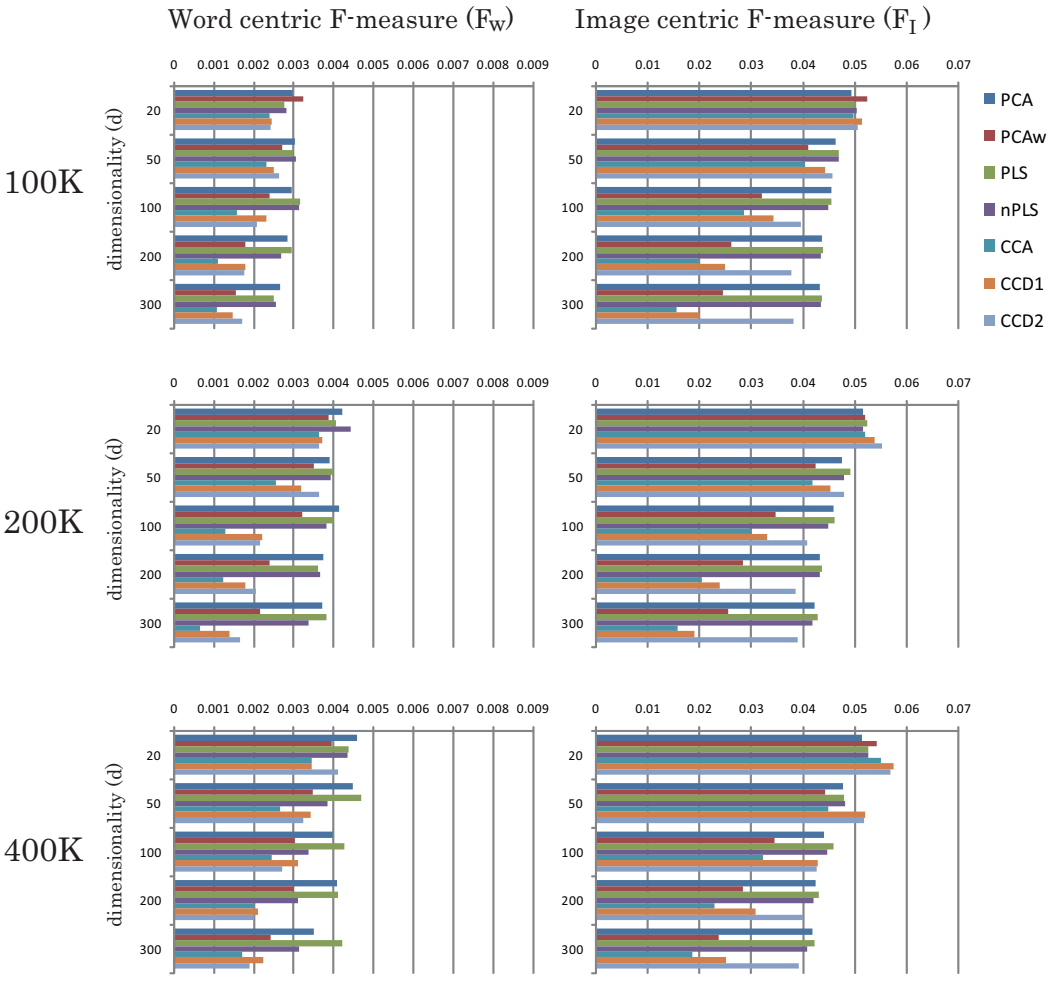


Figure 4: F-measures of **Tiny image** features for the 100K, 200K, and 400K subsets.

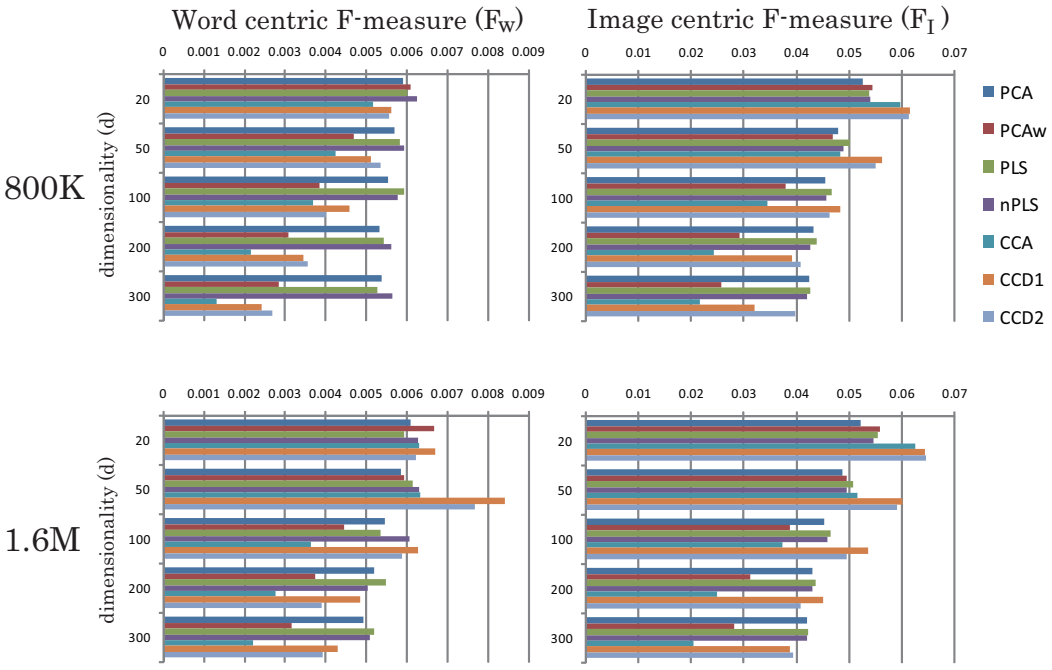


Figure 5: F-measures of **Tiny image** features for the 800K and 1.6M subsets.

APPENDIX D: EXPERIMENTAL RESULTS FOR SUBSETS OF FLICKR12M

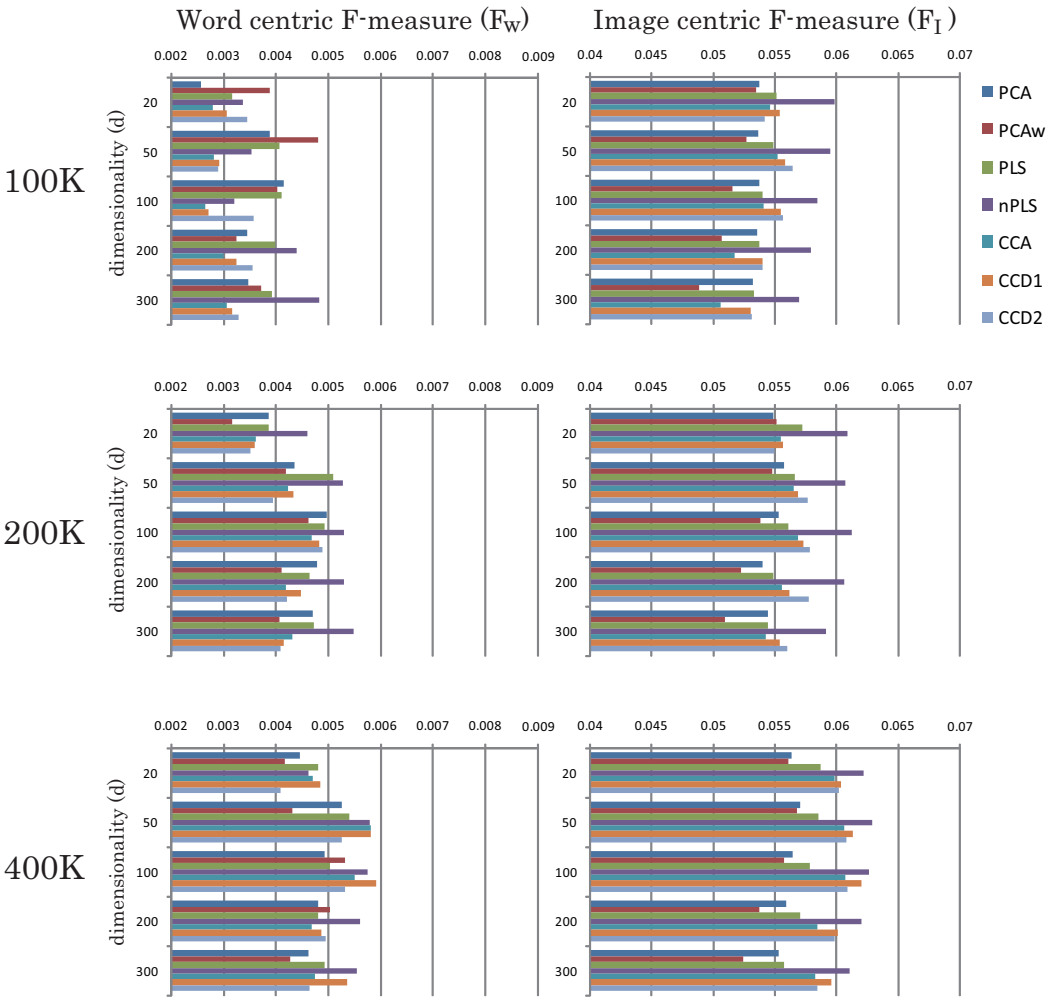


Figure 6: F-measures of the **RGB color histogram** for the 100K, 200K, and 400K subsets.

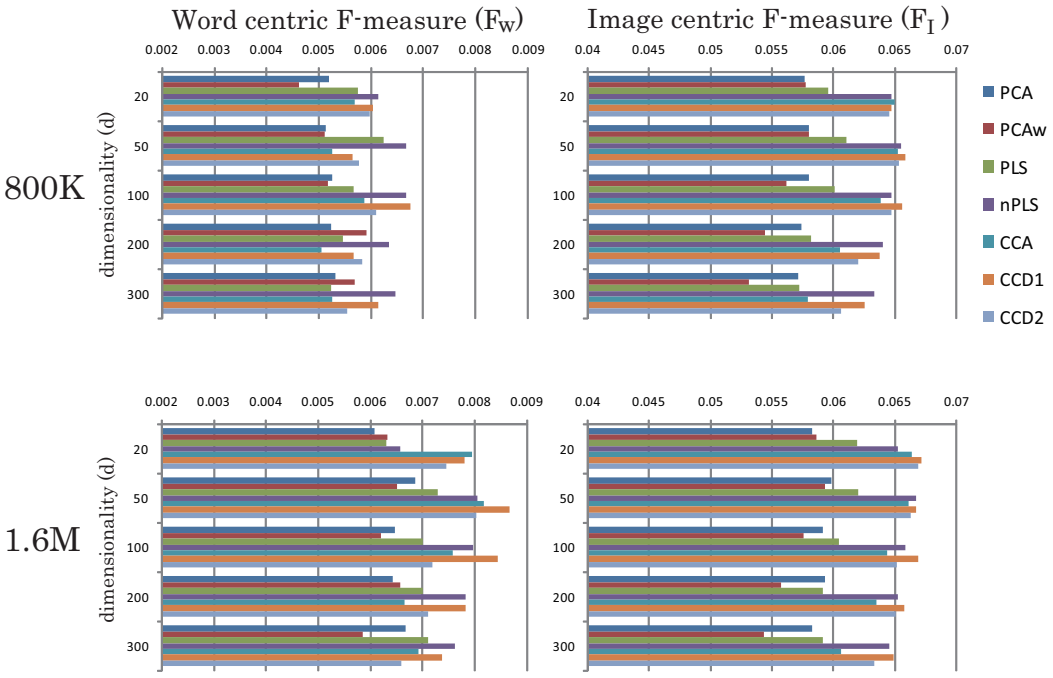


Figure 7: F-measures of the **RGB color histogram** for the 800K and 1.6M subsets.

APPENDIX D: EXPERIMENTAL RESULTS FOR SUBSETS OF FLICKR12M

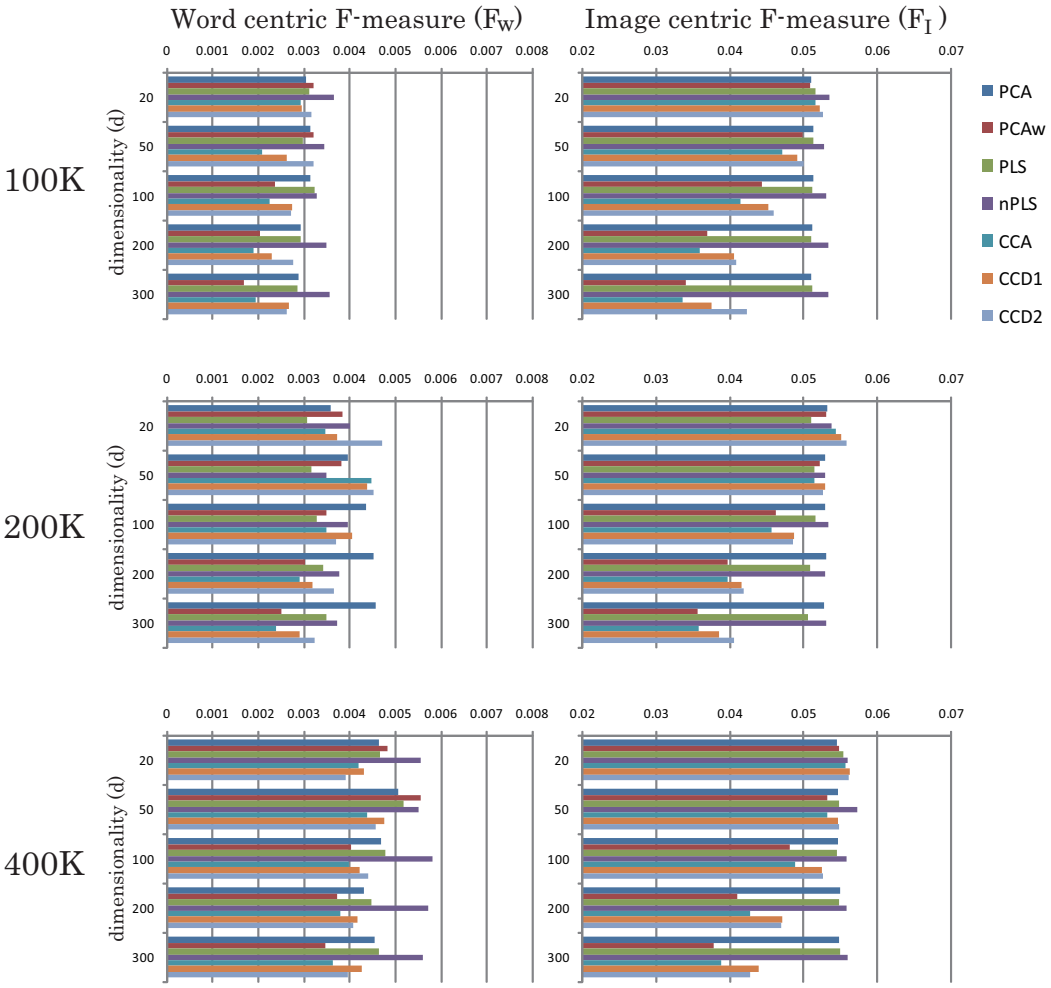


Figure 8: F-measures of **GIST** features for the 100K, 200K, and 400K subsets.

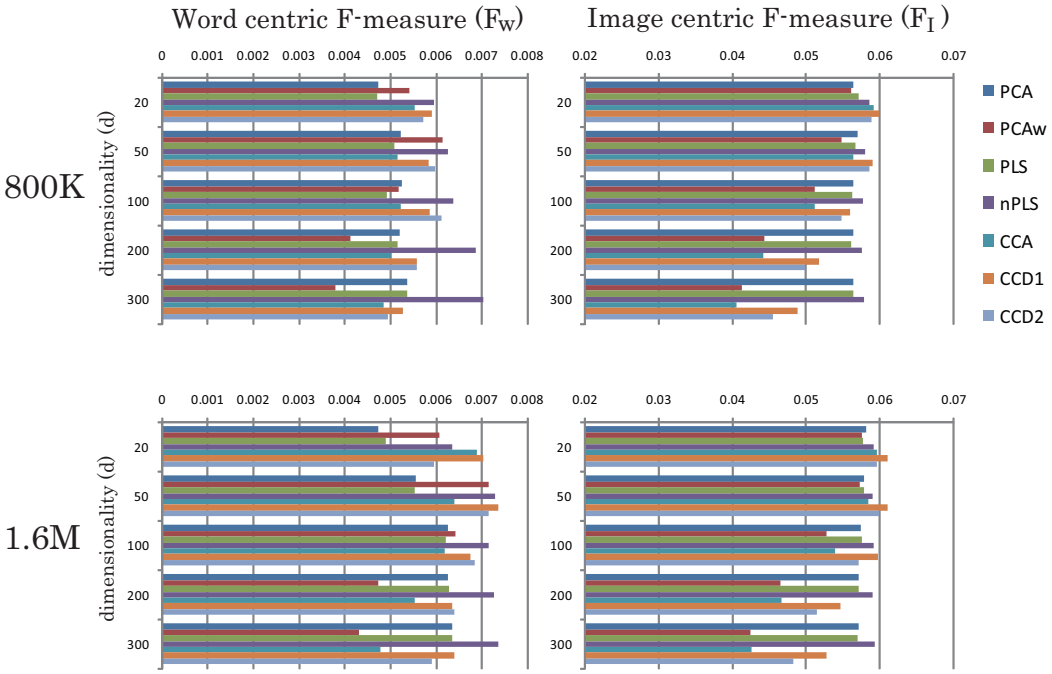


Figure 9: F-measures of **GIST** features for the 800K and 1.6M subsets.

APPENDIX D: EXPERIMENTAL RESULTS FOR SUBSETS OF FLICKR12M

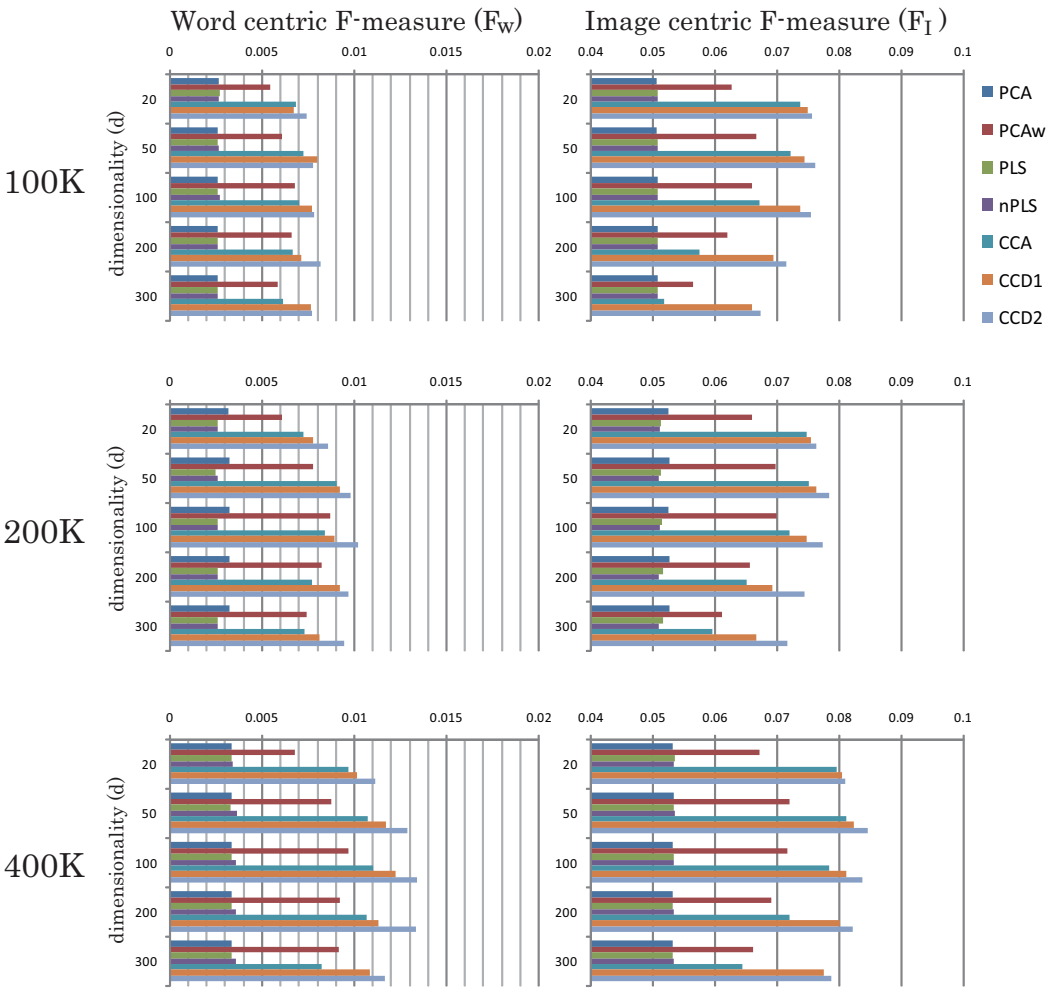


Figure 10: F-measures of **HLAC** features for the 100K, 200K, and 400K subsets.

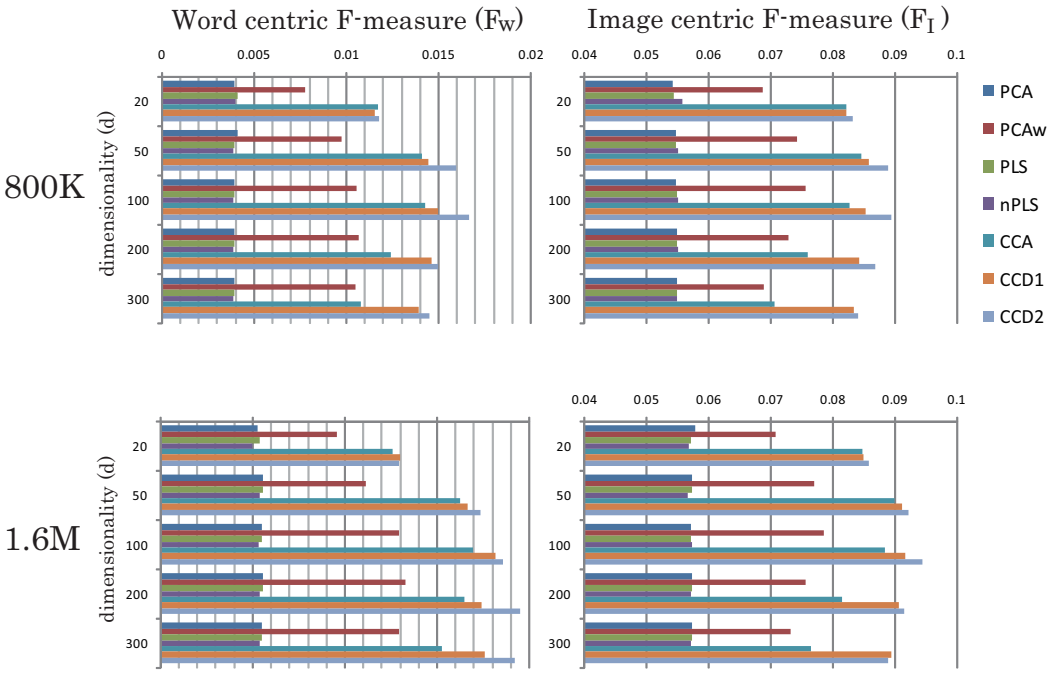


Figure 11: F-measures of **HLAC** features for the 800K and 1.6M subsets.

APPENDIX D: EXPERIMENTAL RESULTS FOR SUBSETS OF FLICKR12M

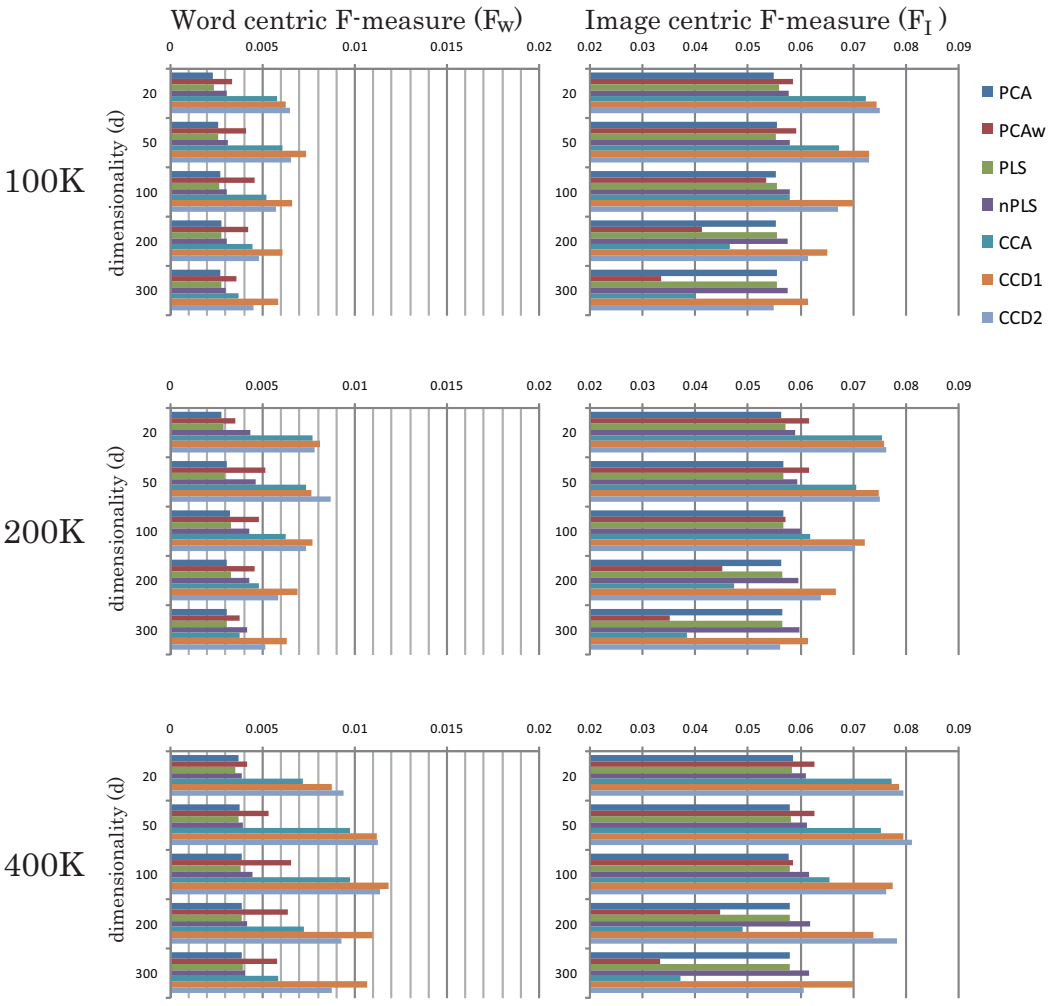


Figure 12: F-measures of **SURF GLC** features for the 100K, 200K, and 400K subsets.

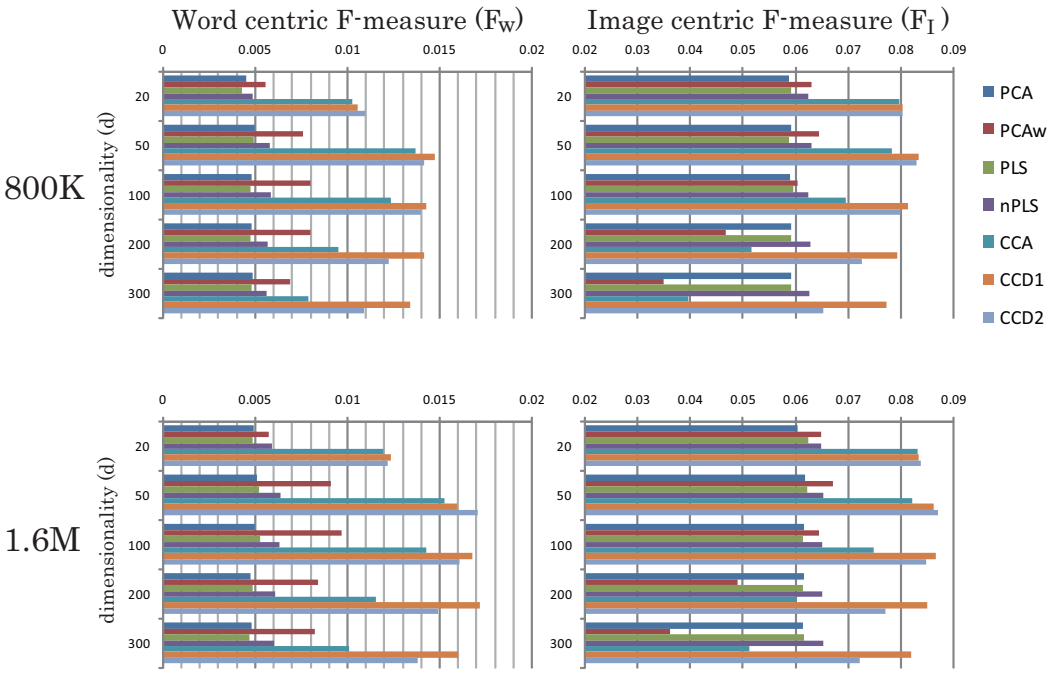


Figure 13: F-measures of **SURF GLC** features for the 800K and 1.6M subsets.

APPENDIX D: EXPERIMENTAL RESULTS FOR SUBSETS OF FLICKR12M

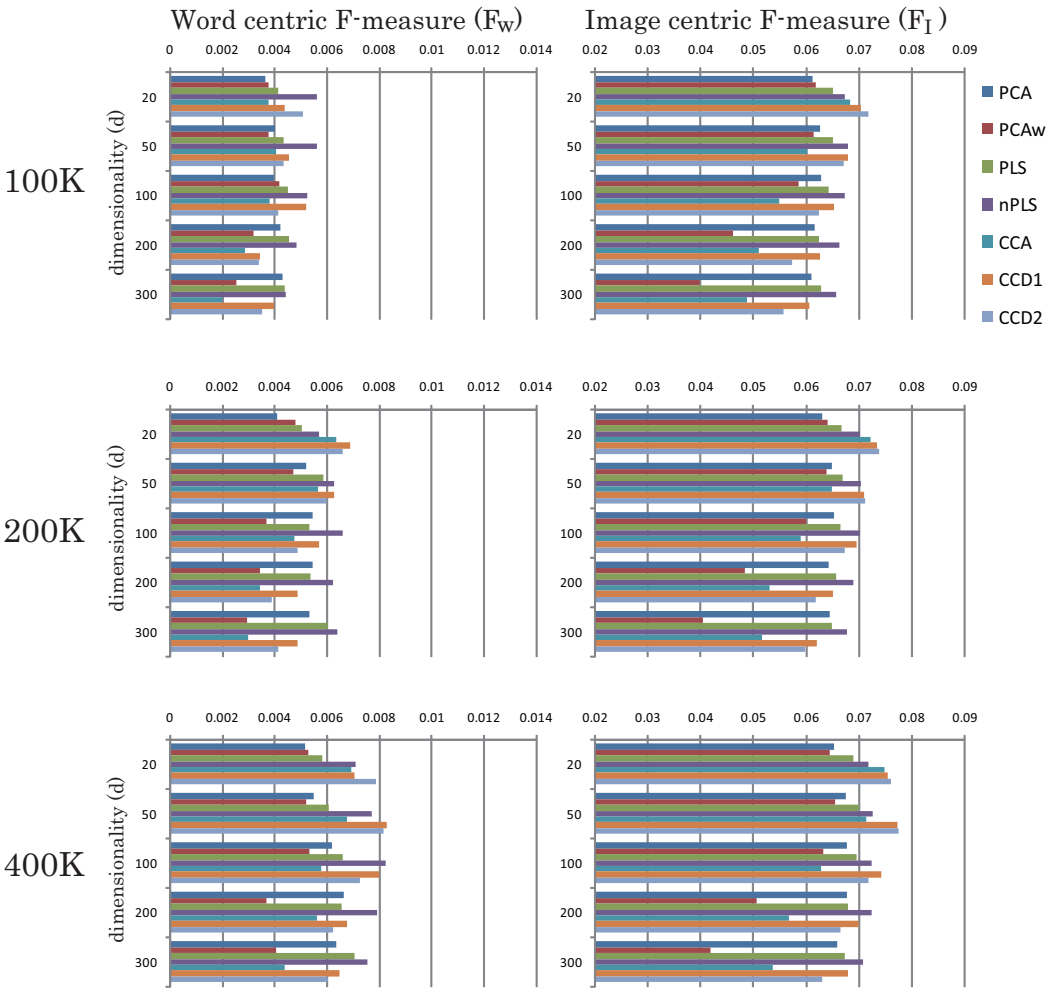


Figure 14: F-measures of **BoVW** features for the 100K, 200K, and 400K subsets.

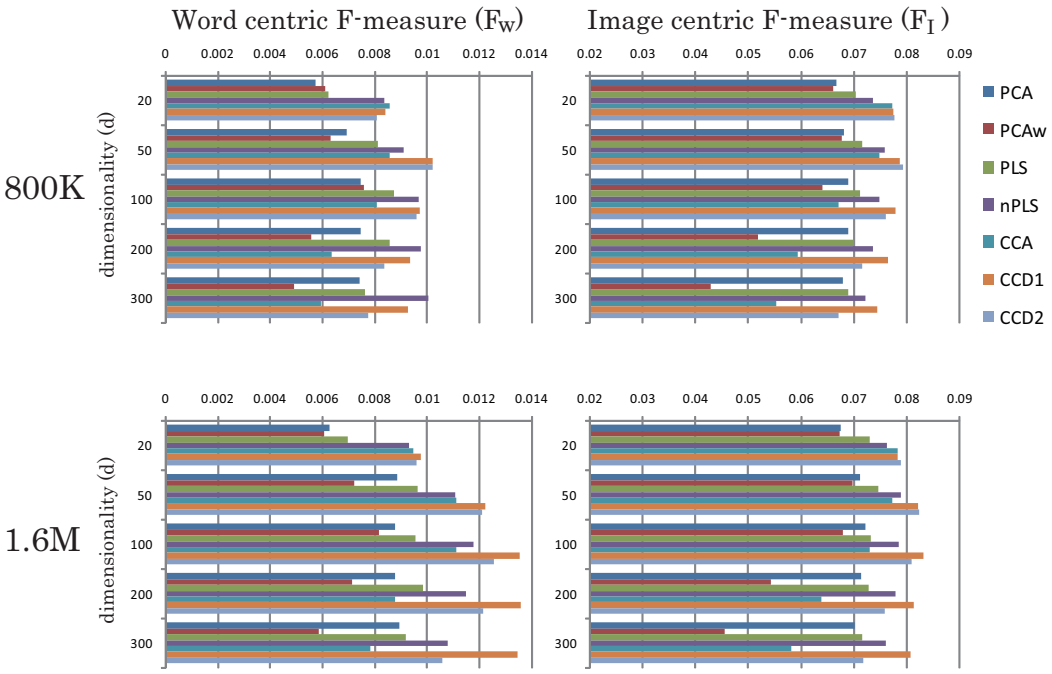


Figure 15: F-measures of **BoVW** features for the 800K and 1.6M subsets.

APPENDIX D: EXPERIMENTAL RESULTS FOR SUBSETS OF FLICKR12M

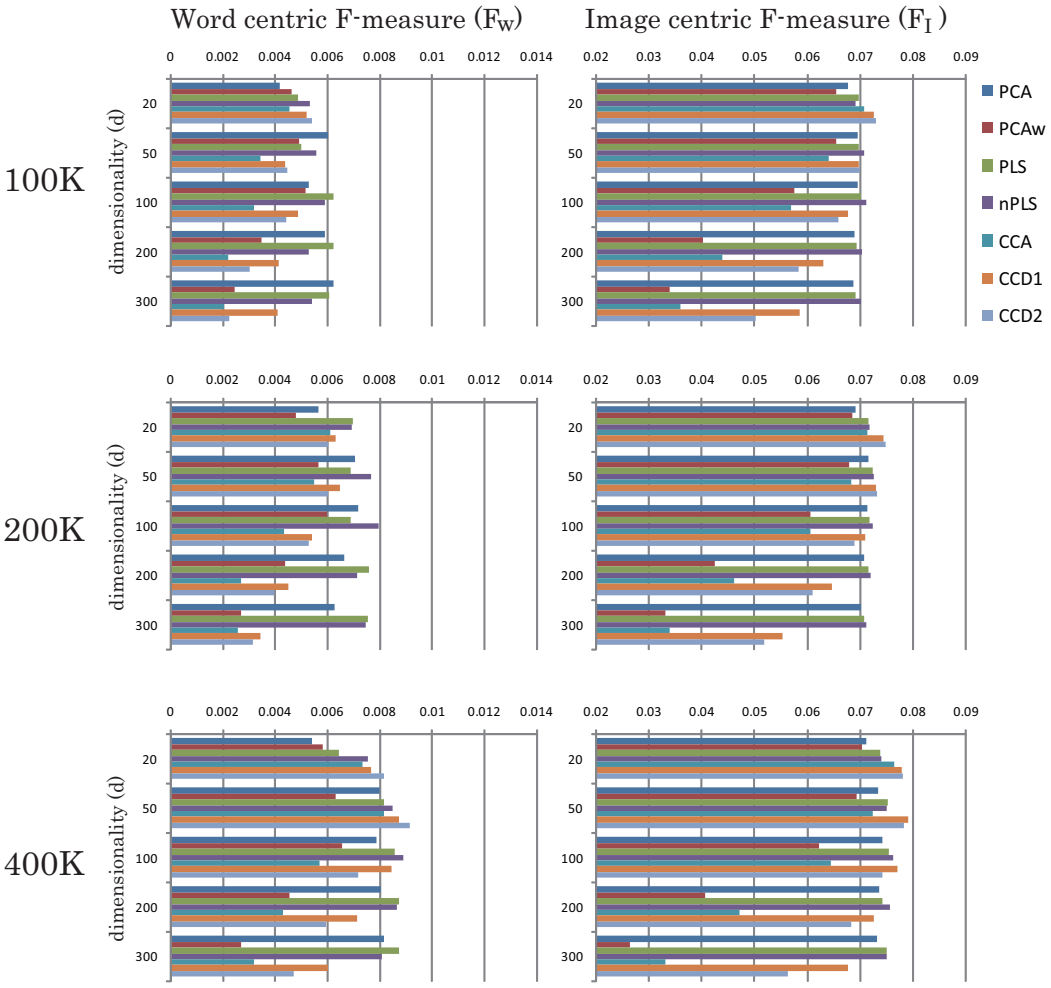


Figure 16: F-measures of **BoVW-sqrt** features for the 100K, 200K, and 400K subsets.

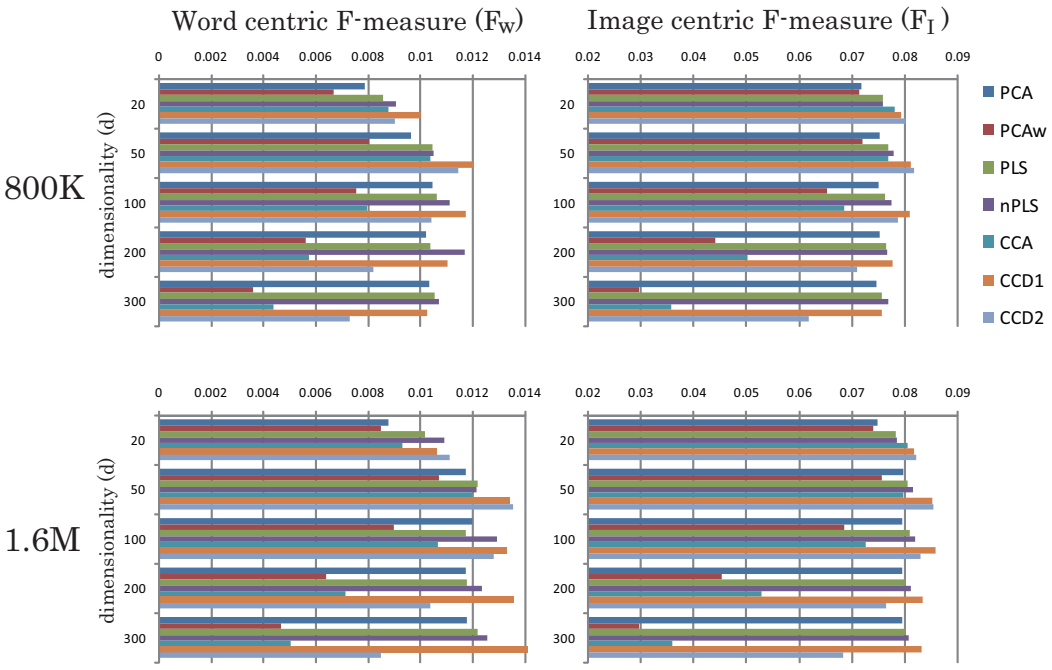


Figure 17: F-measures of **BoVW-sqrt** features for the 800K and 1.6M subsets.

APPENDIX D: EXPERIMENTAL RESULTS FOR SUBSETS OF FLICKR12M

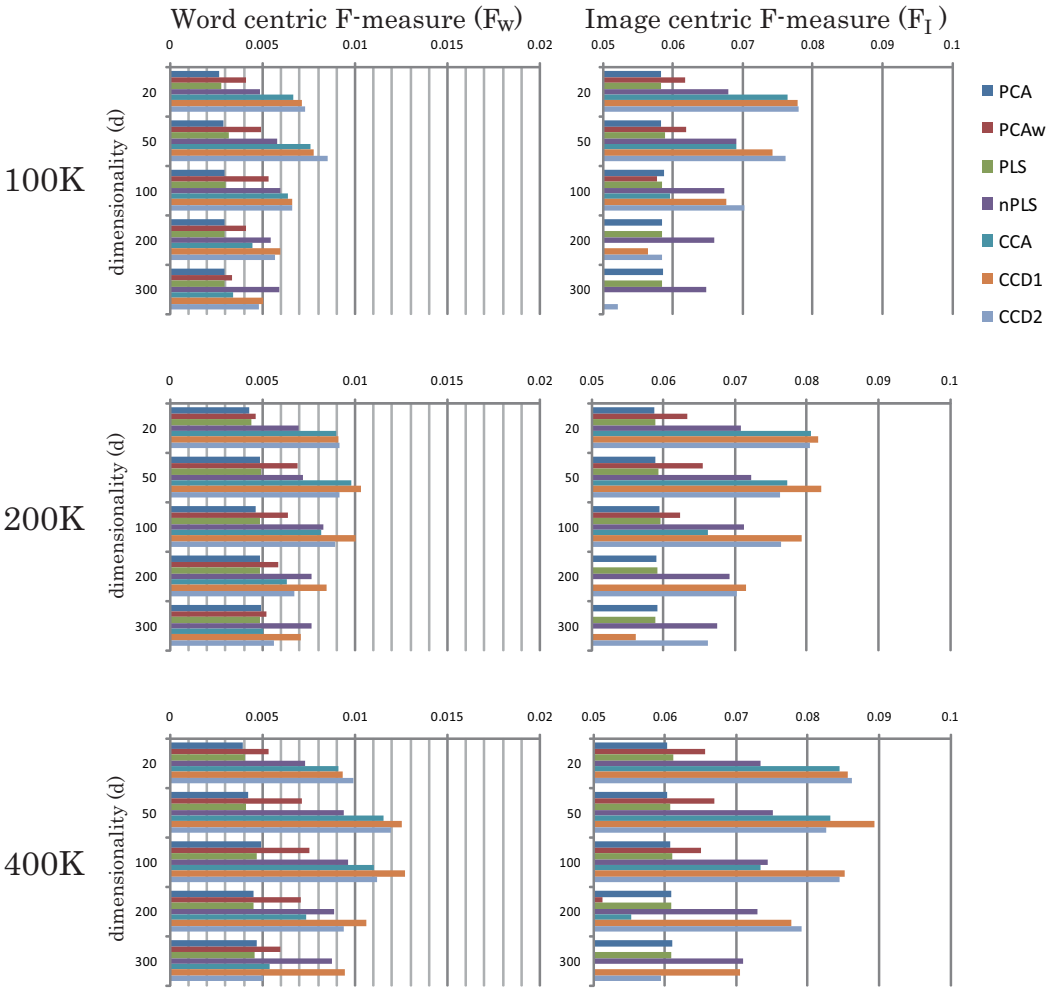


Figure 18: F-measures of **RGB-SURF GLC** features for the 100K, 200K, and 400K subsets.

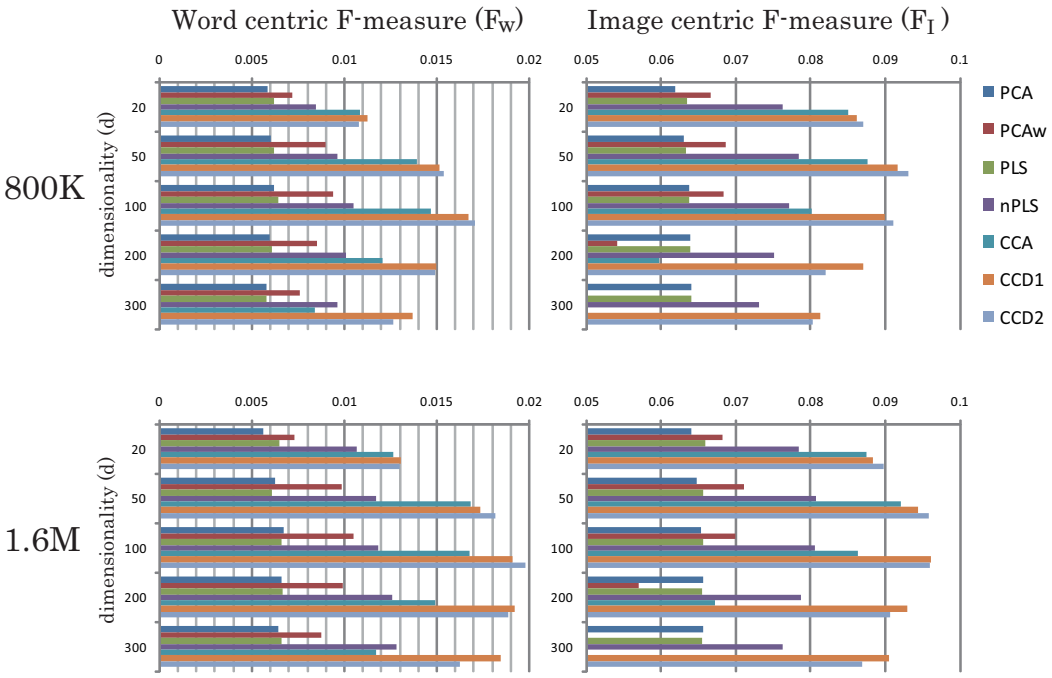


Figure 19: F-measures of **RGB-SURF GLC** features for the 800K and 1.6M subsets.

Appendix E: Hashing-based Rapid Annotation

Here, we develop an approximate, yet extremely fast and efficient annotation method by combining standard hashing methods with the CCD framework. Since each sample is represented by a small Hamming code, millions of samples can easily fit in a single computers memory. We develop the algorithm in the context of content-based image retrieval (CBIR), as this is the core of non-parametric annotation methods. Finally, we apply the algorithm to annotation problems.

E.1. Overview

CBIR has been studied for a long time, and is now flourishing in industrial applications. However, it is still challenging to instantly retrieve a desired image from millions of samples. The difficulty of CBIR stems from two main problems. First, image features are generally high-dimensional, making a naive linear search infeasible for large-scale problems, both in terms of computation time and memory use. Therefore, we need efficient indexing and search algorithms to handle massive amounts of high-dimensional data. Nevertheless, it is extremely difficult for any method to search nearest neighbors in a high-dimensional space. This is known as the "curse of dimensionality" problem.

The second problem is the so-called semantic gap; that is, low-level image features are not directly related to the high-level meanings (Section 2.2.2). In general, many search methods are designed for unsupervised problems. Specifically, they derive a compact representation of an image approximating the original Euclidean distance. While these methods are well suited to searching visually similar images such as near-duplicates, it is still difficult to estimate the semantic distance between images. To relax the semantic gap, it is reasonable to consider a supervised machine learning framework that exploits corresponding modals such as textual information. However, in general, training costs of supervised methods are expensive compared to those of unsupervised ones. It is thus, challenging to apply such methods to very large-scale problems, which is exactly our aim.

In this thesis, we have worked on a multi-modal setting to enhance semantic image retrieval. Specifically, in the CCD2 (Section 4.2.3) framework, we consider the following problem. Suppose we have image features $\boldsymbol{x} \in \mathcal{R}^p$ and corresponding text features $\boldsymbol{y} \in \mathcal{R}^q$. Suppose also, that we have a database of N samples $\mathcal{T} = \{\boldsymbol{x}_i, \boldsymbol{y}_i\}_{i=1}^N$. We wish to retrieve an image from the database that most closely resembles the query image $\boldsymbol{x}_Q$. That is,

$$\mathrm{NN}(\boldsymbol{x}_Q) = \arg\min_{\boldsymbol{x} \in \mathcal{T}} D(\boldsymbol{x}_Q, \{\boldsymbol{x}, \boldsymbol{y}\}), \tag{32}$$

where D is the distance between two instances. This is a reasonable setup considering the exponential growth of the Internet. For example, using a photo taken with a mobile phone as the query, one can retrieve semantically similar images included in web documents without typing explicit keywords.

To compute distance and store samples efficiently, we derive small binary codes in a topic space embedding both visual and textual similarities. D is computed in terms of the Hamming distance between the binary codes. Since each sample is represented by a few bytes of memory, we can conduct fast topic-level retrieval on a large-scale database, even using a linear search. In experiments, we show the effectiveness of our method on real problems, using the Flickr database containing 12 million images.

E.2. Related Work

We review previous studies of nearest neighbor search algorithms. In the beginning, many researchers tried to speed up an exact nearest neighbor search. Binary search trees (*e.g.* the kd-tree [12]) are representative examples. They realized a fast exact nearest neighbor search in a relatively low-dimensional space. However, it became apparent that these binary search methods are not effective for high-dimensional data and even cost as much as a linear search [107; 199]. Speeding up an exact nearest neighbor search in a high-dimensional space is still an unsolved problem today.

Therefore, many recent studies have focused on an alternative approach: an approximate nearest neighbor search. This framework relaxes the problem by giving up searching for the exact nearest neighbors, and attempts to retrieve approximately neighboring samples with a high probability. The objective is to satisfy the trade-off between accuracy and speed for practical tasks. This idea was first proposed for locality-sensitive hashing (LSH) by Indyk *et al.* [42; 85]. LSH constructs hash functions using random projections so that similar samples collide. The LSH research showed that we can control the trade-off between accuracy and speed with a theoretical background. While the original LSH assumes a Euclidean distance as the similarity measure for input samples, it has been modified to use arbitrary Mahalanobis distance [97] and non-linear distance metrics [96].

LSH is capable of realizing a rapid nearest neighbor search on various high-dimensional

data, and has been applied to many computer vision problems. However, the standard LSH based on hash tables only queries candidate samples and requires a final re-ranking phase using the original features to determine the nearest neighbors in the bucket. This means that we need to store all raw instances in memory to implement the rapid retrieval, which is unrealistic for internet-scale problems.

A possible solution is to use binary hash values directly to represent each sample [35]. However, since the hash functions for LSH are generated randomly, hash codes are not directly related to original distance. Therefore, recently, machine learning frameworks have been exploited for learning small binary codes that explicitly approximate the original Euclidean distance [160; 178; 202]. With this approach, fewer bits are required to represent each sample. For example, spectral hashing [202] learns binary codes in an unsupervised manner, using a spectral graph analysis of a uniform distribution.

In a learning based approach, we can naturally integrate label (text) information by considering a supervised framework. A pioneering work is BoostSSC [165], which is based on AdaBoost [62]. Moreover, Torralba *et al.* applied a restricted Boltzmann machine (RBM) [78; 160], a technique developed for information retrieval, to image retrieval [178]. They showed that the RBM can more accurately compress GIST features [144]. However, it has been pointed out that the training cost of an RBM is extremely high.

Binary code learning is now an active research field. Recent works focus on a variety of topics such as semi-supervised learning [115] and online code learning [194].

E.3. Our Approach

We apply unsupervised hashing methods to the semantic subspace obtained by the CCD framework. In general, images and texts are already embedded in a low-dimensional subspace (latent space) relaxing the semantic gap. Our objective is to learn small binary codes that approximate the Euclidean distance in the latent space. In the retrieval phase, an image-only query is also coded as a binary vector. Then the nearest samples are retrieved in terms of Hamming distance. Further, using short codes (up to approximately 30 bits), we can directly exploit them in building a hash table, resulting in an extremely fast retrieval whose computation time is constant in the number of samples [178].

As described in Section 4.2.3, KL divergence in the latent space can be computed in terms of Euclidean distance of $\boldsymbol{r}$ defined by Equations 4.18 and 4.19. For convenience, here we call them topic features. We can expect that $\boldsymbol{r}$ is embedded in a Euclidean space, where the semantic distance between samples can be computed in terms of Euclidean distance. We apply standard unsupervised hashing methods to topic features $\boldsymbol{r}$ and extract c bit binary codes. Note that topic features are zero-mean according to

the definition thereof.

Simple Binarization

As a baseline, we simply binarize each dimension of the topic features setting the threshold at zero, and use these as hash functions. Therefore, the code length c is equal to d in this approach.

Locality Sensitive Hashing (LSH)

Hash functions for LSH are defined by random projections of feature vectors. Typically, the following are used to obtain binary codes.

$$h(\boldsymbol{r}) = sign(\boldsymbol{w}^T \boldsymbol{r} + b), \tag{33}$$

where $\boldsymbol{w}$ is a random hyperplane whose components are independently sampled from a p-stable distribution (a Gaussian in this work) [42], and b is a random offset sampled from a uniform distribution. We randomly generate c hash functions to obtain c bit binary codes. Note that we fix $b = 0$ since it yielded the best performance in our experiments. Furthermore, it should be pointed out that for a dataset on the unit sphere, approximately balanced codes are given by $b = 0$ [35; 115].

Spectral Hashing (SH)

Let $W \in \mathcal{R}^{N \times N}$ denote the affinity matrix of N samples, where $W(i, j) = \exp(-\|\boldsymbol{r}_i - \boldsymbol{r}_j\|^2/\epsilon^2)$. Let $\{\boldsymbol{h}_i\}_{i=1}^N$ denote the c bit compressed binary vectors using c hash functions. We want the Hamming distance of $\boldsymbol{h}$ to approximate the Euclidean distance of $\boldsymbol{r}$. This problem can be formulated as follows.

$$\begin{aligned} \text{minimize:} \quad & \sum_{ij} W_{ij} \|\boldsymbol{h}_i - \boldsymbol{h}_j\|^2, \\ \text{subject to:} \quad & \boldsymbol{h}_i \in \{-1, 1\}^c, \quad \sum_i \boldsymbol{h}_i = 0, \\ & \frac{1}{N} \sum_i \boldsymbol{h}_i \boldsymbol{h}_i^T = I. \end{aligned} \tag{34}$$

This problem is NP-hard, even for $c = 1$ [202]. It becomes much harder in general cases where $c > 1$. However, by relaxing some constraints, we can easily obtain the solution using spectral graph decomposition. In particular, if data follow a multinomial unit distribution, we can derive an analytical solution including an out-of-samples extension, which enables rapid encoding [202]. Of course, real data do not generally

follow a unit distribution. However, it is reported in [202] that merely rotating data with PCA before hashing empirically leads to excellent hash codes. This indicates that uncorrelated data are suitable for spectral hashing. Since the components of topic features are uncorrelated, we can directly apply hash functions to them. For an implementation, we use the Matlab code provided by the authors of [202].

E.4. Retrieval Experiments

We used two datasets. The first is the LabelMe dataset [159]. LabelMe images are manually segmented with a label given to each object. In this experiment, we only used the object labels, discarding their spatial information. Of the publicly available data, we used 60,000 samples for the retrieval database, and 1,191 samples as queries.

For the second dataset, we used images downloaded from Flickr. This dataset consists of 12.3 million images and 4,130 words (Flickr12M). For further details, refer to Section 7.1. We used a further 5,000 images as queries.

For the textual features ($\boldsymbol{y}$), we used label histograms. To implement CCD, we set the best canonical dimension d experimentally.

LabelMe Image Retrieval

We used GIST [144] as image features. Following [178], we evaluated the percentage of the 50 true neighbors included in the n retrieved images. True neighbors are defined in terms of the χ-square distance of the label histograms. Figure 20 shows the scores ($n = 5000$) for a varying number of bits for coding, while Figure 21 shows the scores as a function of the number of images retrieved (n). Overall, CCD based hashing methods substantially outperform unsupervised methods. Using only dozens of bits, they can achieve comparable performance to full GIST. In particular, CCD+LSH show consistently improved performance as the number of bits increases. Figure 22 shows some examples of retrieved images. With more bits, our method can retrieve semantically similar images, rather than visually similar images.

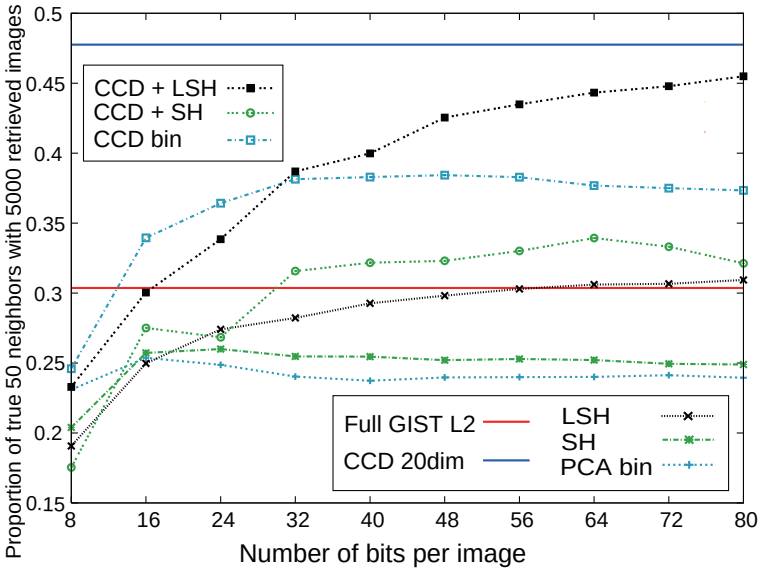


Figure 20: Retrieval performance with a varying number of bits for the LabelMe dataset.

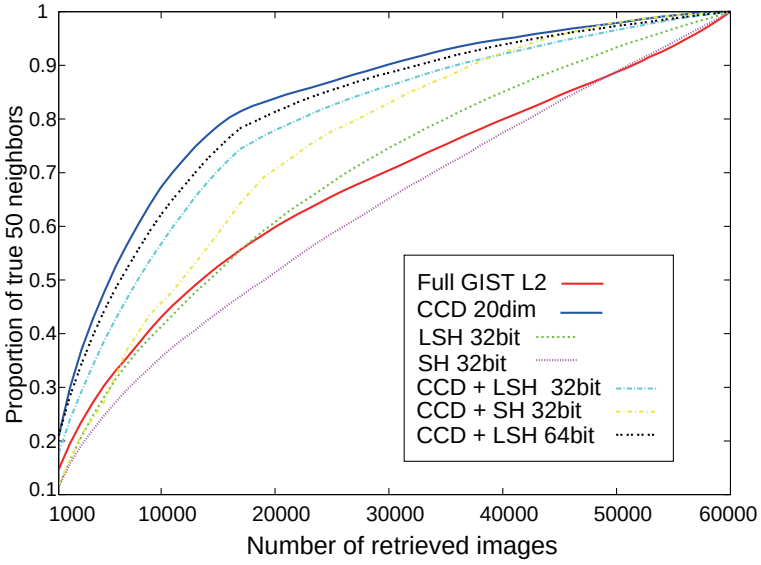


Figure 21: Retrieval performance as a function of retrieved images for the LabelMe dataset.

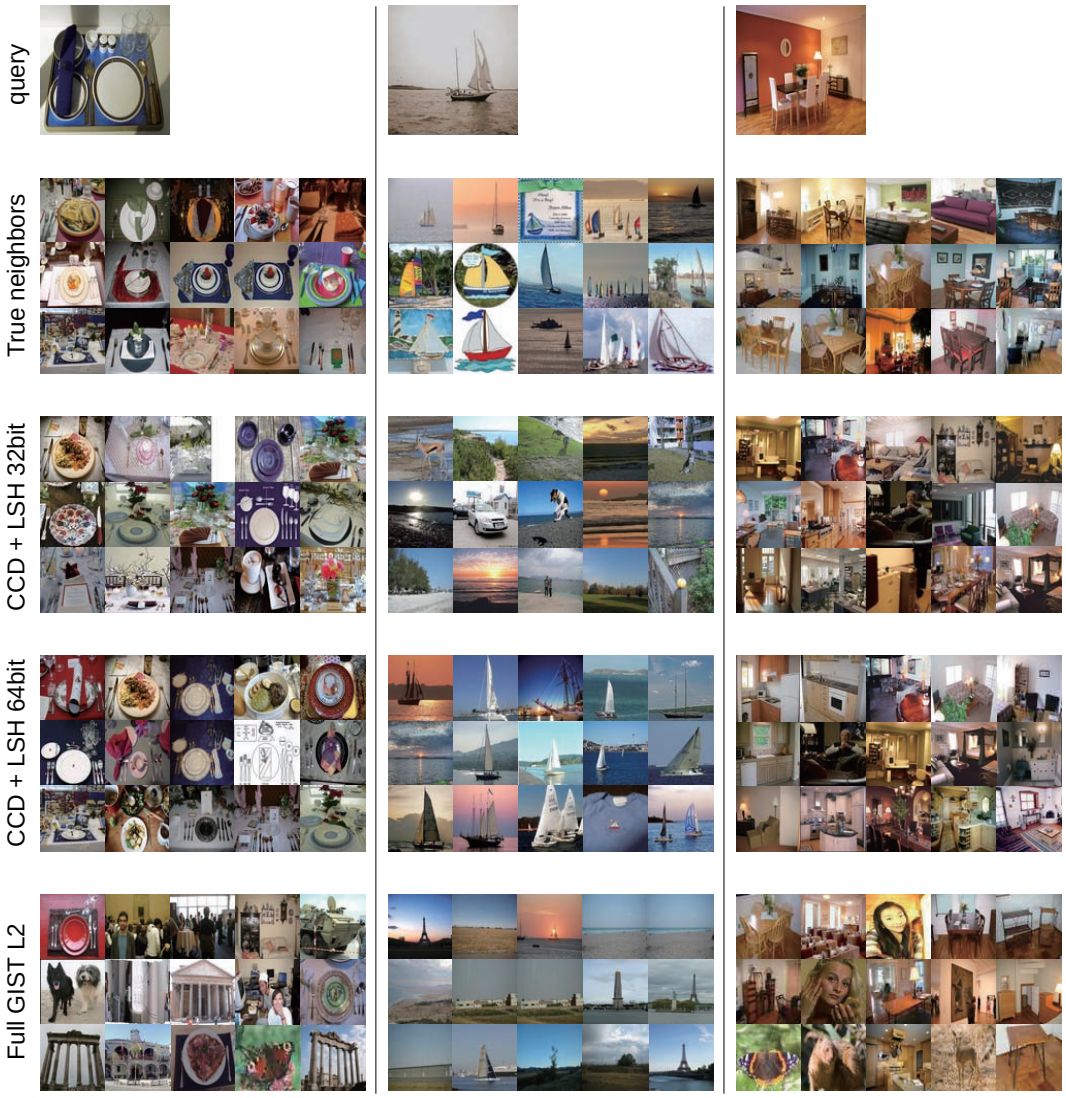


Figure 22: Examples of retrieved images (15 neighbors) for the LabelMe dataset.

Flickr Image Retrieval

Since label histograms are binary in this setup, we define true neighbors as follows. First, images are sorted in descending order of the number of matched labels. Then, they are further sorted in ascending order of the number of mismatched labels. For image features, we used bag-of-visual-words (BoVW) [40]. We created 1000 visual words using the standard k-means algorithm. To evaluate full BoVW, we used the χ-square distance for retrieval. Moreover, it has been noted in [151] that a linear kernel of the square root of BoVW is equivalent to a non-linear Bhattacharyya kernel of the raw BoVW. This means that square rooted BoVW is better suited to linear methods. Therefore, we used square rooted BoVW as the learning method. Figures 23 and 24 show the results[1]. We see that the CCD based hashing methods again outperform unsupervised ones. In this large-scale dataset, we need more bits to retrieve semantically similar images. While SH based methods are relatively effective when a small number of bits are used, their performance stagnates at an early stage. Although these methods are computationally effective, none of them can outperform full BoVW. On the contrary, CCD+LSH show consistently improved performance and achieve reasonable performance with more than 128 bits. Figure 25 illustrates some qualitative examples.

[1]Note that scores may seem low because ground truth labels themselves are noisy social tags.

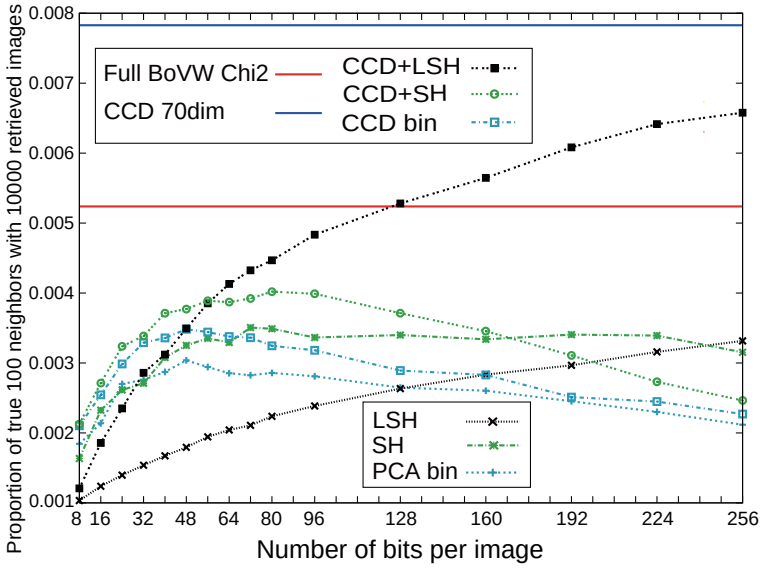


Figure 23: Retrieval performance with a varying number of bits for the Flickr12M dataset.

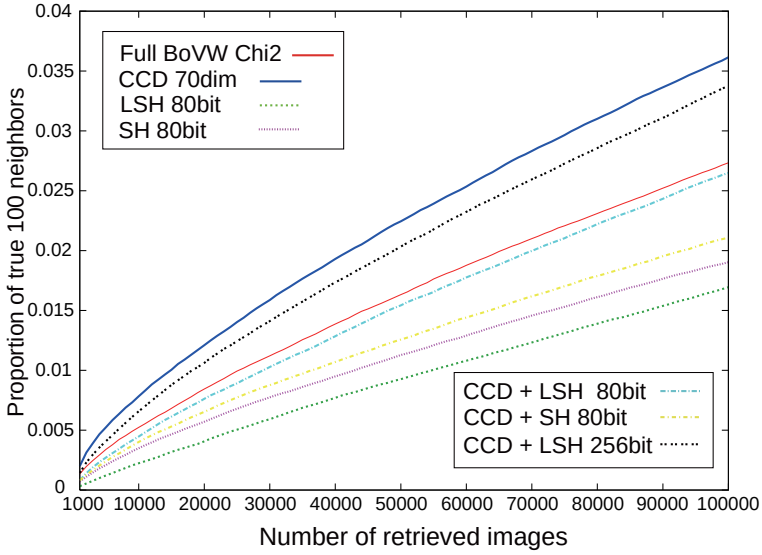


Figure 24: Retrieval performance as a function of retrieved images for the Flickr12M dataset.

APPENDIX E: HASHING-BASED RAPID ANNOTATION

Figure 25: Examples of retrieved images (15 neighbors) for the Flickr12M dataset.

Table 1: Retrieval time per image for Flickr12M (s) using a single CPU.

Full BoVW Chi2	219
CCD 70dim	6.2
32 bit code	0.16
80 bit code	0.21
256 bit code	0.44

Table 2: Computation time for training with the Flickr12M dataset using an 8-core desktop machine.

PCA	31m
CCD	4h 31m
SH	5h 7m
LSH	14m
CCD+SH	5h 42m
CCD+LSH	4h 34m

Computation Time

Table 1 gives the computation time per query for retrieving the nearest images in the Flickr12M dataset using a single CPU (3.20 GHz). Note that feature extraction time is not included. Also, we omit the binary coding time since it is negligible compared to searching time. Using small codes, we can query 12 million images in less than a second, even using a linear search. Table 2 gives the training time for each method for the Flickr12M dataset using an 8-core desktop PC. Although CCD based methods are more expensive than unsupervised methods, they can finish the training phase in several hours on a single machine. This is satisfactory considering the scale of the task.

E.5. Annotation Experiments

Next, we apply the above mentioned hashing-based retrieval methods to k-NN annotation using the Flickr12M dataset. We follow the same experimental setup as in Chapter 7. For image features, we use the concatenation of HLAC + SURF GLC + SURF BoVW-sqrt, which showed the best performance (see Section 7.3). We refer to this as "All features". We first compress the image features using $d = 200$ CCD dimensions, and then apply the hashing methods.

Figures 26 and 27 show the annotation performance as a function of hash length c.

As baselines, we also test normal CCD on several image features. All methods use the full Flickr12M training dataset. Overall, SH is superior to LSH when a small number of bits are allowed. However, its performance drops with more than 256 bits and becomes worse than that for LSH. This result is similar to that which we observed in retrieval experiments. It seems that 256~512-bit codes allow a good trade-off between annotation accuracy and computational costs. They outperform all single-feature CCDs, yet the entire data fits into 375~750 MB memory.

It should be noted that we can also control the trade-off for normal CCD by tuning the latent dimension d and the size of the training dataset. This is an important issue when considering the effectiveness of the hashing approach for annotation problems. Therefore, we investigate annotation accuracy of CCD varying d and the dataset size. We plot the scores as a function of total memory use in Figures 28 and 29. We superimpose the results of the hashing based methods for comparison. Clearly, the hashing approach gives a better trade-off, especially in terms of F_W.

Thus, integrating hashing methods is quite effective, not only for CBIR, but also for annotation.

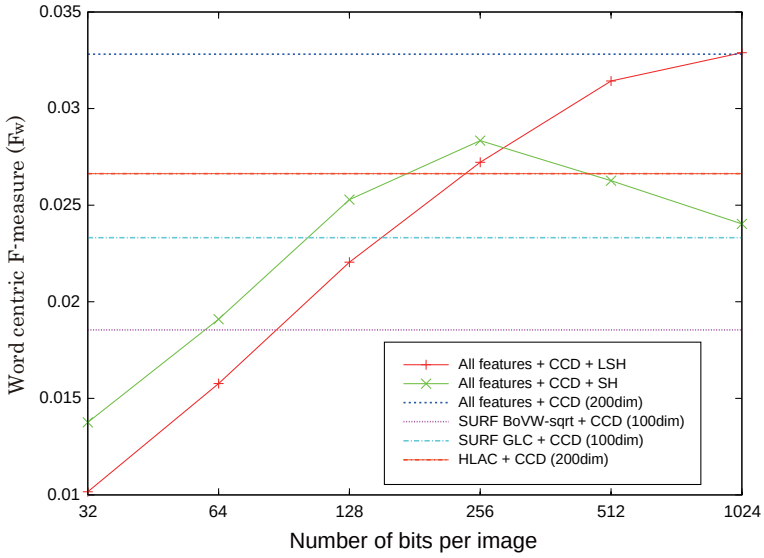


Figure 26: Annotation scores (F_W) with a varying number of bits for the full Flickr12M dataset.

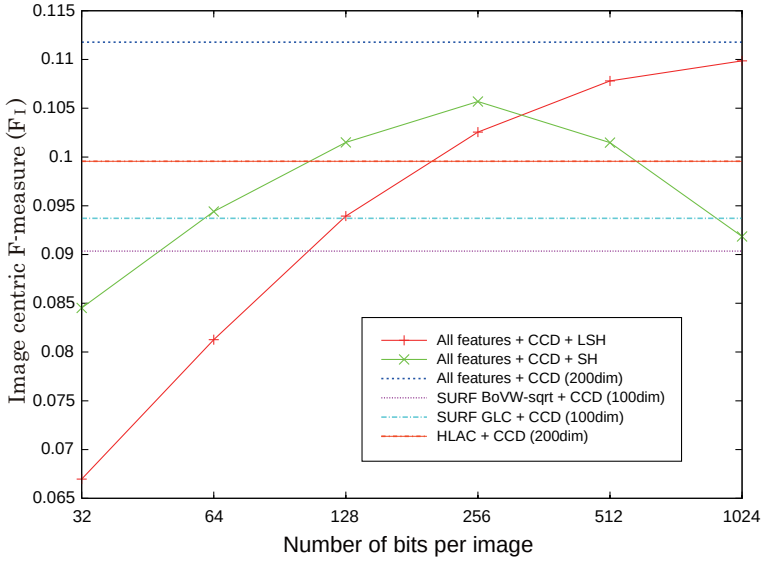


Figure 27: Annotation scores (F_I) with a varying number of bits for the full Flickr12M dataset.

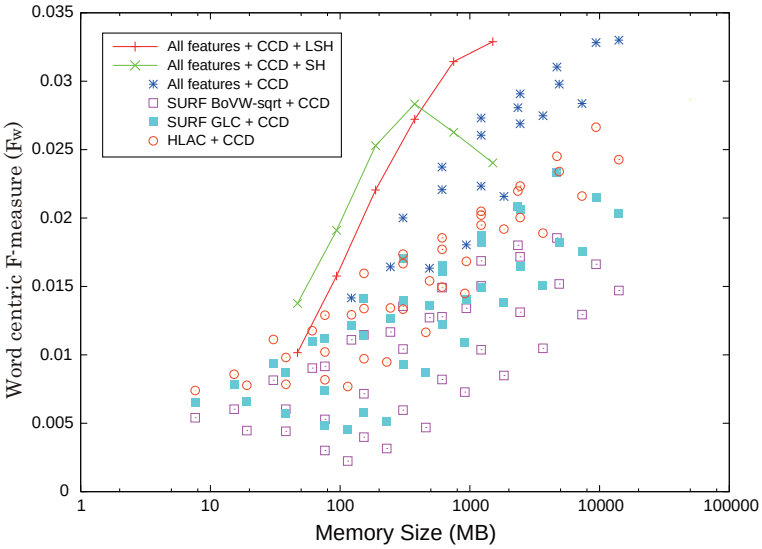


Figure 28: Annotation scores (F_W) with a varying amount of memory (MB).

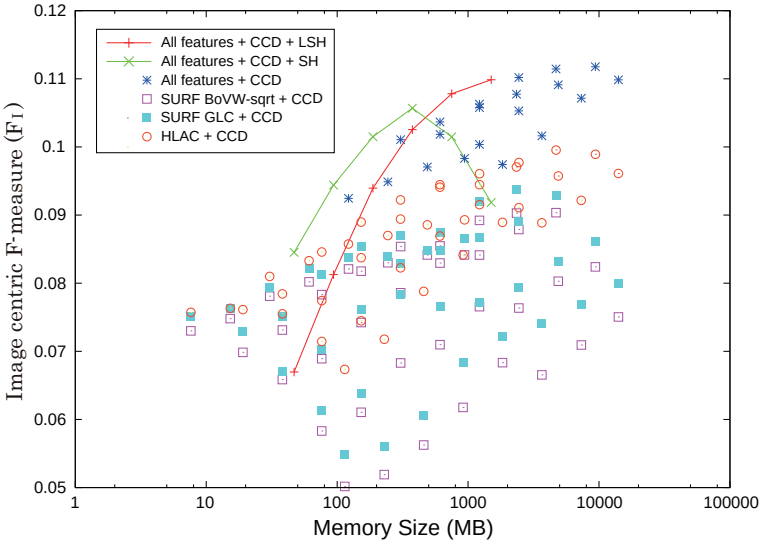


Figure 29: Annotation scores (F_I) with a varying amount of memory (MB).

References

[1] ImageCLEF home page. http://ir.shef.ac.uk/imageclef/. 15

[2] S. Akaho. The e-PCA and m-PCA: Dimension reduction of parameters by information geometry. In *Proceedings of International Joint Conference on Neural Networks*, 2004. 81

[3] S. Amari. Information geometry of the EM and em algorithms for neural networks. *Neural Networks*, **8**, 1995. 77

[4] S. Amari and H. Nagaoka. *Methods of Information Geometry*. AMS and Oxford University Press, 2000. 73, 77

[5] F. R. Bach and M. I. Jordan. A probabilistic interpretation of canonical correlation analysis. Technical Report 688, Department of Statistics, University of California, Berkeley, 2005. 44

[6] M. Banko and E. Brill. Scaling to very very large corpora for natural language disambiguation. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, pages 26–33, 2001. 19

[7] K. Barnard, P. Duygulu, and D. Forsyth. Clustering art. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages II:434–439, 2001. 23, 30

[8] K. Barnard, P. Duygulu, D. Forsyth, N. de Freitas, D. M. Blei, and M. I. Jordan. Matching words and pictures. *Journal of Machine Learning Research*, **3**:1107–1135, 2003. 30

[9] K. Barnard and D. Forsyth. Learning the semantics of words and pictures. In *Proceedings of IEEE International Conference on Computer Vision*, pages II:408–415, 2001. 23, 30

[10] H. Bay, A. Ess, T. Tuytelaars, and L. van Gool. SURF: Speeded up robust features. *Computer Vision and Image Understanding*, **110**[3]:346–359, 2008. 9, 84

[11] P. R. Beaudet. Rotationally invariant image operators. In *Proceedings of the 4th International Joint Conference on Pattern Recognition*, 1978. 8

[12] J. L. Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, **18**[9]:509–517, 1975. 102, 156

[13] A. Berg, J. Deng, and L. Fei-Fei. ImageNet large scale visual recognition challenge 2010. http://image-net.org/challenges/LSVRC/2010/index. 18

[14] T. L. Berg and D. A. Forsyth. Animals on the web. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2006. 19

[15] A. Berger, V. D. Pietra, and S. D. Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, **22**[1]:39–71, 1996. 24

[16] I. Biederman. Human image understanding: recent research and a theory. *Computer Vision, Graphics and Image Processing*, **32**[1]:29–73, 1985. 2, 7

[17] T. O. Binford. Spatial understanding: the successor system. In *Proceedings of IEEE conference on Systems and Control*, 1971. 7

[18] M. B. Blaschko and C. H. Lampert. Correlational spectral clustering. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2008. 38

[19] D. M. Blei and M. I. Jordan. Modeling annotated data. In *Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 127–134, 2003. 30

[20] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, **3**:993–1022, 2003. 29

[21] O. Boiman, E. Shechtman, and M. Irani. In defense of nearest-neighbor based image classification. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2008. 74

[22] M. Borga, T. Landelius, and H. Knutsson. A unified approach to PCA, PLS, MLR and CCA. Report LiTH-ISY-R-1992, ISY, 1997. 37, 38

[23] A. Bosch, A. Zisserman, and X. Muñoz. Scene classification via pLSA. In *Proceedings of European Conference on Computer Vision*, pages 517–530, 2006. 31

[24] A. Bosch, A. Zisserman, and X. Muñoz. Image classification using random forests and ferns. In *Proceedings of IEEE International Conference on Computer Vision*, 2007. 89

[25] A. Bosch, A. Zisserman, and X. Muñoz. Scene classification using a hybrid generative/discriminative approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **30**[4]:712–727, 2008. 31, 84, 89, 92, 98, 101, 103, 104

[26] R. Brooks. Symbolic reasoning among 3D models and 2D images. *Artificial Intelligence Journal*, **17**:285–348, 1982. 7

[27] P. Brown, V. D. Pietra, S. D. Pietra, and R. Mercer. The mathematics of statistical machine translation: parameter estimation. *Computational Linguistics*, **19**[2]:263–311, 1993. 23

[28] M. Calonder, V. Lepetit, and P. Fua. Keypoint signatures for fast learning and recognition. In *Proceedings of European Conference on Computer Vision*, 2008. 9

[29] G. Carneiro, A. B. Chan, P. J. Moreno, and N. Vasconcelos. Supervised learning of semantic classes for image annotation and retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **29**[3]:394–410, 2007. ix, 26, 27, 33, 68

[30] G. Carneiro and N. Vasconcelos. A database centric view of semantic image annotation and retrieval. In *Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, 2005. 26

[31] G. Carneiro and N. Vasconcelos. Formulating semantic image annotation as a supervised learning problem. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2005. 26

[32] C. Carson, S. Belongie, H. Greenspan, and J. Malik. Blobworld: Image segmentation using expectation-maximization and its application to image querying. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **24**[8]:1026–1038, 2002. 23

[33] C.-C. Chang and C.-J. Lin. *LIBSVM: A library for support vector machines*, 2001. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm. 82

[34] E. Chang, K. Goh, G. Sychay, and G. Wu. CBSA: content-based soft annotation for multimodal image retrieval using Bayes point machines. *IEEE Transactions on Circuits and Systems for Video Technology*, **13**[1]:26–38, 2003. 28

[35] M. Charikar. Similarity estimation techniques from rounding algorithms. In *Proc. ACM Ann. Symp. Theory of Computing*, 2002. 157, 158

[36] G. Chechik, V. Sharma, U. Shalit, and S. Bengio. An online algorithm for large scale image similarity learning. In *Proceedings of Conf. Advances in Neural Information Processing Systems*, 2009. 35

[37] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y.-T. Zheng. NUS-WIDE: a real-world web image database from National University of Singapore. In *Proceedings of ACM International Conference on Image and Video Retrieval*, 2009. 53, 54, 55

[38] R. L. Cilibrasi and P. M. B. Vitanyi. The Google similarity distance. *IEEE Transactions on Knowledge and Data Engineering*, **19**[3]:370–383, 2007. 19

[39] V. Clément and M. Thonnat. A knowledge-based approach to integration of image processing procedures. *Computer Vision, Graphics and Image Processing*, **57**[2]:166–184, 1993. 8

[40] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *Proceedings of ECCV Workshop on Statistical Learning in Computer Vision*, 2004. 9, 31, 32, 74, 75, 135, 162

[41] C. Cusano, G. Ciocca, and R. Schettini. Image annotation using SVM. In *Proceedings of Internet Imaging IV*, **SPIE**, 2004. 28

[42] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *Proceedings of ACM Symposium on Computational Geometry*, pages 253–262, 2004. 102, 156, 158

[43] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon. Information-theoretic metric learning. In *Proceedings of International Conference on Machine Learning*, pages 209–216, 2007. 35

[44] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, **41**[6]:391–407, 1990. 29

[45] J. Deng, A. Berg, K. Li, and L. Fei-Fei. What does classifying more than 10,000 image categories tell us? In *Proceedings of European Conference on Computer Vision*, 2010. 10, 18

[46] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: a large-scale hierarchical image database. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2009. 10, 13, 18, 19, 124

[47] C. Desai, D. Ramanan, and C. Fowlkes. Discriminative models for multi-class object layout. In *Proceedings of IEEE International Conference on Computer Vision*, pages 229–236, 2009. 13

[48] A. Deshpande, L. Rademacher, S. Vempala, and G. Wang. Matrix approximation and projective clustering via volume sampling. In *Proceedings of Symposium on Discrete Algorithms*, 2006. 48

[49] H. Drucker, C. J. C. Burges, L. Kaufman, A. Smola, and V. Vapnik. Support vector regression machines. In *Proceedings of Conf. Advances in Neural Information Processing Systems*, 1996. 67

[50] P. Duygulu, K. Barnard, N. de Freitas, and D. Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *Proceedings of European Conference on Computer Vision*, pages 97–112, 2002. ix, 14, 16, 23, 32, 33, 53, 56, 68, 127

[51] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2009 (VOC2009) Results. http://www.pascal-network.org/challenges/VOC/voc2009/. 15

[52] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL visual object classes (VOC) chellenge. *International Journal of Computer Vision*, **88**[2]:303–338, 2010. 15

[53] J. Fan, Y. Shen, N. Zhou, and Y. Gao. Harvesting large-scale weakly-tagged image databases from the web. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2010. 19

[54] L. Fei-Fei, R. Fergus, and P. Perona. A Bayesian approach to unsupervised one-shot learning of object categories. In *Proceedings of IEEE International Conference on Computer Vision*, pages 1134–1141, 2003. 9

[55] 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 *Proceedings of IEEE CVPR Workshop on Generative-Model Based Vision*, 2004. ix, 15, 16, 92

[56] L. Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **28**[4]:594–611, 2006. ix, 15, 16

[57] L. Fei-Fei and P. Perona. A Bayesian hierarchical model for learning natural scene categories. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 524–531, 2005. 31, 84, 92

[58] C. Fellbaum. *WordNet: An electronic lexical database*. Bradford Books, 1998. 10, 18, 19, 124

[59] S. Feng, R. Manmatha, and V. Lavrenko. Multiple Bernoulli relevance models for image and video annotation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2004. ix, 24, 25, 31, 33, 68

[60] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. Learning object categories from Google's image search. In *Proceedings of IEEE International Conference on Computer Vision*, pages 1816–1823, 2005. 19

[61] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 264–271, 2003. 9

[62] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, **55**[1]:119–139, 1997. 157

[63] A. Frome, Y. Singer, and J. Malik. Image retrieval and classification using local distance functions. In *Proceedings of Conf. Advances in Neural Information Processing Systems*, 2006. 35

[64] A. Frome, Y. Singer, F. Sha, and J. Malik. Learning globally-consistent local distance functions for shape-based image retrieval and classification. In *Proceedings of IEEE International Conference on Computer Vision*, 2007. 35

[65] P. Gehler and S. Nowozin. On feature combination for multiclass object classification. In *Proceedings of IEEE International Conference on Computer Vision*, pages 221–228, 2009. 15

[66] A. Globerson and S. Roweis. Metric learning by collapsing classes. In *Proceedings of Conf. Advances in Neural Information Processing Systems*, pages 451–458, 2006. 35

[67] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov. Neighbourhood components analysis. In *Proceedings of Conf. Advances in Neural Information Processing Systems*, pages 513–520, 2005. 35

[68] K. Grauman and T. Darrell. The pyramid match kernel: Efficient learning with sets of features. *Journal of Machine Learning Research*, **8**:725–760, 2007. 103

[69] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology, 2007. ix, 15, 17

[70] G. Griffin and P. Perona. Learning and using taxonomies for fast visual categorization. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2008. 10

[71] M. Guillaumin, T. Mensink, J. Verbeek, and C. Schmid. Tagprop: Discriminative metric learning in nearest neighbor models for image auto-annotation. In *Proceedings of IEEE International Conference on Computer Vision*, pages 309–316, 2009. 32, 33, 53, 54, 55, 66, 68, 69, 70, 71, 110

[72] M. Guillaumin, J. Verbeek, and C. Schmid. Is that you? Metric learning approaches for face identification. In *Proceedings of IEEE International Conference on Computer Vision*, pages 498–505, 2009. 35

[73] A. Guzmán. Analysis of curved line drawings using context and global information. In B. Meltzer and D. Michie, editors, *Machine Intelligence 6*, pages 325–375. John Wiley & Sons, 1971. 7

[74] D. R. Hardoon, C. Saunders, S. Szedmak, and J. Shawe-Taylor. A correlation approach for automatic image annotation. In *Proceedings of International Conference on Advanced Data Mining and Applications*, 2006. 29, 38, 51

[75] C. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of the 4th Alvey Vision Conference*, pages 147–151, 1988. 8, 84

[76] X. He and P. Niyogi. Locality preserving projections. In *Proceedings of Conf. Advances in Neural Information Processing Systems*, 2003. 35

[77] N. Hervé and N. Boujemaa. Image annotation: which approach for realistic databases? In *Proceedings of ACM International Conference on Image and Video Retrieval*, 2007. 102, 103

[78] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Nature*, **313**[5786]:504–507, 2006. 157

[79] T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, **42**:177–196, 2001. 29

[80] S. C. H. Hoi, W. Liu, M. R. Lyu, and W.-Y. Ma. Learning distance metrics with contextual constraints for image retrieval. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 2072–2078, 2006. 35

[81] D. Hoiem, A. A. Efros, and M. Hebert. Putting objects in perspective. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 2137–2144, 2006. 10

[82] H. Hotelling. Relations between two sets of variates. *Biometrika*, **28**[3/4]:321–377, 1936. 37

[83] M. Hu. Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, **8**[2]:179–187, 1962. 7

[84] S. Ikeda, T. Tanaka, and S. Amari. Stochastic reasoning, free energy, and information geometry. *Neural Computation*, **16**:1779–1810, 2004. 77

[85] P. Indyk, R. Motwani, P. Raghavan, and S. Vempala. Locality-preserving hashing in multidimensional spaces. In *Proceedings of ACM Symposium on Theory of Computing*, pages 618–625, 1997. 156

[86] S. Ioffe. Probabilistic linear discriminant analysis. In *Proceedings of European Conference on Computer Vision*, pages 531–542, 2006. 82

[87] P. Jain, B. Kulis, I. S. Dhillon, and K. Grauman. Online metric learning and fast similarity search. In *Proceedings of Conf. Advances in Neural Information Processing Systems*, 2008. 35

[88] J. Jeon, V. Lavrenko, and R. Manmatha. Automatic image annotation and retrieval using cross-media relevance models. In *Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, 2003. 24, 33, 68, 127

[89] J. Jeon and R. Manmatha. Using maximum entropy for automatic image annotation. In *Proceedings of International Conference on Image and Video Retrieval*, pages 24–32, 2004. 24, 33, 68

[90] Y. Jing, S. Baluja, and H. Rowley. Canonical image selection from the web. In *Proceedings of International Conference on Image and Video Retrieval*, 2007. ix, 11

[91] F. Jurie and B. Triggs. Creating efficient codebooks for visual recognition. In *Proceedings of IEEE International Conference on Computer Vision*, 2005. 75

[92] F. Kang, R. Jin, and R. Sukthankar. Correlated label propagation with application to multi-label learning. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1719–1726, 2006. 31

[93] T. Kato, T. Kurita, N. Otsu, and K. Hirata. A sketch retrieval method for full color image database –query by visual example–. In *Proceedings of International Conference on Pattern Recognition*, **1**, pages 530–533, 1992. 135

[94] Y. Ke and R. Sukthankar. PCA-SIFT: A more distinctive representation of local image descriptors. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, **2**, pages 506–513, 2004. 9, 100

[95] J. Kettenring. Canonical analysis of several sets of variables. *Biometrika*, **58**[3]:433–451, 1971. 125

[96] B. Kulis and K. Grauman. Kernelized locality-sensitive hashing for scalable image search. In *Proceedings of IEEE International Conference on Computer Vision*, pages 2130–2137, 2009. 156

[97] B. Kulis, P. Jain, and K. Grauman. Fast similarity search for learned metrics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **31**[12]:2143–2157, 2009. 156

[98] S. Kumar, M. Mohri, and A. Talwalkar. Ensemble Nyström method. In *Proceedings of Conf. Advances in Neural Information Processing Systems*, 2010. 48

[99] A. Kutics, A. Nakagawa, and M. Nakajima. Image retrieval via connecting words to salient objects. In *Proceedings of IEEE International Conference on Image Processing*, 2003. 10

[100] L. Ladický, C. Russell, P. Kohli, and P. H. S. Torr. Associative hierarchical CRFs for object class image segmentation. In *Proceedings of IEEE International Conference on Computer Vision*, 2009. 14

[101] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: probabilistic models for segmenting and labeling sequence data. In *Proceedings of International Conference on Machine Learning*, 2001. 14

[102] C. H. Lampert, M. B. Blaschko, and T. Hofmann. Beyond sliding windows: object localization by efficient subwindow search. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2008. 14

[103] G. R. G. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, and M. I. Jordan. Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research*, **5**:27–72, 2004. 9, 29, 32, 67

[104] V. Lavrenko, M. Choquette, and W. B. Croft. Cross-lingual relevance models. In *Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, 2002. 24

[105] V. Lavrenko, R. Manmatha, and J. Jeon. A model for learning the semantics of pictures. In *Proceedings of Conf. Advances in Neural Information Processing Systems*, 2003. 24, 31, 33, 42, 68

[106] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2006. x, 9, 81, 82, 85, 92, 101, 102, 103

[107] D. T. Lee and C. K. Wong. Worst-case analysis for region and partial region searches in multidimensional binary search trees and balanced quad trees. *Acta Informatica*, **9**[1]:23–29, 1977. 156

[108] Y. J. Lee and K. Grauman. Object-graphs for context-aware category discovery. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2010. 10

[109] B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *Proceedings of ECCV Workshop on Statistical Learning in Computer Vision*, 2004. 14

[110] L.-J. Li, G. Wang, and L. Fei-Fei. OPTIMOL: automatic online picture collection via incremental model learning. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2007. 19

[111] Li-Jia Li and L. Fei-Fei. What, where and who? Classifying events by scene and object recognition. In *Proceedings of IEEE International Conference on Computer Vision*, 2007. x, 81, 82, 89

[112] M. Li, J. T. Kwok, and B.-L. Lu. Making large-scale Nyström approximation possible. In *Proceedings of International Conference on Machine Learning*, 2010. 48

[113] R. Lienhart, S. Romberg, and E. Hörster. Multilayer pLSA for multimodal image retrieval. In *Proceedings of International Conference on Image and Video Retrieval*, 2009. 31

[114] R. Lienhart and M. Slaney. PLSA on large scale image databases. In *Proceedings of IEEE International Conference on Acoustic, Speech and Signal Processing*, **4**, pages 1217–1220, 2007. 31

[115] R.-S. Lin, D. A. Ross, and J. Yagnik. SPEC hashing: similarity preserving algorithm for entropy-based coding. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2010. 157, 158

[116] J. Liu, M. Li, Q. Liu, H. Lu, and S. Ma. Image annotation via graph learning. *Pattern Recognition*, **42**:218–228, 2009. 28, 33, 68

[117] J. Liu, M. Li, W.-Y. Ma, Q. Liu, and H. Lu. An adaptive graph model for automatic image annotation. In *Proceedings of ACM International Workshop on Multimedia Information Retrieval*, pages 61–70, 2006. 28, 33, 68

[118] J. Liu, B. Wang, M. Li, Z. Li, W.-Y. Ma, H. Lu, and S. Ma. Dual cross-media relevance model for image annotation. In *Proceedings of ACM International Conference on Multimedia*, pages 605–614, 2007. 19, 31, 33, 68

[119] N. Loeff and A. Farhadi. Scene discovery by matrix factorization. In *Proceedings of European Conference on Computer Vision*, **451-464**, 2008. 28, 33, 68

[120] D. G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of IEEE International Conference on Computer Vision*, pages 1150–1157, 1999. 8, 54, 74, 84, 92

[121] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, **60**[2]:91–110, 2004. 8

[122] Z. Lu, Horace H. S. Ip, and Q. He. Context-based multi-label image annotation. In *Proceedings of International Conference on Image and Video Retrieval*, 2009. 31, 33, 68

[123] S. Maji and A. C. Berg. Max-margin additive classifiers for detection. In *Proceedings of IEEE International Conference on Computer Vision*, pages 40–47, 2009. 89

[124] S. Maji and J. Malik. Object detection using a max-margin Hough transform. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2009. 14

[125] A. Makadia, V. Pavlovic, and S. Kumar. A new baseline for image annotation. In *Proceedings of European Conference on Computer Vision*, pages 316–329, 2008. ix, 15, 16, 31, 33, 53, 54, 68, 69, 71

[126] C. D. Manning and H. Schütze. *Foundation of Statistical Natural Language Processing*. The MIT Press, 1999. 9, 75

[127] D. Metzler and R. Manmatha. An inference network approach to image retrieval. In *Proceedings of International Conference on Image and Video Retrieval*, pages 42–50, 2004. 24, 33, 68

[128] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **27**[10]:1615–1630, 2005. 84

[129] F. Monay and D. Gatica-Perez. On image auto-annotation with latent space models. In *Proceedings of ACM International Conference on Multimedia*, pages 275–278, 2003. 31

[130] F. Monay and D. Gatica-Perez. PLSA-based image auto-annotation: constraining the latent space. In *Proceedings of ACM International Conference on Multimedia*, pages 348–351, 2004. 31

[131] F. Monay and D. Gatica-Perez. Modeling semantic aspects for cross-media image indexing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **29**[10]:1802–1817, 2007. 31

[132] P. J. Moreno, P. P. Ho, and N. Vasconcelos. A Kullback-Leibler divergence based kernel for SVM classification in multimedia applications. In *Proceedings of Conf. Advances in Neural Information Processing Systems*, 2003. 74, 75, 76, 80

[133] Y. Mori, H. Takahashi, and R. Oka. Image-to-word transformation based on dividing and vector quantizing images with words. In *Proceedings of First International Workshop on Multimedia Intelligent Storage and Retrieval Management*, 1999. 23, 33, 68

[134] N. Morioka and S. Satoh. Building compact local pairwise codebook with joint feature space clustering. In *Proceedings of European Conference on Computer Vision*, pages 692–705, 2010. 89, 90

[135] N. Morioka and S. Satoh. Learning directional local pairwise bases with sparse coding. In *Proceedings of British Machine Vision Conference*, 2010. 89, 102, 103

[136] H. Murase and S. K. Nayar. Visual learning and recognition of 3-D objects from appearance. *International Journal of Computer Vision*, **14**[9]:5–24, 1995. 8

[137] N. Murata, T. Takenouchi, T. Kanamori, and S. Eguchi. Information geometry of U-boost and Bregman divergence. *Neural Computation*, **16**:1437–1481, 2004. 77

[138] H. Nakayama, T. Harada, and Y. Kuniyoshi. Canonical contextual distance for large-scale image annotation and retrieval. In *Proceedings of the 1st ACM International Workshop on Large-Scale Multimedia Mining and Retrieval*, pages 3–10, 2009. 41, 45

[139] H. Nakayama, T. Harada, and Y. Kuniyoshi. Dense sampling low-level statistics of local features. In *Proceedings of ACM International Conference on Image and Video Retrieval*, 2009. 73

[140] H. Nakayama, T. Harada, and Y. Kuniyoshi. Evaluation of dimensionality reduction methods for image auto-annotation. In *Proceedings of British Machine Vision Conference*, 2010. 41, 45

[141] H. Nakayama, T. Harada, and Y. Kuniyoshi. Global Gaussian approach for scene categorization using information geometry. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2010. 73, 76, 102, 103

[142] H. Nakayama, T. Harada, Y. Kuniyoshi, and N. Otsu. High-performance image annotation and retrieval for weakly labeled images. In *Proceedings of Pacific-Rim Conference on Multimedia*, pages 601–610, 2008. 43

[143] E. Nowak, F. Jurie, and B. Triggs. Sampling strategies for bag-of-features image classification. In *Proceedings of European Conference on Computer Vision*, pages 490–503, 2006. 84, 98, 104

[144] A. Oliva and A. Torralba. Modeling the shape of the scene: a holistic representation of the spatial envelope. *International Journal of Computer Vision*, **42**[3]:145–175, 2001. 32, 54, 90, 92, 110, 157, 159

[145] N. Otsu and T. Kurita. A new scheme for practical, flexible and intelligent vision systems. In *Proceedings of IAPR Workshop on Computer Vision*, pages 431–435, 1988. 135

[146] J.-Y. Pan, H.-J. Yang, C. Faloutsos, and P. Duygulu. Automatic multimedia cross-modal correlation discovery. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 663–658, 2004. 28

[147] J.-Y. Pan, H.-J. Yang, C. Faloutsos, and P. Duygulu. GCap: Graph-based automatic image captioning. In *Proceedings of IEEE CVPR Workshop on Multimedia Data and Document Engineering*, 2004. 28

[148] A. Perina, M. Cristani, U. Castellani, V. Murino, and N. Jojic. A hybrid generative/discriminative classification framework based on free-energy terms.

In *Proceedings of IEEE International Conference on Computer Vision*, pages 2058–2065, 2009. 102, 103

[149] F. Perronnin and C. Dance. Fisher kernels on visual vocabularies for image categorization. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2007. 75

[150] F. Perronnin, C. R. Dance, G. Csurka, and M. Bressan. Adapted vocabularies for generic visual categorization. In *Proceedings of European Conference on Computer Vision*, pages 464–475, 2006. 75

[151] F. Perronnin, J. Sánchez, and Y. Liu. Large-scale image categorization with explicit data embedding. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2010. 48, 110, 162

[152] F. Perronnin, J. Sánchez, and T. Mensink. Improving the Fisher kernel for large-scale image classification. In *Proceedings of European Conference on Computer Vision*, 2010. 73

[153] J. Ponce, T. L. Berg, M. Everingham, D. A. Forsyth, M. Hebert, S. Lazebnik, M. Marszalek, C. Schmid, B. C. Russell, A. Torralba, C. K. I. Williams, J. Zhang, and A. Zisserman. Dataset issues in object recognition. In J. Ponce, M. Hebert, C. Schmid, and A. Zisserman, editors, *Toward category-level object recognition*, pages 29–48. Springer, 2006. 14

[154] J. Ponce, M. Hebert, C. Schmid, and A. Zisserman, editors. *Toward category-level object recognition*. LNCS 4170. Springer, 2006. 1

[155] A. R. Pope. Model-based object recognition: a survey of recent research. Report TR-94-04 TR-94-04, University of British Columbia, Computer Science Department, 1994. 8

[156] A. Quattoni and A. Torralba. Recognizing indoor scenes. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2009. x, 81, 82, 89, 90

[157] D. Ramanan and S. Baker. Local distance functions: a taxonomy, new algorithms and an evaluation. In *Proceedings of IEEE International Conference on Computer Vision*, pages 301–308, 2009. 35

[158] L. G. Roberts. Machine perception of three-dimensional solids. In J. Tippett, D. Berkowitz, L. Clapp, C. Koester, and A. Vanderburgh, editors, *Optical and Electro-optical Information processing*, pages 159–197. MIT Press, 1965. 7

[159] B. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. LabelMe: a database and web-based tool for image annotation. *International Journal of Computer Vision*, **77**[1-3]:157–173, 2008. 15, 159

[160] R. R. Salakhutdinov and G. E. Hinton. Semantic hashing. In *Proceedings of ACM SIGIR workshop on Information Retrieval and Applications of Graphical Models*, 2007. 157

[161] B. Schiele and J. L. Crowley. Recognition using multidimensional receptive field histograms. *Proceedings of European Conference on Computer Vision*, pages 610–619, 1996. 8

[162] B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, **10**[5]:1299–1319, 1998. 47

[163] F. Schroff, A. Criminisi, and A. Zisserman. Harvesting image databases from the web. In *Proceedings of IEEE International Conference on Computer Vision*, 2007. 19

[164] W. R. Schwartz, A. Kembhavi, D. Harwood, and L. S. Davis. Human detection using partial least squares analysis. In *Proceedings of IEEE International Conference on Computer Vision*, pages 24–31, 2009. 37

[165] G. Shakhnarovich, P. Viola, and T. Darrell. Fast pose estimation with parameter sensitive hashing. In *Proceedings of IEEE International Conference on Computer Vision*, pages 750–757, 2003. 157

[166] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **22**[8]:888–905, 2000. 23, 24

[167] J. Shotton, M. Johnson, and R. Cipolla. Semantic texton forests for image categorization and segmentation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2008. 76

[168] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost: joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *Proceedings of European Conference on Computer Vision*, 2006. 14

[169] L. Si, R. Jin, S. C. H. Hoi, and M. R. Lyu. Collaborative image retrieval via regularized metric learning. *Multimedia Systems*, **12**[1]:34–44, 2006. 35

[170] B. Siddiquie and A. Gupta. Beyond active noun tagging: modeling contextual interactions for multi-class active learning. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2010. 10

[171] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, **22**[12], 2000. 8, 11

[172] S. Sonnenburg, G. Rätsch, S. Henschel, C. Widmer, J. Behr, A. Zien, F. de Bona, A. Binder, C. Gehl, and V. Franc. The SHOGUN machine learning toolbox. *Journal of Machine Learning Research*, **11**:1799–1802, 2010. 67

[173] A. Sorokin and D. Forsyth. Utility data annotation with Amazon Mechanical Turk. In *Proceedings of CVPR workshop on Internet Vision*, 2008. 18, 124

[174] A. Stein and M. Hebert. Incorporating background invariance into feature-based object recognition. In *Proceedings of IEEE Workshop on Applications of Computer Vision*, pages 37–44, 2005. 9

[175] M. J. Swain and D. H. Ballard. Color indexing. *International Journal of Computer Vision*, **7**[1]:11–32, 1991. 8

[176] A. Talwalkar, S. Kumar, and H. Rowley. Large-scale manifold learning. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2008. 48

[177] A. Torralba, R. Fergus, and W. T. Freeman. 80 million tiny images: a large dataset for non-parametric object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **30**[11]:1958–1970, 2008. 10, 12, 13, 19, 110

[178] A. Torralba, R. Fergus, and Y. Weiss. Small codes and large image databases for recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2008. 157, 159

[179] A. Torralba, K. Murphy, and W. Freeman. Using the forest to see the trees: a graphical model relating features, objects and scenes. In *Proceedings of Conf. Advances in Neural Information Processing Systems*, 2003. 10

[180] M. Turk and A. Pentland. Eigenfaces for recognition. *Cognitive Neuroscience*, **3**[1]:71–96, 1991. 8

[181] H. Turtle and W. B. Croft. Evaluation of an inference network-based retrieval model. *ACM Transactions on Information Systems*, **9**:187–222, 1991. 24

[182] T. Tuytelaars and C. Schmid. Vector quantizing feature space with a regular lattice. In *Proceedings of IEEE International Conference on Computer Vision*, 2007. 76

[183] O. Tuzel, F. Porikli, and P. Meer. Region covariance: A fast descriptor for detection and classification. In *Proceedings of European Conference on Computer Vision*, pages 589–600, 2006. 74, 76

[184] O. Tuzel, F. Porikli, and P. Meer. Pedestrian detection via classification on Riemannian manifolds. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2007. 74, 76

[185] J. van de Weijer and C. Schmid. Coloring local feature extraction. In *Proceedings of European Conference on Computer Vision*, pages 334–348, 2006. 9, 54

[186] J. C. van Gemert, J.-M. Geusebroek, C. J. Veenman, and A. W. M. Smeulders. Kernel codebooks for scene categorization. In *Proceedings of European Conference on Computer Vision*, pages 696–709, 2008. 75

[187] N. Vasconcelos, P. P. Ho, and P. J. Moreno. The Kullback-Leibler kernel as a framework for discriminant and localized representations for visual recognition. In *Proceedings of European Conference on Computer Vision*, 2004. 75

[188] A. Vedaldi and A. Zisserman. Efficient additive kernels via explicit feature maps. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2010. 47, 89

[189] L. von Ahn and L. Dabbish. Labeling images with a computer game. In *Proceedings of ACM SIGCHI Conference on Human Factors in Computing Systems*, pages 319–326, 2004. 15, 18, 54

[190] L. von Ahn, R. Liu, and M. Blum. Peekaboom: a game for locating objects in images. In *Proceedings of ACM SIGCHI Conference on Human Factors in Computing Systems*, pages 55–64, 2006. 18

[191] C. Wang, S. Yan, L. Zhang, and H.-J. Zhang. Multi-label sparse coding automatic image annotation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2009. 31, 33, 68

[192] C. Wang, L. Zhang, and H.-J. Zhang. Learning to reduce the semantic gap in web image retrieval and annotation. In *Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 355–362, 2008. 35

[193] G. Wang and D. Forsyth. Joint learning of visual attributes, object classes and visual saliency. In *Proceedings of IEEE International Conference on Computer Vision*, pages 537–544, 2009. 28

[194] J. Wang, S. Kumar, and S.-F. Chang. Semi-supervised hashing for scalable image retrieval. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2010. 157

[195] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2010. 73, 76, 102, 103

[196] X. J. Wang, L. Zhang, F. Jing, and W. Y. Ma. Annosearch: Image auto-annotation by search. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1483–1490, 2006. 19

[197] X.-J. Wang, L. Zhang, M. Liu, Y. Li, and W.-Y. Ma. ARISTA - image search to annotation on billions of web photos. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2010. 12, 13, 19, 20, 105

[198] Y. Wang and S. Gong. Conditional random field for natural scene categorization. In *Proceedings of British Machine Vision Conference*, 2007. 101, 103

[199] R. Weber, H.-J. Schek, and S. Blott. A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In *Proceedings of International Conference on Very Large DataBases*, pages 194–205, 1998. 156

[200] K. Weinberger, J. Blitzer, and L. Saul. Distance metric learning for large margin nearest neighbor classification. In *Proceedings of Conf. Advances in Neural Information Processing Systems*, pages 1473–1480, 2006. 35

[201] K. Q. Weinberger and L. K. Saul. Fast solvers and efficient implementations for distance metric learning. In *Proceedings of International Conference on Machine Learning*, pages 1160–1167, 2008. 35

[202] Y. Weiss, A. Torralba, and R. Fergus. Spectral hashing. In *Proceedings of Conf. Advances in Neural Information Processing Systems*, 2008. 157, 158, 159

[203] C. K. I. Williams and M. Seeger. Using the Nyström method to speed up kernel machines. In *Proceedings of Conf. Advances in Neural Information Processing Systems*, pages 682–688, 2000. 48

[204] H. Wold. Partial least squares. In S. Kotz and N. Johnson, editors, *Encyclopedia of Statistical Sciences*, **6**, pages 581–591. John Wiley & Sons, 1985. 36

[205] J. Wu and J. M. Rehg. Beyond the Euclidean distance: Creating effective visual codebooks using the histogram intersection kernel. In *Proceedings of IEEE International Conference on Computer Vision*, pages 630–637, 2009. 75, 84, 89, 90, 103

[206] L. Wu, X.-S. Hua, N. Yu, W.-Y. Ma, and S. Li. Flickr distance. In *Proceedings of ACM International Conference on Multimedia*, pages 31–40, 2008. 19

[207] J. Xiao, J. Hays, K. Ehinger, A. Oliva, and A. Torralba. SUN database: large-scale scene recognition from abbey to zoo. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2010. 18, 89

[208] O. Yakhnenko and V. Honavar. Annotating images and image objects using a hierarchical Dirichlet process model. In *Proceedings of ACM SIGKDD workshop on Multimedia Data Mining*, 2008. 31

[209] O. Yakhnenko and V. Honavar. Multiple label prediction for image annotation with multiple kernel correlation models. In *Proceedings of IEEE CVPR workshop on Visual Context Learning*, 2009. 29, 38, 51, 67

[210] J. Yang, Y. Li, Y. Tian, L. Duan, and W. Gao. Group-sensitive multiple kernel learning for object categorization. In *Proceedings of IEEE International Conference on Computer Vision*, pages 436–443, 2009. 15, 89

[211] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2009. 73, 74, 76, 102, 103

[212] B. Yao, X. Yang, and S.-C. Zhu. Introduction to a large-scale general purpose ground truth database: methodology, annotation tool and benchmarks. In *Proceedings of CVPR workshop on Energy Minimization Methods*, pages 169–183, 2007. 18

[213] A. Yavlinsky, E. Schofield, and S. Rüger. Automated image annotation using global features and robust nonparametric density estimation. In *Proceedings of International Conferences on Image and Video Retrieval*, pages 507–517, 2005. 31, 33, 68

[214] H.-F. Yu, C.-J. Hsieh, K.-W. Chang, and C.-J. Lin. Large linear classification when data cannot fit in memory. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2010. 22

[215] J. Yuen, B. Russell, C. Liu, and A. Torralba. LabelMe video: building a video database with human annotations. In *Proceedings of IEEE International Conference on Computer Vision*, pages 1451–1458, 2009. 15

[216] M. Zerroug and R. Nevatia. From an intensity image to 3-d segmented descriptions. In J. Ponce, M. Hebert, and A. Zisserman, editors, *Object Representation in Computer Vision II*, pages 11–24. 1996. 7

[217] H. Zhang, A. C. Berg, M. Maire, and J. Malik. SVM-KNN: Discriminative nearest neighbor classification for visual category recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, **2**, pages 2126–2136, 2006. 103

[218] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: An in-depth study. Technical Report RR-5497, INRIA, 2005. 96

[219] J. Zhang, M. Marszałek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: a comprehensive study. *International Journal of Computer Vision*, **73**[2]:213–238, 2007. 47, 48

[220] S. Zhang, J. Huang, Y. Huang, Y. Yu, H. Li, and D. N. Metaxas. Automatic image annotation using group sparsity. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2010. 33, 68, 69, 71

[221] X. Zhou, N. Cui, Z. Li, F. Liang, and T. S. Huang. Hierarchical Gaussianization for image classification. In *Proceedings of IEEE International Conference on Computer Vision*, pages 1971–1977, 2009. 74, 75, 84, 89, 102, 103

[222] X. Zhou, K. Yu, T. Zhang, and T. S. Huang. Image classification using supervector coding of local image descriptors. In *Proceedings of European Conference on Computer Vision*, 2010. 73

[223] X. Zhou, X. Zhuang, H. Tang, M. Hasegawa-Johnson, and T. S. Huang. A novel Gaussianized vector representation for natural scene categorization. In *Proceedings of International Conference of Pattern Recognition*, 2008. 75

[224] 栗田多喜夫, 加藤俊一, 福田郁美, and 板倉あゆみ. 印象語による絵画データベースの検索. **情報処理学会論文誌**, **33**[11]:1373–1383, 1992. 29, 38, 51

[225] 岡部 孝弘, 近藤 雄飛, 木谷 クリス 真実, and 佐藤 洋一. カテゴリーの共起を考慮した回帰による複数物体認識. **電子情報通信学会論文誌** *D*, **J92-D**[8]:1115–1124, 2009. 29

[226] 中山英樹, 原田達也, and 國吉康夫. 大規模 web 画像のための画像アノテーション・リトリーバル手法-web 集合知からの自律的画像知識獲得へ向けて-. In **第** *12* **回画像の認識・理解シンポジウム** *(MIRU 2009)*, pages 55–62, 2009. 41, 45

[227] 中山英樹, 原田達也, 國吉康夫, AND 大津展之. 画像・単語間概念対応の確率構造学習を利用した超高速画像認識・検索方法. In **電子情報通信学会技術研究報告**, *PRMU2007-147*, pages 65–70, 2007. 43

[228] 柳井啓司. 一般画像自動分類の実現へ向けた world wide web からの画像知識の獲得. **人工知能学会誌**, **19**[5]:429–439, 2004. 19

[229] 柳井啓司. 一般物体認識の現状と今後. **情報処理学会論文誌：コンピュータビジョン・イメージメディア**, **48**[SIG16 (CVIM19)]:1–24, 2007. 1

Publications

Journal

1. 中山英樹, 原田達也, 國吉康夫, “大規模 Web 画像のための画像アノテーション・リトリーバル手法,” 電子情報通信学会論文誌, Vol.J93-D, No.8, pp.1267-1280, 2010.

2. Hideki Nakayama, Tatsuya Harada, and Yasuo Kuniyoshi, “Dense Sampling Low-Level Statistics of Local Features,” IEICE Transactions on Information and Systems, Vol.E93-D, No.7, pp.1727-1736, 2010.

3. Tatsuya Harada, Hideki Nakayama, Yasuo Kuniyoshi, and Nobuyuki Otsu, “Image Annotation and Retrieval for Weakly Labeled Images using Conceptual Learning,” New Generation Computing, Vol.28, No.3, pp.277-298, 2010.

4. 原田達也, 中山英樹, 國吉康夫, “AI Goggles: 追加学習機能を備えたウェアラブル画像アノテーション・リトリーバルシステム,” 電子情報通信学会論文誌, Vol.J93-D, No.6, pp.857-869, 2010.

Reviewed Conference

1. Hideki Nakayama, Tatsuya Harada, and Yasuo Kuniyoshi, “Evaluation of Dimensionality Reduction Methods for Image Auto-Annotation,” Proceedings of the 21st British Machine Vision Conference (BMVC 2010), Aberystwyth, United Kingdom, Sep., 2010.

2. Tatsuya Harada, Hideki Nakayama, and Yasuo Kuniyoshi, “Improving Local Descriptors by Embedding Global and Local Spatial Information,” Proceedings of the 11th European Conference on Computer Vision (ECCV 2010), Crete, Greece, Sep., 2010.

3. Hideki Nakayama, Tatsuya Harada, and Yasuo Kuniyoshi, “Global Gaussian Approach for Scene Categorization Using Information Geometry,” Proceedings of

the 23rd IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2010), San Francisco, USA, June, 2010.

4. Asako Kanezaki, Hideki Nakayama, Tatsuya Harada, and Yasuo Kuniyoshi, "High-speed 3D Object Recognition Using Additive Features in a Linear Subspace," Proceedings of the 2010 IEEE International Conference on Robotics and Automation (ICRA 2010), pp.3128-3134, Anchorage, USA, May, 2010.

5. Hideki Nakayama, Tatsuya Harada, and Yasuo Kuniyoshi, "Canonical Contextual Distance for Large-Scale Image Annotation and Retrieval," Proceedings of the 1st ACM International Workshop on Large-Scale Multimedia Mining and Retrieval (LS-MMRM 2009), pp.3-10, Beijing, China, Oct., 2009.

6. Hideki Nakayama, Tatsuya Harada, and Yasuo Kuniyoshi, "Dense Sampling Low-Level Statistics of Local Features," Proceedings of the 8th ACM International Conference on Image and Video Retrieval (CIVR 2009), Santorini, Greece, July, 2009.

7. Tatsuya Harada, Hideki Nakayama, and Yasuo Kuniyoshi, "Image Annotation and Retrieval Based on Efficient Learning of Contextual Latent Space," Proceedings of the 2009 IEEE International Conference on Multimedia & Expo (ICME 2009), pp.858-861, New York, USA, June, 2009.

8. Hideki Nakayama, Tatsuya Harada, and Yasuo Kuniyoshi, "AI Goggles: Real-time Description and Retrieval in the Real World with Online Learning," Proceedings of the 6th Canadian Conference on Computer and Robot Vision (CRV 2009), pp.184-191, Kelowna, Canada, May, 2009.

9. Hideki Nakayama, Tatsuya Harada, and Yasuo Kuniyoshi, "Scene Classification Using Generalized Local Correlation," Proceedings of the 11th IAPR Conference on Machine Vision Applications (MVA 2009), pp.195-198, Hiyoshi, Japan, May, 2009.

10. Hideki Nakayama, Tatsuya Harada, and Yasuo Kuniyoshi, "High-Performance Image Annotation and Retrieval for Weakly Labeled Images," Proceedings of the 2008 Pacific-Rim Conference on Multimedia (PCM 2008), LNCS 5353, pp.601-610, Tainan, Taiwan, Dec., 2008.

11. Rie Matsumoto, Hideki Nakayama, Tatsuya Harada, Yasuo Kuniyoshi, and Nobuyuki Otsu, "Journalist Robot: Robot System Making News Articles from Real World," Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2007), pp.1234-1241, San Diego, USA, Oct., 2007.

Reviewed Domestic Conference

1. 中山英樹, 原田達也, 國吉康夫, “大規模 Web 画像のための画像アノテーション・リトリーバル手法 -Web 集合知からの自律的画像知識獲得へ向けて-,” 第 12 回画像の認識・理解シンポジウム (MIRU 2009), pp.55-62, 松江, July, 2009.

2. 金崎朝子, 中山英樹, 原田達也, 國吉康夫, “部分空間法とカラー立体高次局所自己相関特徴を用いた高速三次元物体認識,” 第 12 回画像の認識・理解シンポジウム (MIRU 2009), pp.103-110, 松江, July, 2009.

3. 中山英樹, 原田達也, 國吉康夫, “画像情報からリアルタイムに実世界記述・検索を行うサイバーゴーグル,” 第 13 回ロボティクスシンポジア, pp.192-199, 高松, Mar., 2009.

4. 松本理恵, 中山英樹, 原田達也, 國吉康夫, “ジャーナリストロボット -実世界からの自律的ニュース探索と高性能な事象キーワードの記述-,” 第 13 回ロボティクスシンポジア, pp.73-80, 高松, Mar., 2009.

Un-reviewed Domestic Conference

1. 牛久祥孝, 中山英樹, 原田達也, 國吉康夫, “Web 画像と文章の大域的特徴から得る潜在的意味に基づくデータ検索　-Web 上での一般画像認識実現への新たなアプローチを目指して-,” 電子情報通信学会技術研究報告, PRMU2009-100, pp.45-50, 石川, Nov., 2009.

2. 原田達也, 松本理恵, 中山英樹, 國吉康夫, “ニュース性により記事生成を行うジャーナリストロボットの試み,” 第 27 回ロボット学会学術講演会, pp.2G1-07, 横浜, Sep., 2009.

3. 原田達也, 中山英樹, 國吉康夫, “自らの視覚記憶を言葉で検索可能とする AI Goggles,” 第 23 回人工知能学会全国大会, 高松, June., 2009.

4. 原田達也, 中山英樹, 國吉康夫, “超高速汎用的画像認識検索手法の開発と実世界応用,” 第 4 回デジタルコンテンツシンポジウム, 千葉, June., 2008.

5. 中山英樹, 原田達也, 國吉康夫, “サイバーゴーグル: 画像情報からリアルタイムに実世界記述・検索を行うシステム,” 情報処理学会第 70 回全国大会, pp.5-89 - 5-90, 筑波, Mar., 2008.

6. 松本理恵, 中山英樹, 原田達也, 國吉康夫, “ジャーナリストロボットシステム: 実世界からの自律的ニュース探索と事象の記述,” 情報処理学会第 70 回全国大会, pp.5-91 - 5-92, 筑波, Mar., 2008.

7. 原田達也, 中山英樹, 國吉康夫, 大津展之, “画像・単語列間の確率的な概念獲得による高速かつ高精度な汎用的画像認識・検索手法,” 情報処理学会第 70 回全国大会, pp.5-87 - 5-88, 筑波, Mar., 2008.

8. 中山英樹, 原田達也, 國吉康夫, 大津展之, “画像・単語間概念対応の確率構造学習を利用した超高速画像認識・検索方法,” 電子情報通信学会技術研究報告, PRMU2007-147, pp.65-70, 神戸, Dec., 2007.

9. 松本理恵, 中山英樹, 原田達也, 國吉康夫, “ジャーナリストロボット: 実世界からニュース記事を生成するロボットシステム,” ロボティクス・メカトロニクス講演会 2007 (ROBOMECH), 2A1-L02, 秋田, May, 2007.

10. 下畠康幸, 中山英樹, 原田達也, 大津展之, “カメラの動きに頑健な異常検出手法に基づく移動物体の検出,” ロボティクス・メカトロニクス講演会 2007 (ROBOMECH), 2P1-C08, 秋田, May, 2007.

Others

1. **(Competition)** Tatsuya Harada, Hideki Nakayama, Yoshitaka Ushiku, Yuya Yamashita, Jun Imura, and Yasuo Kuniyoshi, Got the 3rd place in the ImageNet Large Scale Visual Recognition Challenge 2010 (in conjunction with ECCV 2010), Crete, Greece, Sep., 2010.

2. **(Invited talk)** 中山英樹, “実世界指向画像認識・検索手法の開発とその応用,” 第 8 回情報科学技術フォーラム（FIT 2008）イベント企画：次世代を担う若い情報・システム研究者を迎えて, pp.11-12, 仙台, Sep., 2009.

Awards

1. 2008 年　　日本機械学会三浦賞

2. 2008 年　　PRMU 研究奨励賞

3. 2008 年　　計測自動制御学会 SI 部門賞若手奨励賞

4. 2009 年　　情報処理学会第 70 回全国大会大会奨励賞

5. 2009 年　　MIRU 2009 シングルトラックオーラルセッション採択

Patents

1. 特徴量生成装置, 特徴量生成法および特徴量生成プログラム, ならびにクラス判別装置, クラス判別方法およびクラス判別プログラム, 特願 2009-121244, 2009.

2. 対応関係学習装置および方法ならびに対応関係学習用プログラム, アノテーション装置および方法ならびにアノテーション用プログラム, および, リトリーバル装置および方法ならびにリトリーバル用プログラム, 特願 2007-240272, 2007.