Building Effective and Scalable Visual Object Recognition Systems

Qiang Chen

A THESIS SUBMITTED FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

Department of Electrical and Computer Engineering

National University of Singapore

2013

Declaration

I hereby declare that this thesis is my original work and it has been written by me in its entirety.

I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.

Qiang Chen

May. 2013

Summary

Visual object recognition is of fundamental importance to artificial intelligence. In this thesis, we aim to build the most effective general object recognition system on well-known benchmarks, e.g. PASCAL VOC. Furthermore, we successfully scale this system into a large scale setting with much less complexity compared with other works.

This thesis addresses a number of key issues that are needed to build a working system. At the feature representation part, we first introduce the SuperCoding which extends the GMM-based coding to the second order statistic while remaining the favourable linearity. Based on the coded features, we perform the object-centric pooling by means of the proposed Generalized Hierarchical Matching (GHM) with useful side information. At the model learning part, we consider the high level task context from the object detection and classification tasks. We develop a novel mutual and iterative contextualization scheme for both tasks based on the so-called Contextualized Support Vector Machine (Context-SVM) method. Extensive experiments show the effectiveness of these novel methods.

Furthermore, we scale this effective system to the large scale setting with thousands of categories and millions of images. By means of efficient Pointwise Fisher Vector coding, per-pixel pooling and the context modelling, our experiments show that the proposed system can perform detection of 1000 object classes in less than one minute on the ImageNet ILSVRC2012 dataset using a single CPU, while achieving comparable performance to state-of-the-art algorithms.

To sum up, by utilizing several novel keys, we build an effective visual object recognition system demonstrated on benchmarks and propose a scalable solution for large scale object recognition problem.

Acknowledgement

There are many people whom I wish to thank for the help and support they have given me throughout my Ph.D. study. My foremost thank goes to my supervisor Dr. Shuicheng Yan. I thank him for all the guidance, advice and support he has given me during my Ph.D. study at NUS. For the last four years, I have been inspired by his vision and passion to research, his attention and curiosity to details, his dedication to the profession, his intense commitment to his work, and his humble and respectful personality. During this most important period in my career, I thoroughly enjoyed working with him, and what I have learned from him will benefit me for my whole life.

I also would like to give my thanks to Dr. Zheng Song, Mr. Zhongyang Huang, Mr. Yang Hua for all their kind help throughout during the PASCAL VOC challenges. The time we fight together is the most precious moment. I would also like to take this opportunity to thank all the students and staffs in Learning and Vision Group. During my Ph.D. study in NUS, I enjoyed all the vivid discussions we had and had lots of fun being a member of this fantastic group.

Last but not least, I would like to thank my parents for always being there when I needed them most, and for supporting me through all these years. I would especially like to thank my wife Shen Shiqun, who with her unwavering support, patience, and love has helped me to achieve this goal.

This dissertation is dedicated to my newborn baby Chen Junyang and my wife Shen Shiqun.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Artificial intelligence (AI) is the intelligence of machines and robots and the branch of computer science that aims to create the intelligence. AI research is highly technical and specialized, divided into subfields that often fail to communicate with each other. The central problems of AI are found in traits as reasoning, knowledge, planning, learning, communication, perception and the ability to move and manipulate objects. Of all these traits, perception is the ability to use input from sensors (such as cameras, microphones, sonar and others more exotic) to deduce aspects of the world. One key function of perception is visual recognition that help the robots/machines to see the world and to understand the world using the visual clues. This thesis focuses on different aspect of visual recognition, especially the problems of visual object recognition.

In the last decade, visual recognition or visual object recognition, has raised a lot of attention in both academia and industry. It is the ability to perceive an object's physical properties (such as shape, colour and texture) and apply semantic attributes to the object, which includes the understanding of its use, previous experience with the object and how it relates to others. Visual object recognition has a lot of applications. For example, in intelligent surveillance system, object detection technique including pedestrian detection and car detection helps to identify the

particular object of interest and object attribute detection helps to further assist humans to localize and search for specific persons or objects. In online social network, face recognition techniques are popular since it provides accessibility for users to annotate and recognize people. More importantly, with the increasing number of digital images, the need for visual object recognition are more and more demanding. This thesis focuses on the general problems of visual object recognition that are to predict/localize any object in the image/video. The techniques discussed can be used in most of the applications. Among them, we are especially interested in some key topics which show promising improvement over the traditional techniques in the past decades. Firstly, to enable effective object classification and detection, sophistic feature encoding, feature pooling and context modelling is needed. Feature encoding and pooling helps to extract more meaningful and robust information from the low level noise feature. Context modelling can be utilized for the discrimination of the ambiguous samples. Furthermore, the large scale visual recognition also attracted a lot of attention recently. The large scale problem often refers to the large scale of categories, the large amount of data. Efficient solution is required to meet these problems.

This thesis focuses on the sub problem of artificial intelligence which is visual object recognition. Objects recognition is the basic level of human real world understanding and building an effective and scalable visual object recognition system for machine is one important building block of artificial intelligence. There is a lot of research work on this problem. This thesis reports on the pioneering work during the year 2010-2012.

1.1 Background and Related works

With the increasing number of digital images, the need for visual recognition is getting greater and greater demanding. Classifying images into semantic categories(e.g.

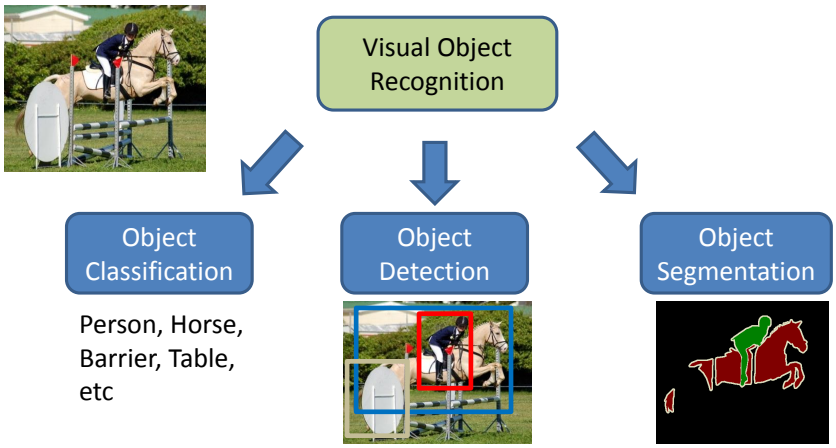

Figure 1.1: Standard visual object recognition tasks: object classification, object detection and object segmentation.

coast, mountains, streets) and also classifying its semantic objects(e.g. motorbikes, sky, planes, faces) is a challenging and important problem nowadays. In visual recognition research, there are several main challenges including *view point variation*,*illumination changes*,*intra-class variation*, *occlusion* and *scale*. All these facts make this problem very challenging.

There are several typical tasks defined for visual object recognition as show in Figure 1.1: (1) **Object Classification** which aims to predict the existence of certain objects in the images, (2)**Object Detection** which targets to predict and localize the objects in the images, and (3) **Object Segmentation** which tries to obtain the per-pixel object level indication masks for the images. Although these tasks seems diverse, the ultimate target (visual recognition) is the same but at different levels, i.e. at the whole-image level - object classification, at the sub-window level - object detection, and at the pixel level - object segmentation.

Due to their intrinsic consistency among these tasks, the standard visual object recognition pipeline shares the most of the parts for different tasks as shown in Figure 1.2. Traditionally, the most practical pattern recognition systems are composed of multiple modules, e.g. feature representation, model learning, context model-

ing [2]. For the feature representation part, the standard main components are these steps: (1) low level feature extraction which extracts meaningful features from the raw image space. Different low level features are often extracted, e.g. Histogram of Gradients (HoG) [3], SIFT [4], Local Binary Pattern(LBP) [5]. (2) feature coding which encodes these low level feature to a predefined model, e.g. Bag of Word [6]. The recent coding schemes can be divided into Vector Quantization (VQ) based, Sparse Coding (SC) based and Gaussian Mixture Models based (GMM). (3) feature pooling which pools these encoded features over sub-clusters through different side information, e.g. the spatial, i,e. Spatial Pyramid Matching (SPM) [7] or feature space domain, i.e. Pyramid Matching Kernel (PMK) [8]. Beyond various modelling (classifier learning) methods, the usage of context has become more and more popular for enhancing the algorithmic performance. Many recent studies have demonstrated considerable improvement for object detection and classification by using external information, which is independently retrieved and complementary with traditional image descriptors. These contexts have been proved useful object recognition tasks [9] [10].

All those integral parts serve as the core of visual object recognition system for different tasks. There are also some specific techniques for different tasks, e.g. the structural learning and hypothesis search for object detection and segmentation tasks which are beyond the discussion of this thesis.

This thesis focuses on the recent progress on core parts of visual object recognition, i.e. the feature coding and feature pooling part at the feature representation part, and the context modeling at the model learning stage. Furthermore, only recently efficient solution of object recognition has attracted increasing attention due to the practical need, the thesis is also interested to discuss these solutions which make the visual system more scalable. The next subsection will first review the related work on Feature Encoding, Feature Pooling and Context Modelling followed by a comprehensive review on Efficient Object Recognition.

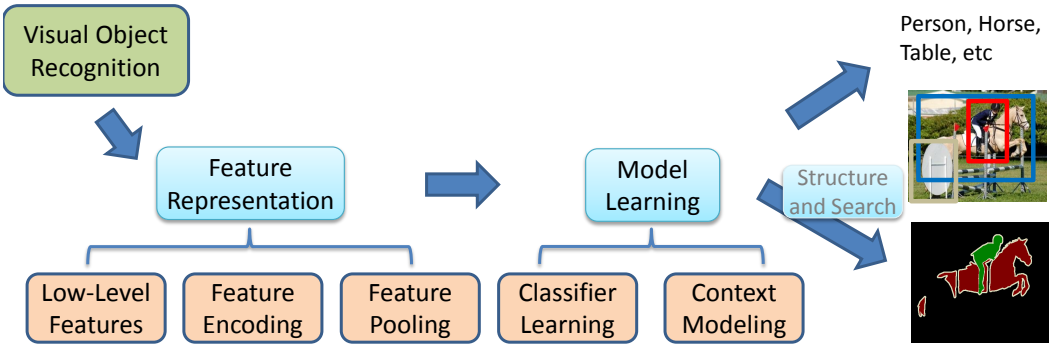

Figure 1.2: Visual object recognition pipeline.

1.1.1 Feature Encoding

We define the term **Feature Encoding** as a process that adapt a set of low level features to a existing model and thus obtain a comparable representation between different sets. For example, the traditional BoWs model adapts the set of low level features, e.g. SIFT, to a predefined visual dictionary. Thus the obtained visual histogram can be used as comparable representation for different images. Recent feature encoding approaches, such as Sparse Coding [11] and Locality-constrained Linear Coding(LLC) [12], introduced soft assignment for local feature quantization to substitute previous discrete quantization methods and can be seen as the gentle extension of Vector Quantization. For the recognition problem, these two coding methods benefit from large size codebooks as demonstrated in a recent comparison survey [13]. The large codebook size and the introduction of soft assignment narrow down the quantization error but also bring a lot of computation cost. Lately, the aggregation coding, e.g. Fisher Vector coding or Super Vector coding, demonstrated to greatly improve the discriminative power of local features [13]. Fisher encoding [14] tries to capture the average first and second order differences between local features and centres of a Mixture Gaussian Distributions learnt from general datasets while Super vector encoding [15] only focuses on the first order difference. Recently, G. Csurka et.al [16] extended Fisher Vector coding to patch level for the

semantic segmentation task and achieves good performance.

1.1.2 Feature Pooling

We define the term **Feature Pooling** as the process that select subsets of the encoded features and get the pooled feature over these subsets. For example, for Sparse Coding based methods [11], the "Max Pooling" is often used to select the max response from the pool of encoded feature and use this as the representation. However, the "Pooling" process is not only restricted to this simple logic operation. The underlining nature of the defined "Feature Pooling" is to select subsets according to some rules so that these subsets have more comparable meaning. For example, the spatial pooling which is widely used in image classification, i.e. Spatial Pyramid Matching [7], forms the subsets according to the spatial locations. This approximate geometric alignment can better make the pooled feature comparable at different levels.

1.1.3 Context Modelling

Traditionally, the context is often considered as special features. Most of the existing strategies [10][9][17] utilize the context via feature concatenation, model fusion or confidence combination, and take the context as another independent component. However, context may have unstable distribution, and its reliability and noise level are not controllable. Therefore it demands adaptive contextualization with proper constraints from the main task to avoid the inappropriate usage of context information. Harzallah et al. [17] introduced the pioneer work for object detection and classification contextualization through the postprocessing of probability combination. The mutual contextualization shows promising performance improvement. However the learning scheme which seamlessly integrates the context information for collaborative learning is missing.

1.1.4 Efficient Object Detection

Recent shape-based object detection methods rely on discriminative shape templates using orientation histograms of image gradients. Initially, Dalal and Triggs [3] used a single rigid template to build a detection model for pedestrians. Thereafter, the PASCAL VOC dataset [18] was released, comprising objects with more deformable shapes like animals and vehicles. Hence the single template model was extended to part-based models [19] by Felzenswalb et al. to handle small shape deformations. Although the deep convolution network [20] shows promising result on ImageNet, the part-based model methods [21, 22, 23] are still the best-performing methods on the practical detection datasets. Generally saying, the part-based models benefit from the relaxed template relation by splitting a single rigid model into smaller part models, and each part model can be learnt on a finer level with more shape details of the object. However, because the shape template is sensitive to position, scale, view, etc., each fine part template can only handle a specific kind of object deformation or view change. Hence the complexity becomes intractable if the object deformation is very large. Consequently, such approaches are not suitable for our proposed large-scale object detection problem with unconstrained deformation.

Previous research [24, 25, 26, 27, 28] have also explored the BoW model detection. The MKL object detection [24] which uses kernel-based models and spatial pyramid (SP) feature combination achieves promising results but the computation cost is very high. Efficient Subwindow Search (ESS) [25, 26, 27, 28] tries to speed up the VQ-based BoW model using a branch and bound technique but often with much poorer performance on standard datasets. The main disadvantage of VQ is that it encodes the local feature as one specific visual word index, thus no complex local discriminative model can be build upon this.

The BoW-based model has the advantage of efficiency if one linear model can be applied and the possible theoretical computation cost is much less than the template-based approach. Suppose we use the same low level feature for both models, e.g.

HOG. For a template model with $m \times n$ cells, we need to compute $m \times n$ times convolution at each pixel for each category test searching over the image. The search complexity is $\mathcal{O}(mnP)$ where P is the searching space complexity for an image. For a BoW model, the cost is separated into two parts, i.e. the local feature coding step and inference (dot-product) over the linear model. The cost of local feature coding step often increases with the codebook size K which is independent for each categories. For multi-class object detection, the only cost addition is the inference cost which depends on the sparseness $\mathcal{E}$ of the coding. The sparseness is $1/K$ for hard Vector Quantization (VQ),and is around 3% for Fisher Vector coding (FV) [14] in our experiments. So the inference complexity is $\mathcal{O}(\mathcal{E}P)$ which is much less than the template-based approach ($mn \gg \mathcal{E}$).

1.2 Thesis Focus and Main Contributions

The recognition system follows the pipeline of feature extraction, feature encoding, feature pooling and model learning. In this thesis, we focus on the later three parts of the pipeline. The main motivations and gaps are as follows:

1. For the feature encoding part, feature encoding has attracted numerous attentions in recent object recognition works. Among those work, the GMM-based approaches achieved the most significant result, e.g. the SuperVector [15] and FisherKernel [14]. However the underlining theoretical analysis is missing.

2. At the feature pooling part, Bag of Words (BoWs) and spatial pyramid matching (SPM) are often used. The popular SPM has been used as the common technique used in object recognition at the feature pooling stage. This method has demonstrated effective for image classification. However, the object-centric task requires object-oriented pooling instead of this weakly spatial pooling.

3. Previously, there are some of the work that focused on the context model learning in terms of object co-occurrence, object size and spatial layout. How-

ever, the significance of mutual context model between object classification and detection has been underestimated.

In this thesis, the demonstrated most effective object recognition system on PASCAL VOC has been presented. Furthermore, we successfully scale this system into a large scale setting with much less complexity compared with other works. More specifically, we conduct research on the following aspects:

1. Recent Advance of Feature Encoding. We give qualitative analysis to explain the question that why these feature encoding methods work well. Based on the analysis, we re-introduce the generative version GMM modelling, called Super-Coding. SuperCoding extends the previous Universal Background Modelling into the second order and it well fits the current encoding framework.

2. Generalized Hierarchical Matching/Pooling with Side Information. To better serve the object-centred problem, we propose the Generalized Hierarchical Matching (GHM) approach which is more suitable for object-centred recognition while SPM is optimized for scene recognition. Each image is expressed as a bag of orderless pairs, each of which includes a local feature vector encoded over a visual dictionary, and its corresponding side information from priors or contexts. The side information is used for hierarchical clustering of the encoded local features. Then a so-called hierarchical matching kernel is derived as the weighted sum of the similarities over the encoded features pooled within clusters at different levels.

3. Contextualized Object Classification and Detection. To further enhance the robustness of the context model, we develop a novel mutual contextualization scheme for object detection and classification based on the so-called Contextualized Support Vector Machine (Context-SVM) method. Extensive experiments show that Context-SVM can efficiently learn the context models under

various conditions and effectively utilize context information for performance boosting.

4. Efficient Maximum Appearance Model for Large Scale Object Detection. Furthermore, we consider the problem of large scale object recognition. We represent the image as an ensemble of densely sampled feature points with the proposed Pointwise Fisher Vector encoding. The learnt discriminative model can be applied to the enriched local representation unlike the state-of-the-art template-based model in which the learned model has to be applied to each testing window exhaustively. Consequently the object detection problem is transformed into searching an image sub-area with maximum local appearance probability. The overall complexity of the proposed framework is much less than the traditional template-based detection methods. The advantage of low computation complexity enables us to explore the large scale object detection problem with huge number of categories.

Each of these works serves as one piece of our visual object recognition system towards the effectiveness and efficiency. There are many other works for visual object recognition in the literature for the past years. It is inevitable that this thesis has bias towards the general problems instead of specific application, e.g. human detection or face recognition techniques. In all the work of the whole thesis, the only used label information are the object bounding box and object existence label. Other popular information, e.g. object masks or object attributes are not utilized.

1.3 Organization of this thesis

In Chapter 2, the related benchmark datasets proposed in recent years are introduced followed by the recent advance analysis of feature encoding in Chapter 3. Then in Chapter 4, we propose the Generalized Hierarchical Matching representa-

tion for image classification, the relation of localized model and global model as context is introduced in Chapter 5. Based on the aforementioned techniques, we further scale the framework in a large scale setting in Chapter 6.

Chapter 2

Datasets and Benchmarks

The task of visual object recognition research needs large amount of annotated data. In the past decade, researchers have provided a lot of well-organized datasets along with different research tasks. The object classification task only needs the proper image labels. The object detection task requires the bounding box annotation along with the label information. For object segmentation task, the per pixel level annotation is often needed. In this chapter, we introduce the relevant datasets from the historic development view.

2.1 The Start

A number of well labeled small datasets (Caltech101/256, MSRC, PASCAL VOC, etc.) have served as training and evaluation benchmarks for most of todays computer vision algorithms. As computer vision research advances, larger and more challenging datasets are needed for the next generation of algorithms.

The first well known object recognition/image classification dataset is the Caltech 101 [29] dataset which was collected by choosing a set of object categories, downloading examples from Google Images and then manually screening out all images that did not fit the category. In the late years, Caltech-256 [30] was collected

in a similar manner with several improvements: a) the number of categories is more than doubled, b) the minimum number of images in any category is increased from 31 to 80, c) artifacts due to image rotation are avoided and d) a new and larger clutter category is introduced for testing background rejection.

In 2006, the PASCAL VOC [18] datasets start to release along with the well-known challenges, i.e. PASCAL VOC Challenges. The main goal of this series of challenges/datasets are to recognize objects from a number of visual object classes in realistic scenes (i.e. not pre-segmented objects). It has been updated yearly since 2006. The images of VOC is obtained from Flickr and manually labeled and close to realistic scenes. 20 classes ranging from outdoor objects to indoor objects are annotated with detection window and a part of them are segmented. Currently, the train/val data of VOC 2011/2012 has 11,530 images containing 27,450 ROI annotated objects and 6,929 segmentations.

Lotus Hill is another general purpose image database with human annotated ground truth. Three levels of information are labeled, i.e. scene level (global geometric description), object level (segmentation, sketch representation, hierarchical decomposition), and low-mid level (2.1D layered representation, object boundary attributes, curve completion, etc.). The database consists of more than 636,748 annotated images and video frames. However, due to its non-academic nature, few researchers reported results on this datasets.

There also exist some other datasets which were collected by researchers available for different research purpose, e.g. MSRC [1] for segmentation, MIRFlickr dataset [31] [2] for image classification and retrieval, 15 scenes [7] [3] for scene understanding and image classification, etc.

[1] http://research.microsoft.com/en-us/projects/ObjectClassRecognition/
[2] http://press.liacs.nl/mirflickr/
[3] http://www-cvr.ai.uiuc.edu/ponce_grp/data/

Name	# of Images	# of Classes	Annotation Level
Caltech101 (2004)	9,146	101	Cls
Caltech256 (2007)	30,607	256	Cls
TinyImages (2008)	79,302,017	53,464	no.
PASCAL VOC (2012)	11,530 +12300	20	Cls, Det, Seg
Lotus Hill (2007)	636,748	13 subsets	Seg, 3D
LabelMe (2007)	187,240		Cls, Seg
ImageNet(2009)	14,197,122+	21841+	Cls, Det

Table 2.1: Some statistical data of different datasets

2.2 Large Scale Datasets

Only in recent years, the explosion of image data on the Internet has the potential to foster more sophisticated and robust models and algorithms to index, retrieve, organize and interact with images and multimedia data. Now, people are trying to grasping the metric of Internet and construct large scale datasets with finer annotation.

TinyImages [32] can be thought as the beginner of large scale dataset from Internet. It is a dataset of 80 million 32x32 low resolution images, collected from the Internet by sending all words in WordNet as queries to image search engines. Each synset in the TinyImage dataset contains an average of 1000 images, among which 10-25% are possibly clean images. Although the TinyImage dataset has had success with certain applications, the high level of noise and low resolution images make it less suitable for general purpose algorithm development, training, and evaluation.

Furthermore, Internet contributes at more aspect for the large scale dataset. (1) LabelMe [33] an online annotation tool to build image databases for computer vision research. (2) ImageNet [34] an image database organized according to the WordNet hierarchy (currently only the nouns), in which each node of the hierarchy is depicted by hundreds and thousands of images.

LabelMe provides a general tool for labeling images with deep information. For each concept, e.g. scene, objects, it provides the polygon annotation so that it

enables more high level content analysis, e.g object detection and segmentation. In 2011, SUN [4], is organized as two parts, i.e. scene recognition and object recognition.

ImageNet [34] is first introduced in 2009. It is an image database organized according to the WordNet hierarchy (currently only the nouns), in which each node of the hierarchy is depicted by hundreds and thousands of images.

the usage of Amazon Mechanical Turk(AMT): To collect a highly accurate dataset, it needs a lot of human labours to verify each candidate image collected in the previous step for a given synset. This is usually achieved by using the service of Amazon Mechanical Turk (AMT), an online platform on which one can put up tasks for users to complete and to get paid. AMT has been used for labeling vision data [35]. With a global user base, AMT is particularly suitable for large scale labeling.

2.3 Challenges

Along with the rise of large scale datasets, some challenging contests are being held. TRECVID [36]: The main goal of the TREC Video Retrieval Evaluation (TRECVID) is to promote progress in content-based analysis of and retrieval from digital video via open, metrics-based evaluation. TRECVID is a laboratory-style evaluation that attempts to model real world situations or significant component tasks involved in such situations. It includes several datasets, each of which contains huge amount of data, e.g. IACC.1.B and IACC.1.A each contains 8000 internet videos (about 200 hours each), the MED task uses 4000 hours of multimedia clips.

The ImageNet Large Scale Visual Recognition Challenge starts from 2010 and continues to be held yearly (ILSVRC2010, 2011, 2012 and 2013). ILSVRC [34] is now a benchmark challenge for large scale object recognition. It starts from 2010 and at each year about 1 million images and 1000 categories data will be released.

[4] `http://groups.csail.mit.edu/vision/SUN/`

In recent two years, several taster challenges also were included, e.g. the large scale object detection task and fine grained object recognition.

2.4 In the future

The real world human recognition ability is surely much beyond the current academic definition. Object classification, detection and segmentation are three separate tasks defined by the academical world. More interesting and important tasks have been proposed with the deeper understanding of visual object recognition system. With the help of the rich data content brought by the Internet, these tasks are becoming possible and these directions have attracted increasing attentions.

- Fine-grained visual recognition. Fine-grained visual recognition aims to recognize fine detailed categories for certain objects. It extents the basic level recognition to a deeper and finer level. For example, one typical fine grained recognition target proposed in ImageNet ILSVRC 2012 [34] is to recognition the dog species from 200 kinds of dogs. Other datasets are also built up to achieve the target of fine-grained visual recognition, e.g. the CUB-Birds containing 200 kinds of birds with bounding box and detailed parts annotation [37]. Another interesting fine-grained recognition system is [111] in which it describes a working computer vision system that aids in the identification of plant species.

- High-level visual annotation/recognition. The ultimate goal of visual recognition is to "understand" the images. Some tasks of visual recognition is higher level than the current category-level understanding, e.g. human action recognition which tries to answer the questions of "what is/are the person doing in the image?". This task is interesting and worth exploring. Some of the pioneering works have been conducted with some preliminary datasets, e.g. the Action Recognition dataset in VOC [18], the People Playing Musical

Instrument (PPMI) dataset [38].

Chapter 3

SuperCoding: High Order Parametric Coding for Visual Recognition

We define the term **Feature Encoding** as a process that adapt a set of low level features to a existing model and thus obtain a comparable representation between different sets. For example, the traditional BoWs model adapts the set of low level features, e.g. SIFT, to a predefined visual dictionary. Thus the obtained visual histogram can be used as comparable representation for different images.

Recently, feature encoding has attracted numerous attention for visual recognition work. Among those work, the distribution-based approaches which depict the images as a distribution over predefined generative model, e.g. the MeanVector [49] and FisherVector [14], achieved the most significant results over the other encoding methods on the standard datasets, e.g. PASCAL VOC [18] and ImageNet [34]. In this work, we first give comprehensive and qualitative analysis on the various distribution-based approaches. Based on the analysis, we introduce the parametric coding, the so-called SuperCoding, where the codes consist of parameters from the

adapted model with the high order statistics. A linear kernel can be obtained for the corresponding KL divergence distance measurement. Thus efficient training and testing can be achieved with the linear representation. We also propose several improvement which promotes the performance and verified by extensive experiments. Further more, we show that the proposed coding method can be generalized to various recognition tasks with formal spatial modeling, e.g. object classification and scene recognition etc. Extensive experiments on these tasks shows the advance of the proposed encoding method.

3.1 Introduction

Visual recognition is one key task of artificial intelligence. The performance of visual recognition highly relies on the construction of representation and the metric learning defined upon this. Representation can be roughly divided into several levels according to its semantic meaning: (1) Low level feature describes certain aspect of one local image patch, e.g. SIFT for edge, Color Moment for color. (2) Middle level feature representation often merge a sets of low level feature from image (e.g. the whole image). (3) high level representation often refers to those work with high level semantic meaning, e.g. attribute, meta information. We focus on the problem of middle level feature learning.

The term **Feature Encoding**[1] can be considered as a process that adapts a set of low level features to a existing model and thus obtain a comparable representation between different sets. There are several main streams of visual representation in the literatures: (1) Template feature which is the naive concatenate of local features. The underlying model is the spatial grid which restricts the way of local feature concatenation. (2) Reconstruction-based representation aims to reconstruct each local feature with a dictionary model with minimum error and pools the reconstruction

[1]The term "feature encoding" and "feature coding" has the same meaning in this paper. We will use them indiscriminately.

coefficients as the representation, e.g. Vector Quantization, Sparse Coding, etc. (3) Distribution-based representation considers each image patch is generated through a probabilistic model, e.g. GMM. The combination of all the local features forms the adapted distribution of model characterized by statistics, e.g. Fisher Kernel [14], Mean Vector [49], Super Vector [15]. In recent years, a large number of novel feature encodings methods, for image analysis have been proposed. Performance are promoted by simply replacing with new image representation. For example, by using vocabulary tree instead of the flat BoW for image retrieval, the speed and accuracy are both enhanced in UKBench datasets. For image classification, the accuracy is boosted from 15% to 34% by replacing the VQ with the sparse coding representation on caltech 256 with the similar learning scheme.

Among all these encoding methods, various studies shows that the distribution-based methods achieved most success on different datasets for object classification task. The distribution-based methods assume the generation of image come from a probabilistic model, thus measuring the distance of two images is equal to the measurement upon the image model. There are two categories between these coding methods. One is the parametric representation, where the feature codes consist of parameters from the utterance-dependent or adapted model, e.g. the MeanVector [49], and the other is the derivative representation, where the derivatives of the loglikelihood with respect to parameters of a generative model are used, e.g. FisherVector [14] and SuperVector [15].

Although there are a lot of work focusing improving the recognition task by means of proposing new coding method, the underlying analysis about the questions about distribution-based methods: "what's the difference" and "why this works" is missing. In this work, we first give qualitative analysis on those distribution-based approaches under the same GMM adaption framework. This analysis directly points out the missing component of current parametric representation. Thus, we propose the SuperCoding with high order statistics after the model adaption. We also pro-

pose several key ingredients for the SuperCoding which promote the performance greatly. We further demonstrate that the proposed SuperCoding, also works for various recognition task, e.g. face age/gender estimation, object recognition and scene classification. Surprisingly good results have been obtained when combined with appropriate spatial modeling techniques.

In the following sections, we first introduce some related works in Section 5.2. We give qualitative analysis on the difference of the distribution-based approaches in Section 3.3. The SuperCoding is introduced in Section 3.4.

3.2 Overview of Recent Coding Schemes

In recent years, huge improvement has been made for the visual recognition research. The most important part among them is progress of the image representation. Performance are promoted by simply replacing with new image representation. For example, by using vocabulary tree instead of the flat BoW for image retrieval, the speed and accuracy are both enhanced in UKBench datasets. For image classification, the accuracy is boosted from 15% to 34% by replacing the BoW with the sparse coding representation on caltech 256 with the similar learning scheme.

We consider the image I consisting of N patches $\{p_i, i \in 1 \cdots, N\}$, feature encoding step aims to assign each local patch to a predefined model/codebook C and generate corresponding codes for further high level tasks. This procedure can be abstracted as follows:

$$Codes(I) = \phi(\Omega(p_1, C), \cdots, \Omega(p_N, C)), \tag{3.1}$$

where Ω is the assignment function for each local patch with predefined codebook C, ϕ is the codes generation function for the patch sets.

3.2.1 BoW and its extension to large scale

The key idea of BoW is that using N local descriptor describing the image to form a unique vector. The sparse vectors often brings efficient comparison and it inherits invariance of the local descriptors. The BoW aims to find the following assignment values:

$$\Omega_{bow}(p, C) : \arg\min_i ||p - C_i||_2; \tag{3.2}$$

Then the codes of the BoW is the average of the assignment/voting for all the local patches. BoW achieved great success for many visual recognition tasks, e.g. image classification, image retrieval, etc. With the development of BoW, researchers found there are two main problems: **Codebook size**. BoW is often favorable to use large codebook for image retrieval and image classification. For large scale setting, the cost for assigning each local features is very sensetive. To solve this problem, hierarchical KD-tree or hashing is often used to reduce to comparison cost [45]. **Quantization**. Another problem of BoW is the quantization error brought by the modeling. Many coding methods built upon this is trying to minimize the quantization error of the local feature, e.g. [12] [11] [40].

3.2.2 Reconstruction-based Encoding

As discussed above, one problem of the original BoW coding is the quantization error. Sparse coding methods, e.g. ScSPM [11], Locality-constrainted Linear Coding [12], aims to find the assignment function with the sparsity constraints:

$$\Omega_{sc}(p, C) : \quad \arg\min \quad ||p - v_i C_i||_2 + \lambda ||v||_1; \tag{3.3}$$

To improve the scalability, researchers aim at obtaining nonlinear feature representations that work better with linear classifies, e.g. [11, 41]. In particular, Yang et

al. [11] proposed the ScSPM method where sparse coding (SC) was used instead of VQ to obtain nonlinear codes. The method achieved state-of-the-art performances on several benchmarks. Yu et al. [41] empirically observed that SC results tend to be local nonzero coefficients are often assigned to bases nearby to the encoded data. They suggested a modification to SC, called Local Coordinate Coding (LCC), which explicitly encourages the coding to be local, and theoretically pointed out that under certain assumptions locality is more essential than sparsity, for successful nonlinear function learning using the obtained codes. Similar to SC, LCC requires to solve L1-norm optimization problem, which is however computationally expensive. To further reduce the computation cost, [12] presented a practical coding scheme called Locality-constrained Linear Coding (LLC), which can be seem as a fast implementation of LCC that utilizes the locality constraint to project each descriptor into its local-coordinate system.

3.2.3 Distribution-based encoding

Another line of recent image representation is distribution-based encoding. Most of these works used the Gaussian Mixture Models (GMM) for describing the data distribution except the original Vector Quantization coding which only records the histogram of model statistics. It has been demonstrated that the higher order statistics achieved much better result than the Vector Quantization approaches. Thus in the following sections, we only focuses on the GMM-based encoding methods which include a lot of diverse techniques. But they have one same part that is the GMM modeling of the data and further use its mixture model parameters. The assignment function of these approaches often refer to calculate the posterior of the patch p belonging to mixture k of GMM C:

$$\Omega_{gmm}(p, C) : \gamma_{p,k}; \tag{3.4}$$

Table 3.1: Summary of different GMM-based coding methods.

Methods	Coding	Rep	Order	L/NonL
SuperVector	$s_k = s\sqrt{p_k}$, $p_k = \frac{1}{N}\sum_{i=1}^{N}\gamma_{ik}$, $u_k = \frac{1}{\sqrt{p_k}}\sum_{i=1}^{N}\gamma_{ik}(x_i - \mu_k)$	$[s_1, u_1, \cdots, s_K, u_K]$	1st	L
MeanVector	$u_k = \sqrt{\pi_k}\frac{\hat{\mu}_k}{\sigma_k}$	$[u_1, u_2, \cdots, u_K]$	1st	L/NL
FisherVector	$u_k = \sum_{i=1}^{N}\frac{1}{N\sqrt{\pi_k}}\gamma_{ik}\frac{x_i-\mu_k}{\sigma_k}$, $v_k = \sum_{i=1}^{N}\frac{1}{N\sqrt{2\pi_k}}\gamma_{ik}[\frac{(x_i-\mu_k)^2}{\sigma_k^2}-1]$	$[u_1, v_1, \cdots, u_K, v_K]$	1st&2nd	L
SuperCoding	$u_k = \frac{\hat{\mu}_k}{\sqrt{\sigma_k}}, v_k = \frac{\hat{\sigma_k}}{\sigma_k}$	$[u_1, v_1, \cdots, u_K, v_K]$	1st&2nd	L

The difference of these encoding function lies on the coding generation function ϕ. The GMM Meanvector [49] takes the adapted mean vector of the GMM as the representation. The SuperVector [15] includes another soft assignments term as the GMM histogram other than the mean vector term. The Fisher Kernel coding [14] and its improved version incorporates higher order of statistics, e.g. first order and second gradient, and show great performance improvement over traditional BoW representation. Some of these works have been evaluated in a recent comparison paper [13] and advantages can be seen when compared with other coding methods. We give detailed comparison in the following section and point out the missing components.

3.3 GMM-based Coding for Visual Recognition

We summary the different GMM-based coding methods for visual recognition and their codes generation functions. All those methods incorporating the posterior calculation. Given a GMM model $u_\lambda(x) = \sum_{i=1}^{K}\omega_i u_i(x)$, for a set of low level features $X = \{x_1, \cdots, x_N\}$ extracted from a image y, the soft assignments of the descriptor x_i to the kth Gaussian components γ_{ik} is computed by:

$$\gamma_{ik} = \frac{\pi_k u_k(x_i)}{\sum_{k=1}^{K}\pi_k u_k(x_i)} \tag{3.5}$$

3.3.1 Parametric and Derivative Coding

There are two categories between these coding methods. One is the parametric representation, where the feature codes consist of parameters from the utterance-dependent or adapted model, e.g. the MeanVector [49], and the other is the derivative representation, where the derivatives of the loglikelihood with respect to parameters of a generative model are used, e.g. FisherVector [14] and SuperVector [15].

Parametric Coding

GMM Meanvector: In [49], the author proposed to use the mean vector of adapted GMM models using MAP (maximum a posterior). The idea is to measure the distance between two images using the distance of two adapted GMM models. The mean vector is used as the representation. The distance has the following forms:

$$d(I_a, I_b) = \frac{1}{2}\sum_{k=1}^{K} \pi_k (\mu_k^a - \mu_k^b)^T \sigma_k^{-1} (\mu_k^a - \mu_k^b), \tag{3.6}$$

In [49], a conventional Gaussian kernel is defined as

$$k(I_a, I_b) = exp^{\frac{-d(I_a, I_b)}{\delta^2}}, \tag{3.7}$$

which can be considered as a conventional Guassian kernel defined on the so-called MeanVector,

$$\phi(x_a) = [\sqrt{\frac{\pi_1}{2}}\sigma_1^{-\frac{1}{2}}\mu_1^a, \cdots, \sqrt{\frac{\pi_K}{2}}\sigma_K^{-\frac{1}{2}}\mu_K^a], \tag{3.8}$$

We can also define the linear kernel from the distance metric with the form:

$$k(I_a, I_b)_{lin} = \sum_{k=1}^{K} \pi_k \mu_k^a \sigma_k^{-1} \mu_k^b, \tag{3.9}$$

$$= \sum_{k=1}^{K} (\sqrt{\pi_k}\sigma_k^{-\frac{1}{2}}\mu_k^a)^t (\sqrt{\pi_k}\sigma_k^{-\frac{1}{2}}\mu_k^a). \tag{3.10}$$

The corresponding linear vector has the same representation as Eqn.(3.8) which means that Mean Vector is a natural linear representation.

Derivative Coding

Super Vector: In [15], the authors provide two variants of feature coding, based on hard assignment to the nearest codeword or soft assignment to several near neighbours. For the hard super vector encoding, let $\gamma_{i,k} = 1$ if x_i is assigned to cluster k by k-means and 0 otherwise. [15] does not specify how $\gamma_{i,k}$ are set in the soft assignment case. We define the $\gamma_{i,k}$ to be essentially the same as for the GMM coding. As reported in [13], this procedure is reasonable. Thus the obtained SuperVector for X is denoted as $\phi(X) = \{s_1, u_1, \cdots, s_K, u_K\}$ where s_k and u_k is defined as:

$$s_k = s\sqrt{p_k}; \tag{3.11}$$

$$u_k = \frac{1}{\sqrt{p_k}} \sum_{i=1}^{N} \gamma_{ik}(x_i - \mu_k), \tag{3.12}$$

where $p_k = \frac{1}{N}\sum_{i=1}^{N} \gamma_{ik}$, is the mean of soft assignments and s is a constant.

Fisher Vector: In [117], the author proposed the Fisher Kernel for image classification and its corresponding Fisher Vector. In [14], notable improvement has been made to promote the performance of the original method by applying power normalization and spatial pyramid.

For each image X, the Fisher Vector is computed as $\phi(X) = \{u_1, v_1, \cdots, u_K, v_K\}$ where u_k and v_k is defined as:

$$u_k = \sum_{i=1}^{N} \frac{1}{N\sqrt{\pi_k}} \gamma_{ik} \frac{x_i - \mu_k}{\sigma_k}, \tag{3.13}$$

$$v_k = \sum_{i=1}^{N} \frac{1}{N\sqrt{2\pi_k}} \gamma_{ik} [\frac{(x_i - \mu_k)^2}{\sigma_k^2} - 1]. \tag{3.14}$$

while σ_k are square root of the diagonal values of Σ_k. The FV has several good properties: (a) Fisher Vector encoding is not limited to computing visual word occurrence. It also encodes additional the distribution information of the feature points, which will perform more stable when encoding a single feature point. Those high order feature encoding brings exciting performance along with high dimensional feature representation in practical. (b) it can naturally separate the video specific information from the noisy local features(b) we can use linear model for this representation.

Variants of Fisher Vector: VLAD [47] includes first order and VLAT [48] including second order. VLAD and VLAT are simple yet efficient way of aggregating local image descriptors into a vector of limited dimension, which can be viewed as a simplification of the Fisher kernel representation.

3.3.2 Analysis

We give summary of these GMM-based coding with respect to their codes generation function, representation, their order of statistics and whether they are suitable for feeding into linear/nonlinear classifier in their original setting in Table 3.1. There are several observation:

- **Parametric vs. Derivative representations**: We can conclude these GMM-based coding as two categories: the parametric approach,e.g. MeanVector, and derivative approach, e.g. FisherVector and SuperVector. The parametric approach first perform the model parameter adaption and takes the adapted parameters as the feature representation, e.g the MeanVector using the adapted mean values. The derivative approach directly calculates the derivatives of model parameter as the feature representation, e.g. SuperVector used the derivative of mean values of GMM and FisherVector used both the derivatives of mean values and variances.

- **Higher order of statistics brings better performance**: All of these GMM-based coding methods incorporate higher order of statistics than the traditional Vector Quantization-based approaches which only utilized the statistics of histogram. Among those approaches, the literature [13] shows that FisherVector achieves much better performance for visual recognition than SuperVector in a similar setting (same size of GMM). The additional gradients of variance of FisherVector brings the further improvement.

- **Linear Representation**: All of these approaches has the nice property that linear classifier can be operated upon these representations since they have meaningful metrics.

We notice in Table 3.1 that only FisherVector has the second order statistics while the model adaption approaches has this kind of information. One important reason of this missing components is due to the lack of proper metrics for the second order statistics. In the following section, we will show how to construct a relaxed metrics for the second order statistics followed by further several improvement which results a efficient and effective high order GMM-based coding which we call as SuperCoding.

3.4 SuperCoding: High Order Parametric Coding

3.4.1 GMM adaption

Given a GMM model $u_\lambda(x) = \sum_{i=1}^{K} \omega_i u_i(x)$, for a set of low level features $X = \{x_1, \cdots, x_N\}$ extracted from a image y, the soft assignments of the descriptor x_i to the kth Gaussian components γ_{ik} is computed by:

Compute a posteriori:

$$\gamma_{ik} = \frac{\pi_k u_k(x_i)}{\sum_{k=1}^{K} \pi_k u_k(x_i)}, \tag{3.15}$$

$$\tag{3.16}$$

We then compute the sufficient statistics for the weight, mean and variance parameters:

$$Weight : n_k = \sum_{i=1}^{N} \gamma_{ik}; \tag{3.17}$$

$$Mean : E_k(x) = \frac{1}{n_k} \sum_{i=1}^{N} \gamma_{ik} x_i; \tag{3.18}$$

$$Variance : E_k(x^2) = \frac{1}{n_k} \sum_{i=1}^{N} \gamma_{ik} x_i^2; \tag{3.19}$$

Lastly, these new sufient statistics from the training data are used to update the prior sufient statistics for mixture i to create the adapted parameters for mixture i (Figure 2(b)) with the equations:

$$\hat{\pi_k} = [\alpha^{\pi} n_i/N + (1-\alpha^{\pi})\pi_k]\Delta, \tag{3.20}$$

$$\hat{\mu_k} = \alpha^{\mu} E_k(x) + (1-\alpha^{\mu})\mu_k, \tag{3.21}$$

$$\hat{\sigma_k^2} = \alpha^{\sigma} E_k(x^2) + (1-\alpha^{\sigma})(\sigma_k^2 + \mu_k^2) - \hat{\mu_k^2}, \tag{3.22}$$

$$\tag{3.23}$$

3.4.2 SuperCoding

Let's look back the deduction of GMM meanvector: Suppose there exists an Gaussina Mixture Model as the universal background model. Then, from the GMM adaptation process, we can obtain two adapted GMMs for them, denoted as g_a and g_b. Consequently, each image is represented by a specific GMM distribution model, and a natural similarity measure between them is the Kullback-Leibler divergence,

$$D(g_a||g_b) = \int g_a(x) \log(\frac{g_a(x)}{g_b(x)}) dx, \tag{3.24}$$

The Kullback-Leibler divergence itself does not satisfy the conditions for a kernel function, but there exists an upper bound from the log-sum inequality,

$$D(g_a||g_b) \leq \sum_{k=1}^{K} \pi_k D(\mathcal{N}(x_a; \mu_k^a, \Sigma_k^a)||\mathcal{N}(x_b; \mu_k^b, \Sigma_k^b)), \tag{3.25}$$

The symmetric KL divergence is based on Kullback Leibler measure of discriminatory information. Kullback realizes the asymmetry of $D_{KL}(g_a, g_b)$ and describes it as the directed divergence. To achieve symmetry, Kullback defines the divergence as $D_{KL}(g_a, g_b) + D_{KL}(g_b, g_a)$ and notes that it is positive and symmetric but violates the triangle inequality. Hence, it can not define a metric structure. The closed form expression for the symmetric KL divergence between $\mathcal{N}_1$ and $\mathcal{N}_2$ can be written as

$$\begin{aligned} D_{KL}(g_a||g_b) &= \frac{1}{2}\mu^T(\Sigma_a^{-1} + \Sigma_b^{-1})\mu & (3.26) \\ &+ \frac{1}{2}TR(\Sigma_a^{-1}\Sigma_b + \Sigma_b^{-1}\Sigma_a - 2\mathbf{I}). & (3.27) \end{aligned}$$

where $\mu = \mu_a - \mu_b$. We can note that if we assume $\Sigma_a = \Sigma_b = \Sigma$, then $D_{KL_{mean}} = \mu^T\Sigma^{-1}\mu$ which is related to the GMM mean vector. Futhermore, if we assume $\mu_a = \mu_b = \mu$, then D_{KL} expresses the difference, or the dissimilarity between covariance matrices Σ_a and Σ_b.

$$D_{KL_{cov}} = \frac{1}{2}TR(\Sigma_a^{-1}\Sigma_b + \Sigma_b^{-1}\Sigma_a - 2\mathbf{I}); \tag{3.28}$$

It is easy to see that $D_{KL_{mean}}$ is the distance measurement for the GMM mean vector representation. As introduce in [50], we can construct a kernel function $k_{mean}(a, b)$ so that it satisfies the condition that $D(a, b) = k(a, a) + k(b, b) - 2k(a, b)$. Thus intuitively we can find that the linear kernel $k_{mean}(a, b) = \mu_a^T\Sigma^{-1}\mu_b$ satisfies this condition.

The problem is now at the part of $D_{KL_{cov}}$. It is not easy to directly obtain the kernel function from this distance. One possible way to achieve this is to approxi-

mate this KL divergence distance of the covariance part.

$$\begin{aligned} D_{KL_{cov}}(g_a||g_b) &= \frac{1}{2}\sum \frac{\sigma_a^2+\sigma_b^2-2\sigma_a\sigma_b}{\sigma_a\sigma_b} \\ &\approx \frac{1}{2}\sum \frac{\sigma_a^2+\sigma_b^2-2\sigma_a\sigma_b}{\sigma^2} \end{aligned} \tag{3.29}$$

It is easy to obtain that $k_{cov}(a,b) = \sigma_a^T\Sigma^{-2}\sigma_b$ is the kernel function of the approximated distance $D_{KL_{cov}}(g_a||g_b)$. Thus we can construct the combined kernel for the D_{KL} with both the mean vector term and the covariance term. $k(a,b) = \mu_a^T\Sigma^{-1}\mu_b + \sigma_a^T\Sigma^{-2}\sigma_b$. It is desirable to see that the kernel defined is the dot product of the representation: $C_a = [\frac{\mu_1^a}{\sqrt{\sigma_1}};\cdots;\frac{\mu_K^a}{\sqrt{\sigma_K}};\frac{\sigma_1^a}{\sigma_1};\cdots;\frac{\sigma_K^a}{\sigma_K}]$. We call this representation as **SuperCoding**.

3.4.3 Further Improvement

There are several possible improvement which can further improve the representation power.

Residual as representation

There are strong evidence demonstrating that it is better to represent adapted model with its residual instead of its model parameter. Thus we derive the residual representation as the final SuperCoding:

$$C_a = [\frac{\mu_1^a-\mu_1}{\sqrt{\sigma_1}};\cdots;\frac{\mu_K^a-\mu_K}{\sqrt{\sigma_K}};\frac{\sigma_1^a-\sigma_1}{\sigma_1};\cdots;\frac{\sigma_K^a-\sigma_K}{\sigma_K}]; \tag{3.30}$$

In fact, it can be derived that the linear kernel of the modified residual representation can also satisfy the KL distance metric. For examples, let's take the first half representation of C_a as C_{mean}, the obtained linear kernel $\hat{k}_{mean}(a,b) =$

$(\mu_a - \mu)^T \Sigma^{-1} (\mu_b - \mu)$. Then we can obtain the following equation:

$$
\begin{aligned}
& k(a,a) + k(b,b) - 2k(a,b) \qquad (3.31)\\
= \; & (\mu_a - \mu)^T \Sigma^{-1} (\mu_a - \mu) + (\mu_b - \mu)^T \Sigma^{-1} (\mu_b - \mu)\\
- \; & 2(\mu_a - \mu)^T \Sigma^{-1} (\mu_b - \mu),\\
= \; & (\mu_a - \mu_b)^T \Sigma^{-1} (\mu_a - \mu_b),\\
= \; & D_{KL_{mean}}.
\end{aligned}
$$

It shows that the residual offset of the mean value part does not change the linear kernel property - the obtained linear kernel is a valid kernel function of the KL divergence distance. Similar deduction can be used to prove that the second half representation of C_a satisfy the covariance KL distance metric. Thus the overall residual representation can still form a valid linear kernel.

Spatial Modeling

The image has 2D structure and has spatial correlation, however all the coding mentioned above treats each local patch equally and does not consider the spatial information. Thus we have to explicitly model this spatial geometry relation. There are typical two types of spatial modeling for visual recognition proved to be effective. (1) **Spatial Pyramid Matching** (SPM) which partitions the image plane into finer subcell and extract corresponding features within each cell. (2) **Spatial Feature** (SF) which often concatenate the patch coordinates l_i into the raw patch feature x_i with proper normalization, e.g. $[x_i, l_i]$.

These two kinds of spatial modeling approaches has different properties which often is favored by different tasks: (1) For scene/image classification, SPM is often favored since it extracts much larger features that can be feeded into classifier. (2) For face-related application, SF is often utilized due to its low cost and effectiveness for well-structured face problem.

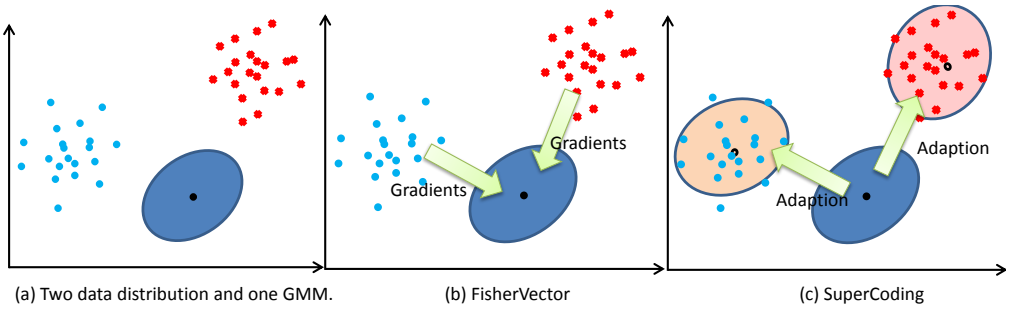

Figure 3.1: The intuition behind FisherVector and SuperCoding. (a) Two data distribution and one GMM model. (b) FisherVector calculates the gradients of the model parameters as the representation. (c) SuperCoding first performs the model adaption and uses the model parameters as the representation.

Normalization

We follow the power normalization as suggested by numerous recent works followed by l_2 normalization. Each code has been through a point-wise normalization $f(z) = sign(z)|z|^{\alpha}$ where $0 \leq \alpha \leq 1$ is the normalization parameter. The idea of power normalization is to depress the noisy value of the representation and gives relative smooth codes.

3.4.4 Discussion

1. **The relation of SuperCoding and MeanVector**: As can be seen in Table 3.1, the SuperCoding proposed here is a natural generation of GMM MeanVector. The MeanVector uses the adapted weighted mean vector as the representation. The SuperCoding considers further by introducing the adapted covariance. The extended representation naturally forms the linear kernel as the similarity measurement which has linear cost at training and testing stage.

2. **The relation of SuperCoding and FisherVector**: As shown in Figure 3.1, both SuperCoding and FisherVector calculate the first order and second statistics. However, these statistics are different in terms of their meanings. For FisherVector, those statistics are gradients with regards to the GMM model.

Then the measurement of these gradients forms the Fisher Kernel which aims to extract the discriminative information. The SuperCoding follows another strategy: Each image has been adapted to a new GMM. Thus measuring the distance of two images has been transferred to the problem of calculating the distance between two GMM distance. This difference has been illustrated in Figure 3.1. Another interesting observation is that we find the main computation cost of the SuperCoding and FisherVector are at the same step, i.e. the posterior calculation. It implies that once getting SuperCoding, we can easily obtain the FisherVector coding. The possible mutual enhancement can promote the performance for different applications.

3. **High dimensionality and compression**: The proposed high order image statistic representation often come with very high dimensionality, e.g. for a model with 1024 Gaussians, 8 tiles SPM for SIFT feature, the dimensionality would be $128 \times 1024 \times 8 \times 2 \approx 2$ million. When meeting large-scale problem, it is often intractable and the role of data compression becomes increasingly important. In this work, we adopt the Product Quantization (PQ) [42] which is to represent each fragment of the coding e.g. 8 dimension, using a simple codebook, e.g. 256 words. Thus a high compression rate will be achieved, i.e. 8 bits for 8 double/float.

3.5 Experiments

3.5.1 Experimental Setting

Object Classification: We perform object classification tasks on two different datasets. The parameter evaluation is conducted on PASCAL VOC 2007 dataset which is the "benchmark" dataset for object recognition. The PASCAL Visual Object Challenge (VOC) datasets [18] are widely used for many image understanding tasks and provide a common evaluation platform for both object classification and

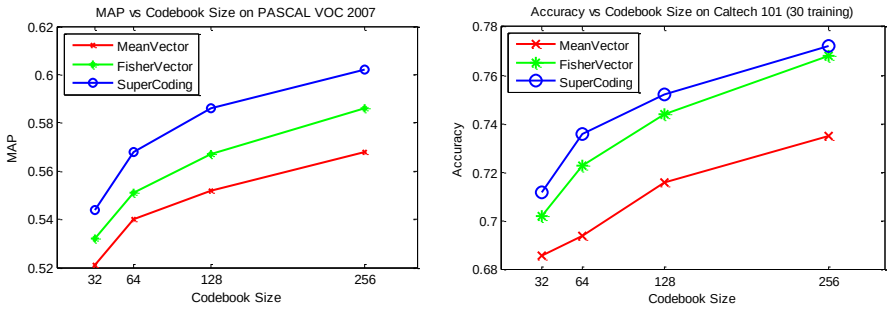

Figure 3.2: The effect of codebook size on different datasets.

detection. We use PASCAL VOC 2007 for experiments. VOC 2007 datasets contains 9,963. The two datasets are divided into "train", "val" and "test" subsets. We conduct our experiments on the "trainval" and "test" splits. The employed evaluation metric is *Average Precision* (AP) and *mean of Average Precision* (mAP) complying with the PASCAL challenge rules.

Scene Recognition We perform scene recognition on the SUN397 dataset [118] which is probably the largest database for scene classification. It contains 108,754 images over 397 well-sampled categories. The number of images varies across categories, but there are at least 100 images per category. Ten subsets of the dataset have been chosen for evaluation, each of which has 50 training images and 50 testing images per class. We follow the common experimental setting [118] on this database: In each experiment, different number of images are used for training, and all the 50 testing images are used for testing no matter what size the training set is.

3.5.2 The Effect of GMM Size

The number of the mixtures in the GMM is critical for representation power. Previous works clearly demonstrate that larger number of mixtures lead to higher accuracy. Here we compare different size setting on PASCAL VOC and Caltech 101. The obtained results are listed in Figure 3.2. It shows that performance of SuperCoding is always superior than the baseline of MeanVector and FisherVector.

3.5.3 Task 1: Object Classification

PASCAL VOC 2007

The detailed comparison results are listed in Table 3.2. We can observe that the GMM-based coding methods is general better than the VQ-based and the Sparse Coding-based methods. The mAP for VQ is only 0.484 even with non linear kernel. The LLC performs better than VQ and it only utilizes the linear solver. The GMM based methods achieve the most improvement. Only with the first order statistic, MeanVector [49] obtains 0.568 mAP and Fisher Vector obtains 0.586 with second order gradients. Our SuperCoding achieves impressive 0.602 mAP. This is the state-of-the-art result for single SIFT feature and no SPM setting as far as we know.

Table 3.2: Performance Evaluation on PASCAL VOC 2007 dataset.

	VQ(4K)	LLC(8K)	MeanVector [49]	FisherVector [14]	SuperCoding
aeroplane	0.685	0.641	0.750	0.760	**0.799**
bicycle	0.496	0.578	0.629	0.659	**0.676**
bird	0.394	0.350	0.465	0.466	**0.508**
boat	0.608	0.616	0.695	0.707	**0.709**
bottle	0.207	0.177	0.270	**0.309**	0.293
bus	0.480	0.528	0.639	0.656	**0.671**
car	0.679	0.730	0.790	0.779	**0.809**
cat	0.452	0.515	0.588	0.576	**0.617**
chair	0.470	0.448	0.451	0.480	**0.481**
cow	0.318	0.385	0.435	0.485	**0.485**
diningtable	0.352	0.328	0.480	**0.530**	0.522
dog	0.408	0.353	0.437	0.455	**0.461**
horse	0.664	0.710	0.778	0.770	**0.807**
motorbike	0.518	0.578	0.644	0.657	**0.682**
person	0.796	0.786	0.837	0.830	**0.857**
pottedplant	0.236	0.139	0.257	0.300	**0.318**
sheep	0.351	0.348	0.485	0.515	**0.517**
sofa	0.429	0.391	0.446	0.481	**0.488**
train	0.671	0.694	0.761	0.766	**0.792**
tvmonitor	0.465	0.459	0.515	0.546	**0.556**
mAP	0.484	0.488	0.568	0.586	**0.602**
Time/Img	N.A.	16sec	1.0sec	1.1sec	1.2sec

3.5.4 Task 2: Scene Recognition

We also conduct our experiments on the SUN397 dataset which is the largest dataset for scene classification. We compare our result with the multiple feature combination result from [118] and the work using attribute as middle feature [116] and our implementation of FisherVector and MeanVector. It shows again that the proposed SuperCoding has comparable performance with FisherVector given the single/multiple feature setting. Even more, we can observe further improvement when naively combine the result of FK and SC.

Table 3.3: Scene recognition performance on SUN397 dataset.

Methods	Number of training samples			
dSIFT	5	10	20	50
MultiFea [118]	14.46	20.87	28.12	38.0
Context+Semantic [116]			35.6	
FisherVector	17.06	23.38	30.37	38.4
SuperCoding	17.53	24.02	30.7	38.59
MeanVector	14.25	21.02	27.85	36.01
FV+SC	18.2	24.67	31.49	39.17
dSIFT+CM	5	10	20	50
FisherVector	20.13	27.43	35.24	43.96
SuperCoding	20.51	27.83	35.78	44.43
MeanVector	17.92	25.53	33.45	42.02
FK+SC	**21.2**	**28.53**	**36.51**	**45.09**

3.5.5 Data Compression vs. Performance

In [1], the authors thoroughly compared the performance vs. data compression for Product Quantization. It appears that for the feature coding with obvious data structure, e,g. Fisher Vector, the PQ achieves impressive good tradeoff between compression rate (CR) and recognition performance. For example, in their experiments, the performance dropped about 1% with a 32 compression rate. We follow this setting and conduct the experiment to verify the effectiveness of using PQ for SuperCoding. The detailed mAPs on VOC 2007 dataset are shown in Table 3.4.

Table 3.4: Data Compression vs Performance on VOC 2007 with Product Quantization [1].

	FisherVector	SuperCoding
Baseline	0.586	0.602
Compressed(CR=32)	0.574	0.586
Peformance Drop	-0.012	-0.016
Data Size(Before& After)	736MB vs. 23 MB	736MB vs. 23 MB

It can be observed that the performance of SuperCoding is slightly dropped (0.016) due to the data compression step. However, we can add more compressed training samples with this high compression rate (32) to improve the performance. This property is very important in the problem of large scale image classification in terms of large scale training samples and high dimensional feature representation.

3.6 Conclusion

In this chapter, we firstly reviewed the recent coding methods including the traditional VQ-based coding and the Sparse Coding-based methods. Then we focus on GMM-based coding. From the point view of GMM adaption, we extend the current the adaption-based method to the second order while retaining the favourable linear kernel representation. The experimental part demonstrated the effectiveness of the proposed SuperCoding.

Chapter 4

Generalized Hierarchical Matching/Pooling with Side Information

In this chapter, we aim to study the problem of "Feature Pooling". We define the term **Feature Pooling** as the process that select subsets of the encoded features and get the pooled feature over these subsets. For example, for Sparse Coding based methods [11], the "Max Pooling" is often used to select the max response from the pool of encoded feature and use this as the representation. However, the "Pooling" process is not only restricted to this simple logic operation. The underlining nature of the defined "Feature Pooling" is to select subsets according to some rules so that these subsets have more comparable meaning. For example, the spatial pooling which is widely used in image classification, i.e. Spatial Pyramid Matching [7], forms the subsets according to the spatial locations. This approximate geometric alignment can better make the pooled feature comparable at different levels.

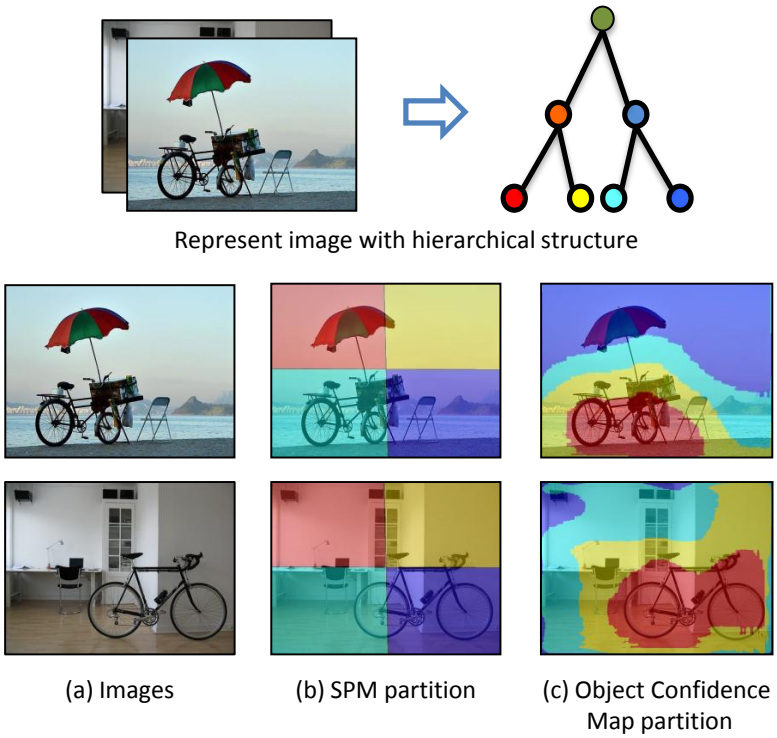

Figure 4.1: Illustration of the hierarchical matching representation. The local features are pooled according to partition of (b) traditional SPM and (c) the proposed object confidence prior. The figure shows our framework is superior than SPM in object matching across different images. For better viewing of all figures in this chapter, please see original color pdf file.

4.1 Introduction

In this work, we focus on image classification according to the objects contained in the images. More specifically, we focus on the classification of complex images which contain objects as well as cluttered background areas. Ideally, different parts of image should serve different roles for the classification. The appearance model of object itself plays a key factor while rich context information from background is helpful for the classification process. However, since the objects may only occupy a small portion of the images, rich context information as well as background noise introduced by the rest area of the image must be well handled in practice. State-of-the-art methods following the bag-of-words (BoW) framework [6] mainly contain

three steps: local feature extraction, feature encoding/pooling, and classifier learning. The local features are extracted from the dense grids, or via sparse interest point detection in the images. Feature encoding forms global image representations, e.g. a frequency histogram of visual words, which encodes the local features with a predefined visual dictionary such that the image representation has a comparable unified coordinate. The classifier learning step generally uses the kernel built on matching scores of the global image representations.

Traditional BoW framework equally encodes all local features and does not emphasize any elements with regard to image layout. Hence, pyramid structure representation is often used to extend the global BoW representation in image classification, e.g. Spatial Pyramid Matching (SPM) [7] for natural scene classification. SPM models global geometric correspondence by partitioning the image plane into increasingly fine sub-regions. The success of SPM comes from the valid assumption that the images with similar scene and geometry layout possibly belong to the same category. However, we argue that this representation is not optimum for object-centered recognition problem. As Figure 5.1 indicates, the spatial partition based on SPM may have mismatch problem caused by different object locations and scene layouts. In other words, if a prior knowledge, e.g. the possibility of object existence confidence in the image as shown in Figure 5.1 is acquired, we can construct the representation to match the corresponding object and background more accurately.

To this end, we propose a generalized hierarchical matching/pooling (GHM), which is capable to integrate different kinds of prior knowledge, including clues of object layout, for enhancing feature matching and towards object-oriented recognition. The prior knowledge, which is called *side information* in this chapter, is associated with each local feature in image. Using the side information, the image local feature pool can be clustered into cells and further a coarse to fine hierarchical representation can be generated. Since the partition of the cells is guided with side information more semantically concerned, the encoding within each cell tends to be

more semantically matchable and thus is expected to achieve better performance. Figure 5.1 demonstrates an example of how object-level side information is supplied to the proposed GHM framework. The side information of object confidence map can be used as an object-oriented prior for spatial partition of the image local feature pool. Consequently the images represented as hierarchical structures could carry out a coarse to fine matching.

Our contributions are two-fold. First, we propose the Generalized Hierarchical Matching framework for image classification. It gracefully extends the popular pyramid matching work, but further enables us to integrate other semantically useful side information with the flexibility. Second, two novel kinds of side information, i.e. object confidence map and visual saliency map, are introduced to enhance object-oriented image classification tasks based on the proposed GHM framework.

4.2 Related Work

4.2.1 Hierarchical Matching

Pyramid structure representation is often used to extend the global BoW representation in image classification, e.g. Spatial Pyramid Matching (SPM) [7] and Pyramid Match Kernel (PMK) [8]. SPM models approximate geometric layout by partitioning the image plane into increasingly fine sub-regions, and due to its better performance and simple implementation, it has become a standard procedure for image classification. However, for object-oriented classification, the increased complexity brought by SPM cannot contribute much to the recognition target because the object may appear in arbitrary position within an image, which thus may reduce the recognition efficiency and bring misalignment issue due to the unpredictable object locations in images.

PMK maps each feature set to a multi-resolution histogram that preserves the individual features' distinctness at the finest level. The histogram pyramids are

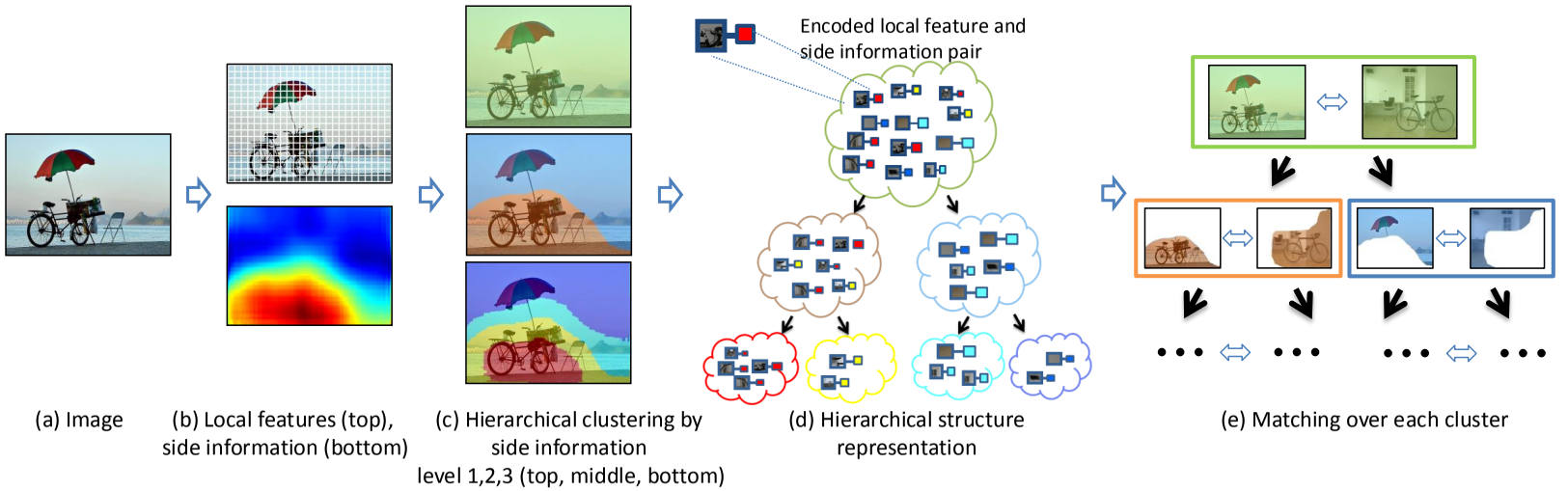

Figure 4.2: Diagrammatic flowchart of the proposed framework for image classification. The image is along with the (b) local features and side information. (c) The side information is hierarchically clustered to different levels. Different color mask represents different clusters at each level. (d) The encoded features are pooled over each cluster to form the hierarchical representation. (e) Finally, the matching over each corresponding cluster is performed.

then compared using a weighted histogram intersection computation, which implicitly defines the correspondence based on the finest resolution histogram cell where a matched pair first appears. It focuses on the mismatch problem caused by inaccurate Vector Quantization in feature encoding procedure. GHM framework well generalizes the SPM and PMK approaches and Section 4.3.3 will detail their relationship.

4.2.2 Saliency-guided Object Recognition

The saliency map [52] is a topographically arranged map that represents visual saliency of a corresponding visual scene. The purpose of the saliency map is to represent the conspicuity or "saliency" at every location in the visual field by a scalar quantity and to guide the selection of attended locations, based on the spatial distribution of saliency. Many of these saliency models are based on findings from psychology and neurobiology and explain the mechanisms guiding attention allocation [53, 52]. More recently, a number of models [54, 55] attempt to explain attention based on more mathematically motivated principles. Both types of models tend to rely solely on the statistics of the current test image when it comes to computing the saliency of a point in the image.

Some previous studies attempt to use saliency map as guidance for object recognition. [56] use color to guide attention by means of a top-down category-specific attention map. The color attention map is deployed to modulate more shape features from regions within an image that are likely to contain an object instance. [57] attempt to solve image classification using a biologically-inspired model to approximate the human eye fixations. These fixations are extracted from the feature maps at the sampled location, followed by probabilistic classification and the acquisition of additional fixations. The major difference between the proposed saliency map based GHM algorithm and these methods lies on how to utilize the saliency maps. In other words, GHM attempts to re-partition the features so that the group of

features has more meaningful structure and each layer of partition has consistent elements to be matched.

4.2.3 Region-based Object Recognition

Recently, some work attempts to process the object recognition at the image region level. [58, 59] explore multiple instance learning respectively to classify images by the highest scored image region. Following this idea, [60] use a latent-SVM model, which scores an image using all regions and associates each region with a latent variable indicating whether the region represents the object of interest or not. The solution takes the classification and foreground estimation into a joint inference framework. Though simpler than our proposed two-step solution, the critical drawback of the joint inference is that it will restrict the source of side information and cannot handle information from too complex sources. Other similar recognition work for image classification also exists. [61] propose to segment the images into foreground and background within co-segmentation scenario to improve image classification performance. [62] define a Region-Of-Interest in the image and take the maximum response over the coarse image grid as the output of classifier. Comparing to these region-based approaches, the GHM framework aims to utilize all image information including object itself and context from different kinds of sources.

4.3 Generalized Hierarchical Matching/Pooling

4.3.1 Image Classification Flowchart

Figure 4.2 shows the diagrammatic flowchart for image classification. Each image is expressed as a bag of orderless pairs I, each of which includes a local feature vector x_i encoded as c_i over a visual dictionary, and the side information f_i from priors and/or context, i.e. $I = \{\{x_i, c_i\}, f_i\}_{i=1}^{N}$. The side information is used for hierarchical clustering of the encoded local feature.

Table 4.1: Unified framework of Generalized Hierarchical Matching

Method Name		Side information	Coding method	Similarity function
PMK [8]		Histogram Index	Vector Quantization	Intersection
SPM	General [7]	Location Coordinate	Vector Quantization	Arbitrary
	ScSPM [11]		Sparse Coding	Linear
	ImprovedFV [14]		Fisher Vector Coding	Linear
The proposed GHM		Object Confidence Map, Visual Saliency Map	Arbitrary	

Along with the image itself, we may obtain the side information from various sources, e.g. the object confidence map denoting the existence probability of an object from object detector as shown in Figure 4.2. The side information is quantized into M discrete types. The encoding vectors c_i are assigned into different levels of clusters according to the quantization of side information, and form the hierarchical matching representation. To measure the similarity of two images $I_1 = \{\{x_i^1, c_i^1\}, f_i^1\}_{i=1}^{N_1}$ and $I_2 = \{\{x_i^2, c_i^2\}, f_i^2\}_{i=1}^{N_2}$, a kernel is constructed based on this representation. The kernel could be fed into any popular machine learning algorithm for classification purpose. We detail the GHM representation in the following section.

4.3.2 Hierarchical Matching Kernel

Assuming there are two images I_1, I_2, we can allocate each pair in I_1, I_2 into a hierarchical structure $G = \{G_1, G_2, ..., G_L\}$, where L is the number of hierarchical levels. Same as in previous hierarchical matching algorithms, only the elements grouped to the same cluster are supposed to match to each other. Hence we quantize all encoded feature vectors into M_l cells at level l, and the corresponding pooling is functioned on each cluster. We explored two ways to construct hierarchical structure. One is to perform hierarchical clustering on single/combined maps. The clustering is operated on the side information of training set. The other one is to design mixed meaningful structure from prior knowledge instead of automatic hierarchical clustering.

Then we can define a cluster kernel through a similarity function, *i.e.* $\kappa_{12}^{jl} = S(I_1, I_2, G_l^j)$, where S is a similarity function based on local feature cluster G_l^j on cell j at level l for images I_1 and I_2, and κ_{12}^{jl} represents the similarity value on cell j at level l. Then the similarity kernel between two images is defined as the weighted sum of similarity values:

$$K_{12} = \sum_{l=1}^{L} \sum_{j=1}^{M_l} w_{jl} \kappa_{12}^{jl}. \tag{4.1}$$

Similar to other hierarchical methods, it degenerates to a standard BoW when $L = 1, M_l = 1$. It is easy to verify that if the κ^{jl} is a Mercer Kernel, then K is also a Mercer Kernel and thus it can be embedded into any popular kernel-based machine learning algorithm. The kernel weight w_{jl} can be intuitively set or learnt by popular Multiple Kernel Learning (MKL) [63] method.

4.3.3 Generalization and Flexibility

In Table 4.1, we demonstrate the generalization capability with various configurations of GHM to realize previous hierarchical matching algorithms as well as our proposed object-oriented recognition with new side information.

First, we show that the Pyramid Match Kernel (PMK) [8] is one exemplar of the GHM framework. To encode and match the local feature with more accurate quantization, PMK uses multiple levels of local feature pooling and intersection kernel matching based on Vector Quantization (VQ). The pool of local image features is hierarchically partitioned into clusters according to their histogram indices and the final matching score is defined as weighted sum of all cluster matching scores, which can be straightforwardly explained by our GHM framework. As aforementioned, SPM uses the location coordinate of local features as side information for clustering and it is easily adapted as one special case of the GHM framework. GHM is the general form of PMK and SPM, which use diverse side information respectively.

Table 4.1 also illustrates that GHM framework can embed any popular cod-

ing method with flexibility. The BoW feature encoding approaches such as Sparse Coding [11] and Locality-constrained Linear Coding (LLC) [12] introduce soft assignment for local feature quantization. Fisher encoding[14] and Super vector encoding [15] capture the average first and second order differences between local features and their distribution centres modeled by Gaussian Mixture Models. Most of the coding work include SPM as the spatial pooling step. GHM could also help this step and indicate image coding on well-designed clusters based on provided side information, e.g. object confidence map and visual saliency map which is detailed in next section.

4.4 Side Information Design

In this section, we design two schemes to construct side information: (1) the object confidence map which reveals the possibility of a local patch containing a object. (2) the visual saliency map which takes advantage of natural image statistic and distinguishes the foreground against the background. Further these two kinds of information, as well as the location coordinate information, can be combined parallelly or hierarchically as side information to reflect meaningful structure for GHM-based image recognition.

4.4.1 Object Confidence Map

For object recognition task, it is commonly believed that in traditional well-proposed object recognition datasets, such as CMU PIE face [64] and Caltech-101 [29], most objects are cropped after fine alignment and with little background noises, and such preprocess always leads to much better performance. But it does not work for general object recognition datasets such as Caltech-UCSD Birds [65], PASCAL VOC [18], etc, where no object pre-alignment and cropping is performed. Intuitively the most useful recognition prior for these object-unaligned images is object

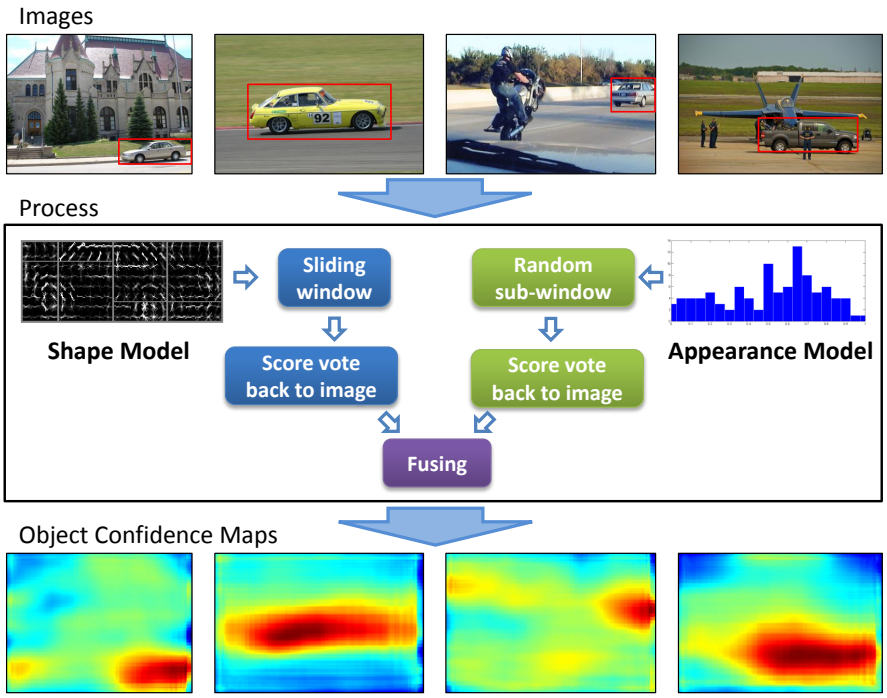

Figure 4.3: Object confidence map and some examples from car category.

position. And object position should be extremely beneficial for fine-grained image classification task.

The steps to construct an object confidence map, denoted as GHM Object, is illustrated in Figure 4.3. For each object category, e.g. car, we train one shape-based and one appearance-based object detectors, respectively. The usage of two detectors is to guarantee both high precision and high recall on object detection since none of the detectors can achieve this alone and they complement each other in certain way. Instead of constructing the local classifiers on a super-pixel representation as in other work [66, 67], we use square grid samples and sliding-window approach for efficiency consideration.

The shape-based object detection adopts the state-of-the-art part-based model from [19] using HOG [3] features. And the appearance-based object detector is trained with BoW features. We use dense SIFT [4] and LBP [5] as local features and

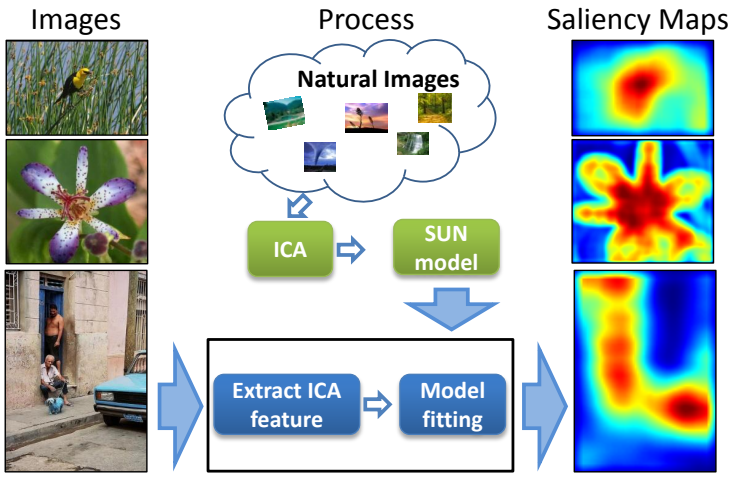

Figure 4.4: Visual saliency map generation and some examples.

the codebook sizes for dense SIFT and LBP are 2000 and 1000 respectively. Each detection sub-window is divided into 3x1 spatial pyramid to provide weak geometry constraint. The BoW histogram is mapped into high-dimension space via Additive Kernel Mapping [68]. This nonlinear transformation guarantees the possibility of using linear classifier for fast detection. We further accelerate the detection by using integral image to construct BoW representation within sub-window. Multiple scale detection is performed in each image and the obtained multi-scale scores are averaged to get final single object confidence map.

4.4.2 Visual Saliency Map

For some object categories, such as flowers, detection models may perform poorly. We propose another apparent foreground prior on finding visually salient image regions from human attention models and construct saliency maps as side information, denoted as GHM Saliency.

We consider the saliency under the scenario of general visual classification problem. In other words, the saliency information should reflect how human sees the objects against the natural background clutter. For this reason, we use the saliency

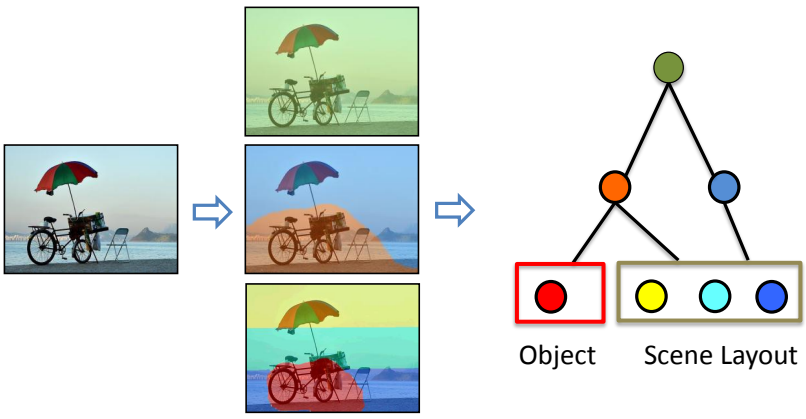

Figure 4.5: Combine object confidence map and spatial layout into one GHM. Level 2 is clustered according to object confidence map. Level 3 is designed for foreground matching and scene layout matching.

model SUN (Saliency Using Natural statistics) [69]. This measure of saliency is based on natural image statistics, rather than based on a single test image, providing a straightforward explanation for many search asymmetries observed by humans.

The SUN model illustrated in Figure 4.4 defines the bottom up saliency as $P(F)^{-1}$, where F indicates the transformed color features through Independent Component Analysis (ICA) [70] on local color patch. Since the components of F have been made largely statistically independent by ICA, SUN models $P(F)$ as the product of unidimensional distributions: $P(F = f) = \prod_i P(f_i)$, where f_i is the ith value of these filter responses at this location. The ICA feature responses to natural images can be fitted very well using Generalized Gaussian Distributions [71], and we obtain the shape and scale parameters for each ICA filter by fitting its response to the ICA training images.

4.4.3 Side Information Combination

The nature of the GHM framework enables us to flexibly combine side information from various sources. One straight way to combine the side information is parallel information fusion, e.g. the spatial location information and the saliency

map coupling as $f = \{f_{location}, f_{saliency}\}$ collaboratively. The clustering over this combination aims to consider the geometric constraint and saliency information so that each of the sub-cluster in the image contains equal amount of salient area. We denote this parallel combination as GHM LocSaliency.

Another feasible solution for side information combination is to design mixed hierarchical structure. Most natural images (e.g. those from PASCAL VOC dataset) contain large amount of background area, which in fact supplies rich contexts for the recognition of certain object categories e.g. sky for aeroplane/bird, urban scene for various vehicles. This motivates us to design a configuration which simultaneously matches foreground objects and background scenes. The background confidence can be simply obtained from the foreground object confidence with reversed process, i.e. small object confidence map value meaning higher possibility of background. The spatial layout is proved to be useful for the recognition of background scenes [7]. We design a 3 level hierarchical structure with combined side information: the whole image as level 1, object confidence map is used in level 2 as the foreground confidence map, and the small value denoting the background area will be further utilized in level 3 to construct the 3×1 spatial layout modeling the background scene as shown in Figure 4.5. We denote this hierarchical combination as GHM ObjHierarchy.

In summary, we propose two useful resources of side information to fit into proposed GHM framework for image classification, i.e. the object confidence map and the visual saliency map. We further propose to associate the side information from multiple resources, either through simple parallel combination or via sophisticated hierarchical design to reflect the semantic complexity in real image recognition task.

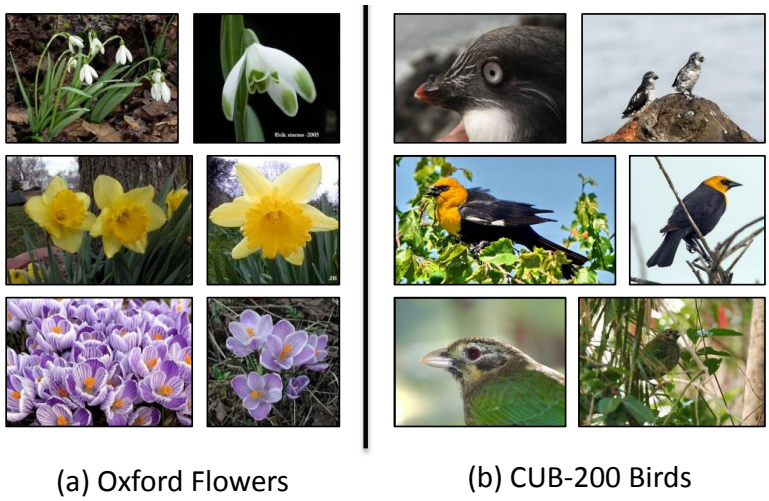

Figure 4.6: Sample images from Oxford Flowers 17 and CUB 200. The images in the same row belong to the same category.

4.5 Experiments

4.5.1 Datasets and Metric

We evaluate our proposed Generalized Hierarchical Matching framework on several popular datasets, the recently released Caltech-UCSD Birds 200 (CUB-200) [65], the Oxford Flowers 17 (Flowers 17) [72] and 102 (Flowers 102) [73], and the PASCAL Visual Object Challenge (VOC) datasets [18].

The CUB-200 contains 200 bird categories and 6033 images in total. It is created to enable the study of subordinate categorization. The Flowers 17 [72] dataset contains 17 different flower species with 80 images per category. The dataset provides three different data splits with each including 60 training and 20 test images. The Flowers 102 [73] dataset includes 8289 images divided into 102 categories with 40 to 250 images per category. We use the provided data split with 20 images per-category for training and the rest for testing. Figure 4.6 shows some examples of the Oxford Flowers and CUB-200 images. It can be seen that these two fine-category classification datasets are very challenging due to the large intra variances.

The PASCAL Visual Object Challenge (VOC) datasets [18] are widely used for many image understanding tasks and provide a common evaluation platform for both object classification and detection. We use PASCAL VOC 2007 and 2010 datasets for experiments. VOC 2007 and VOC 2010 datasets contain 9,963 and 21,738 images respectively. The two datasets are divided into "train", "val" and "test" subsets. We conduct our experiments on the "trainval" and "test" splits. The employed evaluation metric is *Average Precision* (AP) and *mean of Average Precision* (mAP) complying with the PASCAL challenge rules.

4.5.2 Experimental Details

Baseline Configuration: For CUB-200, Flowers17 and Flowers102 datasets, the local features used for the image recognition are RGB color moment and dense SIFT descriptors. The implementation of dense SIFT is based on VL-Feat [51] using multiple scales setting (spatial bins are set as 4, 6, 8, 10) with step 4. We use the improved Fisher vector coding [14] with SPM setting which has demonstrated the superiority over other coding methods in a fair setting [13]. The size of Gaussian Mixture Model in Fisher vector coding is set to 256 for these two features separately. One-vs-All SVM is learnt for each category using the representation generated by GHM and returns the class with the maximum score over all the image classifiers. The SPM is with typical setting, 3 levels are used, $1\times1, 2\times2, 4\times4$ spatial separation.

For PASCAL VOC 2007 [18] datasets, we use only dense SIFT feature with the Fisher vector coding to make it comparable with other popular works. We also conduct the experiments with "heavy" setting to obtain state-of-the-art performance for PASCAL VOC 2010 dataset. For local features, we extract dense SIFT, HOG, color moment and LBP features in a multi-scale setting. Typically, the number of local features for each image is around 30K for SIFT, 5K-10K for others. This is critical in feature coding to produce non-sparse representation. All these features are also encoded with improved Fisher vector coding. One-vs-All SVM is learnt and

the performance is evaluated by AP.

Side Information Generation: We implement the proposed two kinds of side information: (1) the supervised object confidence map and (2) the unsupervised visual saliency map. The two detectors used to generate object confidence map are trained with PASCAL VOC images. For part-based model [19], the HOG and LBP features are used for object description and the number of part models for each object category is set to 8. For appearance-based approach, we sample 4000 sub-windows with different size and scale and perform the BoW based object detector on these sub-windows. We construct the hierarchical structure with three-level clusters, each of which includes 1, 2, 4 nodes respectively on the training images. For each class, we sample the responses from the positive images and the same number of negative images and get various cluster centers with clustering process. Finally each local feature is assigned to the nearest center at each level.

For the saliency map generation, we follow the SUN [69] framework and adopt the ICA filters model from [57]. These filters are learned with the images from the McGill color image dataset [74]. For the following experiments, we use this setting unless otherwise stated: three-level clusters for hierarchical structure, each of level with 1, 2, 4 nodes respectively. The clustering is operated on single image but not cross dataset since we find that the saliency map values for different images are not comparable.

The weight w_{jl} is intuitively set without fine tuning: the higher confidence cluster has higher weight within each level and the weights are normalized to have unit sum for each level.

4.5.3 Exp1: Caltech-UCSD Birds 200

We first evaluate our methods on the newly released Caltech-UCSD Bird 200 dataset and show that the visual saliency map and the object confidence map are very helpful for the fine categorization problem. The dataset is extremely challenging, and its

Table 4.2: Performance comparison on Caltech-UCSD Birds 200. The proposed methods lead to the highest recognition accuracy.

Methods	Recognition Acc.
[61]	17.0
[75]	19.0
BoW Baseline	15.2
FVSPM	15.0
GHM Saliency	**18.1**
GHM Object	**19.2**

authors report only 19% recognition accuracy [75] when using ground truth masks. The recognition performance is listed on Table 4.2 (using the suggested 20 training images per class split). [61] first segment the image into foreground and background and then extracted feature on the foreground. We also implement the Fisher vector coding with SPM (FVSPM) [14].

For this fine-grained categorization problem, the spatial layout has no exact meaning for different fine classes since most of classes share the same background. We propose to use saliency map (GHM Saliency) and the object confidence map (GHM Object) as a guidance to partition the images into different levels. The object confidence map is obtained by performing the "bird" detector trained from VOC 2010 datasets. Both of the results are much better than FVSPM. The results show that the unsupervised saliency performs very well on this dataset and the object confidence map gives strong support for separating the foreground and background so that fine-grained categorization is possible.

4.5.4 Exp2: Oxford Flowers 17 and 102

We compare our proposed GHM method with other state-of-the-art results on Oxford Flowers datasets. [76] adopt multiple feature combination method. [57] use the same saliency map as ours. [61] use segmentation to get the foreground area which is current leading method in this dataset. It is almost impossible to train a "flower" detector for this dataset, on the other hand, the saliency map shows strong

Table 4.3: Performance comparison on Oxford Flowers datasets.

	Flowers 17	Flowers 102
Methods	Recognition Acc.	
[76]	88.5 ± 3.0	–
[57]	–	72.8
[61]	90.4 ± 2.3	80.0
FVSPM	93.0 ± 1.7	82.0
GHM Saliency	**93.1 ± 1.8**	**82.3**
GHM LocSaliency	**93.5 ± 1.5**	**82.6**

evidence over this datasets: most of the flowers are within the salient foreground area of the images. So we evaluate the GHM with saliency map performance and its combination with spatial information. The recognition performances on Oxford Flowers 17 and 102 are listed on Table 4.3.

The GHM with the saliency map (GHM Saliency) achieves comparable performance with FVSPM. It shows that the saliency map is comparable prior for object recognition with the weak geometric alignment at these two datasets. It is worth noting that for these two datasets, we use compact representation. i.e. 3 levels of saliency map with total 1+2+4=7 cells compared with 21 cells in SPM. We also use the parallel combination design of side information by using saliency map together with spatial information (GHM LocSaliency). The side information is designed as $f = \{f_{location}, f_{saliency}\}$. Then a 2 level GHM with $1 \times 1, 2 \times 2$ setting is constructed. The results show the additional improvement over the single channel of side information with very compact representation.

4.5.5 Exp3: VOC 2007 and VOC 2010

We evaluate our proposed method on PASCAL VOC 2007 and VOC 2010 dataset. The classification results on VOC 2007 are listed on Table 6.3. INRIA [77] is the winner of VOC 2007 and uses multiple kernel learning to balance the weight of different features. LLC [12] is the popular state-of-the-art feature coding method. We

follow coding method in FisherVec [14] which results in mAP 58.3%. Our baseline FVSPM (mAP 60.6%) achieves higher performance than FisherVec approach, since more dense SIFT features with smaller step for one image is extracted. All these methods report much lower mAP than the leading score in [78] which uses "heavy" setting. Also note that the object classes in this dataset are conflicted with the saliency map assumption since many of the concerned classes and object instances in VOC are not at the foreground area, e.g. bottle, chair, tv. So we mainly use the object confidence map for each class and encode the features with GHM. The results (GHM Object) show mAP +3% absolute improvement over the baseline method using SPM. The prior of object confidence map is much stronger than the spatial layout for object-oriented classification.

VOC images contain large amount of background area which provides rich context information for recognition of certain objects. This also leads us to design a configuration which simultaneously matches foreground objects and background contexts. We design the mixed hierarchical structure setting with combined side information as proposed in Sec. 4.4.3. The significant performance improvement from mAP 60.6% (by FVSPM) to 64.7% (by GHM ObjHierarchy) demonstrates the effectiveness of this hierarchical structure of mixed spatial layout and object confidence modeling.

We also compare our method with the current leading approach [78] on PASCAL VOC 2010 with "heavy" setting. We adopt the Context SVM method with its configuration which combines the object detection and classification in a context-aware scenario, but generate the representation with GHM ObjHierarchy. The classification results on VOC 2010 are listed in Table 4.5. The final results of GHM ObjHierarchy outperform the leading scores in VOC 2010 challenge. The usage of hierarchical object and scene layout side information provides great gain for this classification task.

Table 4.4: Classification results (AP in %) on VOC 2007. The proposed GHM Object and GHM ObjHierarchy outperform the baseline methods.

	INRIA [77]	LLC [12]	FisherVec [14]	FVSPM	GHMObject	GHMObjHierarchy
plane	**77.5**	74.8	75.7	75.8	77.0	76.7
bike	63.6	65.2	64.8	68.1	73.5	**74.7**
bird	**56.1**	50.7	52.8	51.6	51.8	53.8
boat	71.9	70.9	70.6	71.6	71.1	**72.1**
bottle	33.1	28.7	30.0	30.0	37.1	**40.4**
bus	60.6	68.8	64.1	69.4	70.8	**71.7**
car	78.0	78.5	77.5	78.9	82.3	**83.6**
cat	58.8	61.7	55.5	61.9	63.4	**66.5**
chair	53.5	54.3	**55.6**	50.7	52.0	52.5
cow	42.6	48.6	41.8	50.6	55.2	**57.5**
table	54.9	51.8	56.3	55.5	60.9	**62.8**
dog	45.8	44.1	41.7	45.8	49.9	**51.1**
horse	77.5	76.6	76.3	79.2	80.7	**81.4**
motor	64.0	66.9	64.4	69.1	71.2	**71.5**
person	85.9	83.5	82.7	84.6	86.0	**86.5**
plant	36.3	30.8	28.3	31.9	36.3	**36.4**
sheep	44.7	44.6	39.7	49.9	53.8	**55.3**
sofa	50.6	53.4	56.6	53.1	59.8	**60.6**
train	79.2	78.2	79.7	79.7	79.6	**80.6**
tv	53.2	53.5	51.5	54.4	**57.8**	57.8
mAP	59.4	59.3	58.3	60.6	63.5	**64.7**

4.6 Conclusions and Future Work

In this work, we introduced a generalized hierarchical matching (GHM) framework for image classification task. This general and flexible scheme allows us to embed any useful side information into the image recognition framework. We also presented two novel exemplar approaches for side information generation towards object-oriented recognition, i.e. object confidence map and visual saliency map. Extensive experimental results clearly demonstrated the proposed GHM together with designed varieties of side information could achieve state-of-art performance on diverse and popular image recognition datasets. In future, we shall further explore more semantically meaningful side information and new approach for combining different types

Table 4.5: Classification results (AP in %) on VOC 2010. The proposed GHM ObjHierarchy outperforms the state-of-the-art performance.

	NLPR [18]	NEC [18]	ContextSVM [78]	GHMObjHierarchy
plane	90.3	93.3	93.1	**94.3**
bike	77.0	72.9	78.9	**81.3**
bird	65.3	69.9	73.2	**77.2**
boat	75.0	77.2	77.1	**80.3**
bottle	53.7	47.9	54.3	**56.3**
bus	85.9	85.6	85.3	**87.3**
car	80.4	79.7	80.7	**83.8**
cat	74.6	79.4	78.9	**82.2**
chair	62.9	61.7	64.5	**65.8**
cow	66.2	56.6	68.4	**73.7**
table	54.1	61.1	64.1	**67.0**
dog	66.8	71.1	70.3	**75.9**
horse	76.1	76.7	81.3	**82.3**
motor	81.7	79.3	83.9	**86.5**
person	89.9	86.8	91.5	**92.0**
plant	41.6	38.1	48.9	**51.7**
sheep	66.3	63.9	72.6	**75.1**
sofa	57.0	55.8	58.2	**63.3**
train	85.0	87.5	87.8	**89.9**
tv	74.3	72.9	76.6	**77.3**
mAP	71.2	70.9	74.5	**77.2**

of side information.

Chapter 5

Context Modelling: High Level Task Context for Object Detection and Classification

In this chapter, we study the problem of **Context Modeling**. Traditionally, the context is often considered as special features. Most of the existing strategies [10][9][17] utilize the context via feature concatenation, model fusion or confidence combination, and take the context as another independent component. However, context may have unstable distribution, and its reliability and noise level are not controllable. Therefore it demands adaptive contextualization with proper constraints from the main task to avoid the inappropriate usage of context information. Harzallah et al. [17] introduced the pioneer work for object detection and classification contextualization through the postprocessing of probability combination. The mutual contextualization shows promising performance improvement. However the learning scheme which seamlessly integrates the context information for collaborative learning is missing.

5.1 Introduction

Recognizing objects in an image requires combining many different signals from the raw image data. Two kinds of information are often used: the local appearance that describes the object itself and the global representation that captures the image specific information. These two types of information are often used in two tasks on visual recognition: object detection and classification. Object detection and classification are two key tasks for image understanding, and have attracted much attention in the past decade [19] [7] [11]. The object classification task aims to predict the existence of objects within images, whereas the object detection task targets to localize the objects. Several image databases tailored for these two tasks have been constructed, such as Caltech-101 [79]/256 [30], SUN dataset [80] and PASCAL Visual Object Classes (VOC) [18]. Many efforts [19][7] have been devoted to these two tasks.

Beyond various image descriptors and modeling methods, the usage of context for visual recognition has become increasingly popular for enhancing the algorithmic performance. Many recent studies have demonstrated considerable improvements for object detection and classification by using external information, which is independently retrieved and complementary with traditional image descriptors. Specifically, the external context includes user-provided tags [81][10], surrounding texts from Internet [82][83], geo-tags and time stamps [9], etc. The context may also be the information lying within individual images. Intuitively, spatial location of the object and background scene from the global view can be used as intrinsic context of the image [84][85].

We consider the context from the high-level task perspective. It has been demonstrated that the object detection and classification tasks can provide natural comprehensive context for each other without any external assistance, and thus can be mutually contextualized for performance boosting [17]. It is intuitively straightfor-

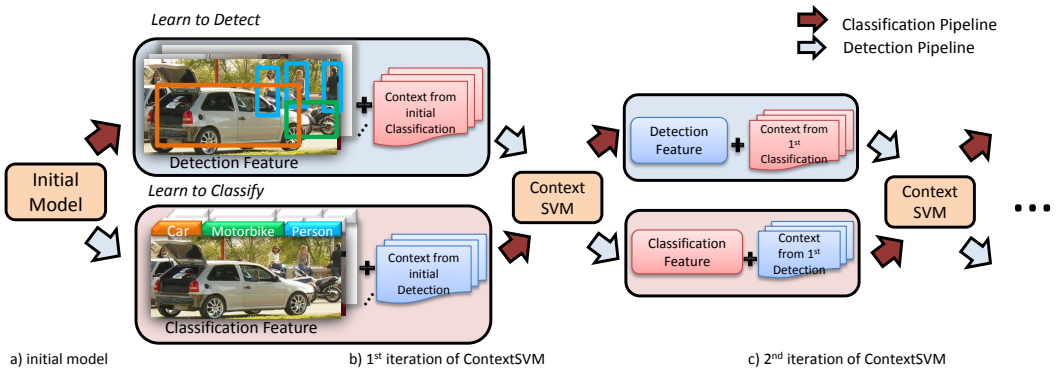

Figure 5.1: Illustration of the iterative contextualizing procedure. The object detection and classification tasks utilize context from each other and mutually boost performance iteratively. For better viewing, please see original color PDF file.

ward that for object classification task, the information from the local appearance promotes the performance significantly. For object detection task, the global context from object classification helps the detector better eliminate the false alarm. Although there are some works focusing on this direction, we notice that the underlying improvements brought by the context models for both two tasks have been underestimated. And the previous works take the context model in a multi-feature fusion fashion [81, 17] without dedicated design.

In this work, we develop a novel mutual contextualization scheme for object detection and classification based on the *Contextualized Support Vector Machine* (Context-SVM) method. First, we present a *contextualized learning* scheme via Context-SVM with the following characteristics:

- *Adaptive contextualization*: As many studies have shown [86][87], context should be activated to be supportive mostly for those *ambiguous samples* and thus the context effectiveness should be conditional on the ambiguity of sample classification. The Context-SVM is superior over traditional learning schemes by complying with this principle in its formulation.

- *Multi-mode contextualization*: The ambiguity nature of the recognition prob-

lem at the boundary requires elegant design of the context model. We are interested in designing the localized context model along the decision boundary which often shows various modalities. We propose to learn the multi-mode context model with mode selection function. Based on the general formulation, we further extend the context model to the ambiguity-guided mixture model. The mixture model naturally partitions the feature space at the decision boundary with regards to the ambiguity degree. Thus the proposed Context-SVM with multi-mode initialization can naturally embed the context model at the classification hyperplane.

- *Configurable model complexity*: The contextualization process should be efficient for both detection and classification tasks, and thus the solution should not involve many parameters. In this work, the Context-SVM with tractable control on the complexity of the context model is well formulated, so that the generalization capability is guaranteed.

Then we propose an iterative contextualization procedure based on the Context-SVM, such that the performance of object classification and detection can be iteratively and mutually boosted as illustrated in Figure 5.1. Extensive experiments show that Context-SVM can efficiently learn the context models under various conditions and effectively utilize context information for performance boosting. We implement and evaluate the proposed scheme on object detection and classification tasks of the VOC 2007, VOC 2010 datasets [18] and SUN09 [80], and the results are superior over the state-of-the-art on most object categories.

An earlier version of this manuscript was presented as [78]. This version includes a clearer motivation section with a refined max margin model. Two ambiguity modeling methods are introduced with deeper analysis. Additional diagnostic experiments are conducted on both VOC and SUN09 datasets and new state-of-the-art results are presented. In the following, we first briefly review the related

work for object recognition context modeling in Section 5.2. Then we introduce our ContextSVM model with two ambiguity modeling approaches in Section 5.3. Section 5.4 details our mutual and iterative contextulization for object detection and classification tasks. And we give extensive experiments on different datasets in Section 6.5.

5.2 Related Work

5.2.1 Context Modeling for Object Recognition

In recent years there has been a surge of interest in context modeling for numerous applications in computer vision. The basic motivation behind these diverse efforts is generally the same-attempting to enhance current image analysis technologies by incorporating information other than the image itself, e.g. semantic analysis result and metadata.

In the early work of Galleguillos and Belongie [88], the context refers to three main types of contextual information that can be exploited in computer vision: (1) the semantic context which refers to the likelihood of an object being found in some scenes but not in others, and from the point of view of modeling, can be expressed in terms of the corresponding object's probability of co-occurrence with other objects and the probability of occurrence in certain scenes; (2) the position (spatial) context which corresponds to the likelihood of finding an object in some positions and not others with respect to other objects in the scene; and (3) the size (scale) context which exploits the fact that objects have a limited set of size relations with other objects in the scene.

A natural way of representing the context of an object is in terms of its relationship with other objects, e.g. co-occurrence based context model [89]. An alternative terminology was proposed by Heitz and Koller [84] who introduced a "Things and Stuff" (TAS) context model. In their work, the terms "stuff" and "things" (originally

introduced by Forsyth et al. [90]) are used to distinguish "material" that is defined by a homogeneous or repetitive pattern of fine-scale properties, but has no specific or distinctive spatial extent or shape (stuff) from "objects with specific size and shape" (things). Heitz and Koller claimed that "classifiers for both things or stuff can benefit from the proper use of contextual cues". Rabinovich and Belongie [91] proposed a classification of contextual models for computer vision (in general) and object recognition (in particular), consisting of models with contextual inference based on the statistical summary of the scene (which they referred as Scene Based Context models, SBC for short) and models representing the context in terms of relationships among objects in the image (Object Based Context, OBC for short).

Also, some methods have been proposed to model the context in a comprehensive manner, e.g. [92], but they are quite specified and designed for one certain task, and thus cannot be generalized for our target in this work.

Only recently, object hierarchy context has drawn much research attention [93, 80]. The object hierarchy is the further research of object co-occurrence context under the assumption that objects should be related with a semantic hierarchy. With the increased number of object categories, object relationship is naturally exhibited as a hierarchical structure. Context modeling with hundreds or thousands of object categories seeks to model this relationship with high level semantic structure or learned from data [94].

5.2.2 Mutual Contextualization for Object Classification and Detection

Although there are lots of works on context representation and modeling, few of them focus on contextualization between object detection and classification, namely, high level task context.

For object classification, the task cares more about whether the image contains a certain kind of object rather than where it is. The task is solvable due to the facts

that (1) many datasets only concern the objects which occupy most of the images, e.g. Caltech 101 and 256 [79], (2) the same category objects often share similar scene level information, e.g. VOC and SUN09 datasets, and (3) the current prevalent object classification pipeline uses the sophistic feature encoding and learning method to extract image specific information which often reveals the object-specific contents, e.g. Fisher Vector Coding [14] and SVM classifier [95]. The methods used in classification are often built with a top-down manner that uses global information to infer the existence of a local object. For object detection, the task tries to localize the object within the image. Usually, the object detector models the object appearance [96] or object shape [3][19] through the annotated object samples while discarding the context information defined by the object surrounding. The localized nature of the object detector restricts the model to effectively differentiate the false alarm which occurs at obviously different context. Harzallah et al. [17] introduced the pioneering work for object detection and classification contextualization through the post-processing of probability combination.

Moreover, traditionally, the context is considered as special features. Most of the existing strategies [10][9][17] utilize the context via feature concatenation, model fusion or confidence combination, and take the context as another independent component. However, context may have instable distribution, and its reliability and noise level are not controllable. Therefore it demands adaptive contextualization with proper constraints to avoid the inappropriate usage of context information. In this work, we follow this line to design the learning scheme for utilizing context information.

5.3 Contextualized SVM

In this work, the *context* is generally defined as certain extra supportive information for one task, which is retrieved independently from the *subject* task [1]. In this section, we first introduce the probabilistic motivation of the contextualized SVM (Context-SVM) and derive its linear formulation based on the probabilistic motivation. We then propose two ambiguity modeling methods for the Context-SVM which enables the multi-mode context modeling. Finally, we extend the linear Context-SVM to the kernel version for more general usage.

5.3.1 Probabilistic Motivation

Let $x_i^f \in \mathbb{R}^n$ denote the features of a sample for the subject task, $x_i^c \in \mathbb{R}^m$ denote the features of the corresponding context, and $y_i \in \mathbb{R}$ denote the ground-truth class label. Then the entire training data can be expressed as

$$\{x_i = \{x_i^f, x_i^c\}, y_i; i = 1, 2, ..., N\}. \tag{5.1}$$

Generally, the objective of a discriminative learning model can be defined to maximize:

$$\prod_{i=1}^{N} P(y = y_i | x_i),$$

namely the Maximum a Posteriori (MAP).

There are two components within x_i, and usually the independent assumption of the subject features x_i^f and the context x_i^c is made and then maximizing the probability of label y for a given sample x_i, i.e. $p(y|x_i)$ can be approximated to

[1] We refer the main/principal task concerned as the *subject* task.

maximize the following formulation:

$$p(y|x_i^f)p(y|x_i^c). \tag{5.2}$$

The inference based on (5.2) is right for the traditional solution of confidence combination [17][9] or multiple feature/model fusion [10].

The independence assumption, however, is often invalid for real data, and hence we propose to infer the label probability by (5.3) which explicitly models the conditional usage of context with respect to the given subject features, i.e, maximizing $p(y|x_i) = p(y|x_i^f, x_i^c)$ is converted to maximizing:

$$p(y|x_i^f) \cdot p(y, x_i^c|x_i^f). \tag{5.3}$$

More specifically, we aim to infer the label probability via two components simultaneously. The first one is based on the subject features, i.e. $p(y|x_i^f)$, and the second one is based on the context features, which contribute to the inference when only ambiguous decision from the first component is expected, i.e. $p(y, x_i^c|x_i^f)$.

The second component is critical for a contextualized learning model. For object detection, the context of scene information from object classification is nearly the same for all detected windows within one image and may be unnecessary for many windows. Instead, only the most ambiguous detections need the assistance from context.

For object classification, the context from object detection generally shows low reliability due to the possible false alarms and the selective usage of context can effectively avoid the disturbance caused by the false context to those already high-confident object patterns.

5.3.2 Context-SVM: Formulation and Solution

General Formulation

For ease of formulation, we only consider the binary classification problem for object detection or classification task, i.e. $y_i \in \{+1, -1\}$ and the N_c-class problem can be decomposed into N_c binary classification problems through one-vs-all strategy. SVM [95] provides a general supervised learning framework by maximum margin optimization, and in this work, we extend SVM by introducing a novel parametrized model to describe the dependence between the context features and the subject features.

The general SVM learns a classifier over the subject feature space:

$$f(x_i, w_0) = w_0^T \cdot x^f + b. \tag{5.4}$$

We can relate this scoring function with the log probability $\log p(y|x^f)$. As the corresponding context features x_i^c can provide extra supportive information for the classification of x_i^f, we propose to utilize x_i^c to adapt w_0 for sample x_i. Then a sample-specific w_i can be obtained to substitute w_0, which essentially optimizes the margin of sample i and can consequently improve the discriminative power of the classifier. The probabilistic formulation indicates that we need to formulate the context model with regards to the subject feature distribution. Explicitly, we model the context model as,

$$\begin{aligned} \log p(y, x_i^c|x_i^f) &= \log p(y, x_i^c|x_i^f, \theta) && (5.5) \\ &\approx \sum_{r=1}^{R} u_{r,i}(x_i^f, w_0, \theta) q_r^T x_i^c; && (5.6) \end{aligned}$$

where $u_{r,i}$ is the ambiguity indicator function which determines how ambiguous the sample i is with the context mode r, and θ is the parameter associated with $u_{r,i}$. Each $u_{r,i}$ along with the corresponding q_r models one aspect of context when given

the subject hyperplane. The combination of R modes, each of which is composed of one $\{u_{r,i}, q_r\}$, forms a natural multi-mode structure of the context model.

By defining $w_i = [w_0; u_{1,i}q_1; \cdots; u_{R,i}q_R]$ as the sample specific hyperplane which consists of the subject task model and R modes of context model parameters, and the sample feature as $x_i = [x_i^f; x_i^c, \cdots, x_i^c]$, we obtain the sample scoring function (5.7), and the margin γ_i for sample x_i can be derived as in (5.8):

$$\begin{aligned} f(x_i, w) &= w_i^T \cdot x_i + b; \qquad (5.7) \\ &= w_0^T x_i^f + \sum_{r=1}^{R} u_{r,i} \cdot (q_r^T x_i^c) + b; \\ \gamma_i &= y_i(w_0^T x_i^f + \sum_{r=1}^{R} u_{r,i} \cdot (q_r^T x_i^c) + b). \qquad (5.8) \end{aligned}$$

Here, we model the log probability $\log p(y|x^f, x^c)$ with the sample scoring function $f(x_i, w)$. These two equations well show the more insightful meaning of the contextualized SVM formulation:

- The adaptive hyperplane w_i is the combination of the subject hyperplane w_0 and R rectifications via $\{u_{r,i}, q_r\}$'s with the corresponding contributions determined by the context feature x_i^c. Intuitively, we can treat $u_{r,i}$ as a switch to determine whether the context should be activated while the value $q_r^T x_i^c$ determines how to rectify w_0.

- Motivated by probabilistic motivation (5.3), the $\{u_{r,i}\}$ and $\{q_r\}$ collaboratively describe one mode of the context model. $\{u_{r,i}\}$ serves to judge the discrimination ambiguity of x_i^f, and $\{q_r\}$ is utilized to integrate the context feature x_i^c for the classification of the samples with different ambiguities. The combination of R modes, each of which is composed of one $\{u_{r,i}, q_r\}$, enables the context model to approximate complex decision boundary.

We can formulate the Context-SVM as a max-margin optimization problem with

the margin described as the average of the rectified individual margins related to $\|w_i\|$'s, namely,

$$\begin{aligned} \min_{w_0,\{q_r\}} \ & \frac{1}{2N}\sum_{i=1}^{N}\|w_i\|_2^2 + C\sum_{i=1}^{N}\xi_i, \\ s.t. \quad & y_i(w_i^T x_i + b) - 1 + \xi_i \geq 0,\ \xi_i \geq 0,\ \forall i, \end{aligned} \tag{5.9}$$

where C is a tunable parameter for balancing two items and ξ_i are relaxation parameters.

Optimization for Context-SVM

The formulation can be further compiled with respect to $\{q_r\}$ and w_0 as:

$$\begin{aligned} \min_v \quad & \frac{1}{2N}\sum_{i=1}^{N} v^T U_i^T U_i v + C\sum_{i=1}^{N}\xi_i, \\ s.t. \quad & y_i[(U_i v)^T x_i + b] - 1 + \xi_i \geq 0, \quad \xi_i \geq 0, \quad \forall i, \end{aligned} \tag{5.10}$$

where we can set the matrices $U_i = diag([I_n, u_{1,i}I_{n^c}, \cdots, u_{R,i}I_{n^c}])$, $v = [w_0; q_1; q_2; \cdots; q_R]$ with the instantiated $\{u_{r,i}\}$. I_n is an $n \times n$ identity matrix; n and n^c are the dimension of subject and context feature separately.

Then the first part of the objective can be translated as:

$$\frac{1}{N}\sum_{i=1}^{N} v^T U_i^T U_i v = v^T \sum_{i=1}^{N}\frac{1}{N}U_i^T U_i v = v^T M v. \tag{5.11}$$

Here M is a symmetric positive matrix and can be uniquely factorized as $M = F^T F$ using Cholesky decomposition. Set $z = Fv$. Then the objective function turns to:

$$\min_{w_0,\{q_r\},\xi} \frac{1}{2N}\sum_{i=1}^{N}||w_i||_2^2 + C\sum_{i=1}^{N}\xi_i = \min_{z,\xi}\frac{1}{2}||z||_2^2 + C\sum_{i=1}^{N}\xi_i. \tag{5.12}$$

We can set $\hat{x_i} = (F^{-1}U_i^T x_i)$ with $U_i v = U_i F^{-1} z$. Then overall objective function

with constraints can be defined as follows:

$$\begin{aligned} \min_{z,\xi} \quad & \frac{1}{2}||z||_2^2 + C\sum_{i=1}^{N}\xi_i, \\ s.t. \quad & y_i(z^T\hat{x_i} + b) - 1 + \xi_i \geq 0, \quad \xi_i \geq 0 \ \ \forall i, \end{aligned} \tag{5.13}$$

which can be optimized by traditional SVM solvers. Once we get the z, $v = F^{-1}z$ and $u_{r,i}, q_r, w_0$ are obtained by selecting corresponding elements from v.

Note that in this optimization problem, there are only $(R \times m + n)$ parameters to optimize, and generally R is small. Therefore the overfitting issue can be well alleviated. Eqn. (5.10) has been converted to a standard SVM problem and its solution can be derived with standard SVM solvers, e.g. LibSVM [97].

5.3.3 Ambiguity Modeling

In this subsection, we describe two methods to instantiate the $\{u_{r,i}\}$. As aforementioned, $\{u_{r,i}\}$ is the ambiguity selection function used to identify the ambiguity samples around the classification hyperplane so that finer classification is possible with the context feature. The flexible nature of Context-SVM allows us to instantiate the $\{u_{r,i}\}$ with multiple choices. Here we list two methods which we use in our experiments to instantiate the ambiguity selection function. The first one is the **Linear Scaling Instantiation** (LSI) which uses two linear scaling functions to select the ambiguity samples. The second one takes the estimation error of the original hyperplane as the ambiguity degree and then an **Ambiguity-guided Mixture Model** (AMM) is learned. The corresponding $\{u_{r,i}\}$ severs as a context mode selection function at the decision boundary.

Linear Scaling Instantiation

As aforementioned, we design $\{u_{r,i}\}$ to highlight samples which are classified ambiguously with their subject features $\{x_i^f\}$. Practically, we instantiate $\{u_{r,i}\}$ as a

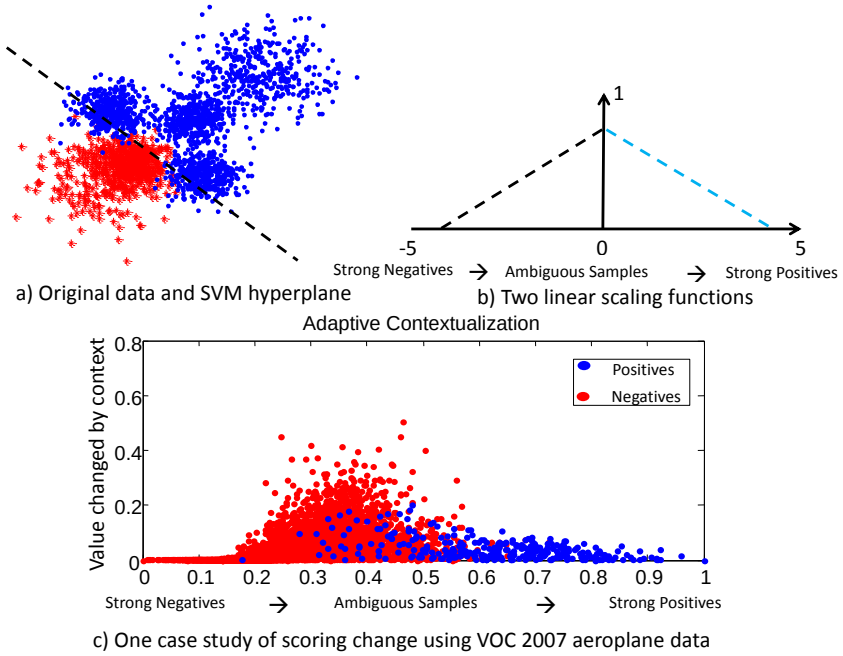

Figure 5.2: Illustration of Linear Scaling Instantiation. a) The sample data with SVM hyperplane, red and blue dots representing positive and negative samples. b) The linear scaling functions. The black and blue dashed lines represent two different scaling functions. Each function scales one part of SVM scores with the range of $[0, 1]$. c) Illustration of the relationship between original sample confidence and confidence variation amount from context. The blue and red dots represent positive and negative samples respectively. The x-axis denotes the sample confidence in subject feature space and y-axis denotes the absolute amount of confidence changed by the contextualization procedure. The confidences are converted into probabilistic values within $[0, 1]$ indicating strongest negative and positive decisions respectively. For better viewing, please see original color PDF file.

set of scores with a learned hyperplane w_0 in subject feature space by traditional SVM:

$$u_{r,i} = \alpha_r w_0^T x_i^f + \beta_r, \ r = 1, 2, \cdots, R. \tag{5.14}$$

Intuitively, for $\alpha_r > 0$, if we set α_r and β_r properly such that all $\{u_{r,i}^T\}$ are within $[0, 1]$, those samples classified as negative by w_0 with high confidences shall be suppressed, namely their corresponding values of $\{u_{r,i}\}$ being small. At the same

time, for $\alpha_r < 0$, if we set α_r and β_r properly such that all $\{u_{r,i}^T\}$ are within $[0, 1]$, those samples classified as positive by w_0 with high confidences shall be suppressed, namely their corresponding values of $\{u_{r,i}^T\}$ being small. Therefore we can sample multiple combinations of α_r and β_r, and both strong negative and positive samples shall be suppressed by $\{u_{r,i}\}$ such that the samples with ambiguous decisions by w_0 are highlighted.

More complicated $\{u_{r,i}\}$ with larger R may derive better ambiguity modeling but may also lead to overfitting. Our empirical study shows that it is a good trade-off by setting $R = 2$, i.e. using two auxiliary functions $u_{1,i}$ and $u_{2,i}$ where $\alpha_1 > 0$ and $\alpha_2 < 0$. Then the combination of $u_{1,i}$ and $u_{2,i}$ can provide a rough yet efficient judgement for the decision ambiguity of a sample and force the context model to concentrate on the samples with large ambiguities.

We illustrate one exemplar contextualization result by Context-SVM on object classification task of the "aeroplane" category in Figure 5.2. This figure shows the adaptive contextualization with respect to the sample ambiguity: the output of the samples with higher ambiguities (i.e. samples lying in the middle of the figure) are changed (absolute difference value of the pre and after contextualization) largely by the contextualization procedure while the well-classified samples (i.e. samples lying on the two sides of the figure) are nearly unaffected.

Ambiguity-guided Mixture Model

The flexibility of $\{u_{r,i}\}$ enables us to create the more complex context model near the classification boundaries. In the subject feature space, the ambiguous areas may be distributed in multiple localized areas and those areas naturally generate different modes. Thus an Ambiguity-guided Mixture Model is necessarily learned to describe this ambiguity distribution. The local classifiers are then placed in areas with high ambiguity. We first define the ambiguity degree a_i of a sample i as the hinge loss from the subject classification model:

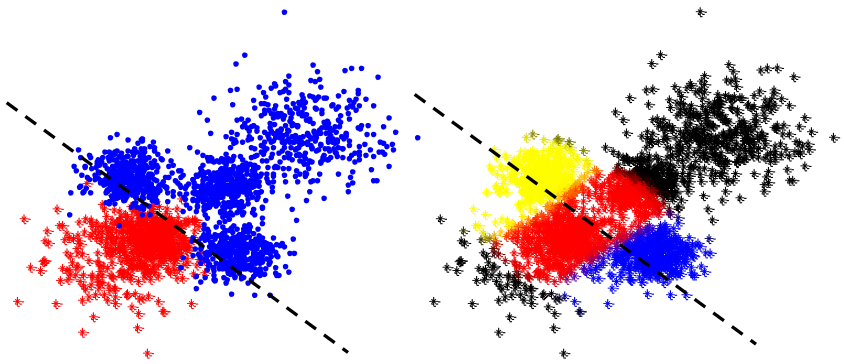

Figure 5.3: Illustration of the Ambiguity-guided Mixture Model (AMM) on a toy problem. The left figure shows the original data. The red and blue dots represent the positive and negative samples. The linear SVM hyperplane is illustrated by the black dashed line. The right figure shows the AMM model with three mixtures (yellow, red and blue). It can be seen that the three mixtures are spreading over the hyperplane where the most ambiguous samples exist. The black dots represent the confidence samples which may not require the context model.

$$a_i = max(0, 1 - y_i(w_0^T x_i^f + b)). \tag{5.15}$$

We propose the ambiguity-based mixture model for modeling the ambiguity distribution of the data. It is a mixture of R Gaussians, with each mixture component normally distributed as $N(\Sigma^r, \mu^r)$ with prior π^r, mean m^r and covariance matrix Σ^r. Assuming the parameter of the mixture model is ρ, the (combined) distribution function $p(x_i|\rho)$ at a particular sample x_i is the mixture probability. Obviously, the local classifiers should be placed near the decision boundary, where classification is the most difficult. Consequently, the mixture should have a high responsibility for areas with high uncertainties. In other words, $p(x_i|\rho)$ should be large when a_i is large, and vice verse. To achieve this goal, we maximize the following objective function:

$$F(a, X|\rho) = \sum_{i=1}^{N} a_i \log p(x_i^f|\rho). \tag{5.16}$$

We use Expectation-Maximization (EM) to solve this objective.

E-Step

$$p(r|x_i^f) = \frac{\pi_r p(x_i^f|r, \rho)}{\sum_{r=1}^{R} \pi_r p(x_i^f|r, \rho)}. \tag{5.17}$$

M-Step

$$\mu^r = \frac{\sum_{i=1}^{N} a_i p(r|x_i^f) x_i^f}{\sum_{i=1}^{N} p(r|x_i^f) a_i}, \tag{5.18}$$

$$\Sigma^r = [\sum_{i=1}^{N} \frac{p(r|x_i^f) a_i}{\sum_{i=1}^{N} p(r|x_i^f) a_i} (x_i^f - \mu^r)(x_i^f - \mu^r)^T]^{-1}, \tag{5.19}$$

$$\pi^r = \frac{\sum_{i=1}^{N} p(r|x_i^f) a_i}{\sum_{r=1}^{R} \sum_{i=1}^{N} p(r|x_i^f) a_i}. \tag{5.20}$$

We can then optimize the parameters of the mixture model iteratively until convergence. Then the $u_{r,i}$ is defined as the posterior probability of each mixture, i.e. $u_{r,i} = p(r|x_i^f)$. In practice, we notice that the dimensionality of x_i^f is often very high. The mixture model built upon this can be inaccurate. Thus we use Principle Components Analysis (PCA) [98] to reduce the dimensionality, e.g. 512, while keeping the majority of data covariance.

We illustrate the concept of the Ambiguity-guided Mixture Context Model (AMM) on a toy problem in Figure 5.3. The red and blue dots on the left figure represent the positive and negative samples. The linear SVM hyperplane is illustrated by the black dashed line. It is obvious that linear SVM cannot get perfect separation on this data distribution. AMM models the ambiguity weighted data distribution. Each mixture describes one local ambiguous area without considering the data distribution of the most confident samples. Thus the learned context model forming the localized classifier can better separate the data. The right figure shows the

AMM model with three learned mixtures (yellow, red and blue). It can be seen that the three mixtures are spreading over the hyperplane where the most ambiguous samples exist. The black dots represent the confident samples which will not utilize the context model.

5.3.4 Kernel Extension

For many visual understanding problems, image descriptors are further encoded as similarity measurements or kernel matrices, and there is no explicit vector representation for each image. Therefore, it is necessary to generalize the Context-SVM formulation to the case with only kernel matrices available. It is worth noting that we only consider the subject feature in the kernel space. The context feature mentioned in this work is with low dimension and thus kernelization is not necessary. We consider the problem in a feature space $\mathcal{F}$ induced by a certain nonlinear mapping function $\phi : \mathbb{R}^n \to \mathcal{F}$. For a properly chosen ϕ, an inner product $\langle \cdot, \cdot \rangle$ can be defined on $\mathcal{F}$ which induces a Reproducing Kernel Hilbert Space (RKHS). More specifically, $\langle \phi(x_i^f), \phi(x_j^f) \rangle = \mathcal{K}(x_i^f, x_j^f)$ where $\mathcal{K}(\cdot, \cdot)$ is a positive semi-definite kernel function.

The context-adaptive scoring function for each sample can be defined as:

$$f(x_i, w_0) = w_0^T \phi(x_i^f) + \sum_{r=1}^{R} u_{r,i} \cdot (q_r^T x_i^c) + b = 0, \tag{5.21}$$

which is similar to (5.7).

By Representer Theorem [99], w_0 can be expressed as linear combinations of $\{\phi(x_i^f)\}$. Thus, there exist sets of coefficients such that $w_0 = \sum_{i=1}^{N} \alpha_i \phi(x_i^f)$. Let $\alpha = [\alpha_1, \cdots, \alpha_N]^T$ and $\Phi(\mathrm{X}^f) = [\phi(x_1^f), \cdots, \phi(x_N^f)]$. Then, the scoring function can be expressed as:

$$\alpha^T \cdot K(:, i) + \sum_{r=1}^{R} u_{r,i} \cdot (q_r^T x_i^c) + b = 0, \tag{5.22}$$

where K is the kernel matrix with $K_{ij} = \langle \phi(x_i^f), \phi(x_j^f) \rangle$ and $K(:, i)$ is the i-th column

vector of the matrix K.

Then the formulation can be compiled with respect to $\{q_r\}$ and α_0 as:

$$\begin{aligned} \min_c \quad & \frac{1}{2N}\sum_{i=1}^{N} c^T B_i^T B_i c + C\sum_{i=1}^{N}\xi_i, \\ s.t. \quad & y_i[(B_i c)^T t_i + b] - 1 + \xi_i \geq 0, \quad \xi_i \geq 0, \quad \forall i, \end{aligned} \tag{5.23}$$

in which we define $B_i = diag([I_N, u_{1,i}I_{n^c}, \cdots, u_{R,i}I_{n^c}])$, $c = [\alpha; q_1; q_2; \cdots; q_R]$, $t_i = [K(:,i); x_i^c, \cdots, x_i^c]$ and I_N is an $N \times N$ identity matrix. The main differences between the kernel version and the linear version include: 1) the original subject feature vector x_i^f is replaced by the column vector of the kernel matrix K, and 2) the formulation in Eqn (5.23) is similar with (5.10). Thus, the same optimization approach can be used for solving the kernel extension of Context-SVM.

5.4 Application: Contextualizing Object Detection and Classification

Algorithm 1 Contextualizing Classification and Detection

Input:
$M_{det}(0)$: Initial object detection model,
$M_{cls}(0)$: Initial object classification model,
$\{I_i\}$: Training images,
For $t = 1, 2, \ldots, T_{max}$

1. Extract detection features and context for each image,

$$\begin{aligned} x_i^f(t) \quad &\leftarrow \quad \text{extract}(I_i), \ \forall i, \\ x_i^c(t) \quad &\leftarrow \quad \text{eval}(M_{cls}(t-1), I_i), \ \forall i. \end{aligned} \tag{5.24}$$

2. Instantiate $\{u_{r,i}\}$ with $\{\{x_i^f(t)\}, R\}$ and $M_{det}(t-1)$.

3. Learn $M_{det}(t)$ via Context-SVM on $\{x_i^f(t), x_i^c(t)\}$.

4. Similarly, learn $M_{cls}(t)$ via Context-SVM by using the outputs from $M_{det}(t-1)$ as context.

EndFor
Output $M_{det}(T_{max}), M_{cls}(T_{max})$.

In this section, we apply the Context-SVM to contextualize two prevalent tasks of image understanding, namely object detection and classification.

5.4.1 Initializations

The initial object detection and classification models $M_{det}(0)$ and $M_{cls}(0)$ for the first iteration are learned based on the state-of-the-art algorithms. For VOC dataset, we follow the part-based model proposed by Felzenswalb et al. [19] for the initial detection model training. The Histogram of Gradient (HOG) [3] and Local Binary Pattern (LBP) [5] features are used for object description and the number of part models for each object category is set as 6. For SUN09 dataset, we use the newly proposed EMAS [96] object detection method due to its efficiency dealing with large number of categories.

For the object classification task, the traditional Bag-of-Words (BoW) model [6] is employed. We first extract the low-level features including SIFT and its color variants [100], LBP and HOG by dense sampling strategy in three scales. Each image is represented by BoW model with spatial pyramid matching [7]. The kernel function is based on χ^2 distance for each type of features, and then all kernels are combined as an average kernel for kernelized Context-SVM.

5.4.2 Iterative Mutual Contextualization

The detailed algorithm for contextualizing object detection and classification by iterative Context-SVM is listed in Algorithm 1. At the t-th step, the context features of one task are the summarized outputs by evaluating the $(t-1)$-th model of the other task on the training data $\{I_i\}$. We use cross validation method to obtain context from object classification in (5.24) as kernel model is easy to overfit on its training data. Hence we use 10-fold of training data and evaluate each fold via the model trained on all other folds.

More specifically, the context features for both tasks refer to the probabilities

that the object categories exist in the image. Thus the context feature values are within $[0, 1]$ and the dimension of context feature vector is the number of object categories. The context from the object classification task is obtained by converting classification scores on the training set to probabilities via sigmoid scaling. The context features from the object detection task are obtained by converting the detected highest score for each object category to the probability in the same manner as for object classification. If there is no object detected for a certain category, the corresponding entry in context feature vector is set as 0.

We instantiate $\{u_{r,i}\}$ based on the extracted subject features and the learnt model from the previous step. For **Linear Scaling Instantiation**, we use two linear functions to model the ambiguity, i.e. $R = 2$. One function is used to suppress the strong positive samples and the other is used to suppress the strong negative samples. For **Ambiguity-guided Mixture Model**, all the raw features are first reduced to 512 dimensions using PCA. Then the ambiguity degree a_i is obtained from the baseline models. A mixture model with $R = 20$ is constructed for each class.

Then we can proceed to conduct Context-SVM based on $\{u_{r,i}\}$, subject features and the corresponding context features for all training images.

5.5 Experiments

5.5.1 Datasets and Metrics

The PASCAL Visual Object Classes Challenge (VOC) datasets [18] are widely used as testbeds for evaluating algorithms for image understanding tasks, and provide a common evaluation platform for both object classification and detection. These datasets are extremely challenging since the objects vary significantly in size, view angle, illumination, appearance and pose. We use PASCAL VOC 2007 and 2010 datasets for experiments in this chapter.

VOC 2007 and VOC 2010 datasets contain 20 object classes with 9,963 and 21,738 images respectively. The two datasets are divided into "train", "val" and "test" subsets, i.e. 25% for training, 25% for validation and 50% for testing. The annotations for the whole dataset of VOC 2007 and "train", "val" set of VOC 2010 are provided while the annotations for "test" set of VOC 2010 are still confidential and can only be evaluated on the web server with limited trials. The employed evaluation metric is *Average Precision* (AP) and mean of AP (mAP) complying with the PASCAL challenge rules.

We also use the SUN 09 dataset [80], which contains 4,367 training images and 4,317 testing images, for object classification and detection evaluation of 107 object categories. SUN 09 [80] has been annotated using LabelMe[33]. The author also annotated an additional set of 26,000 objects using Amazon Mechanical Turk to have enough training samples for the baseline detectors [19]. In the SUN09 dataset, the average object size is 5% of the image size, and a typical image contains seven different object categories while the average PASCAL VOC bounding box occupies 20% of the image. These classes span from regions (e.g., road, sky, buildings) to well defined objects (e.g., car, sofa, refrigerator, sink, bowl, bed) and highly deformable objects (e.g., river, towel, curtain). The employed evaluation metric is *Average Precision* (AP) and mean of AP (mAP) following [80].

In the following experiments, we first evaluate the mutual contextualization capability for ContextSVM with different ambiguity modelings (i.e. ContextSVM_LSI and ContextSVM_AMM) using VOC 2010 "train/val" dataset (i.e. "train" set for training and "val" set for test) for both object classification and detection tasks for proof of concept and ease of parameter tuning. The iterative performance boosting is demonstrated in Section 5.5.3 on the VOC 2010 trainval/test dataset. Then several traditional methods for contextualizing object detection and classification are compared with our iterative Context-SVM on the VOC 2010 trainval/test dataset in Section 5.5.4. Finally, we evaluate the optimal configuration of our method on

Table 5.1: The results of ContextSVM and its baseline for object detection and classification tasks on VOC 2010 train/val. One iteration of ContextSVM is performed with two different ambiguity modeling methods, i.e. LSI and AMM. The relative improvement of mAP over the baseline without contextualization is also listed.

	Classification			Detection		
	Baseline	CtxSVM_LSI	CtxSVM_AMM	Baseline	CtxSVM_LSI	CtxSVM_AMM
plane	86.9	**89.2**	89.2	46.6	50.2	**51.3**
bike	59.1	72.8	**73.0**	48.0	49.3	**50.5**
bird	61.7	64.7	**66.1**	9.8	16.8	**17.2**
boat	68.3	72.5	**73.4**	6.8	11.6	**11.8**
bottle	29.9	**49.9**	49.4	25.6	27.0	**27.5**
bus	82.9	87.4	**87.9**	54.0	55.4	**58.8**
car	63.3	77.3	**78.1**	38.5	39.8	**40.9**
cat	70.1	74.3	**75.4**	26.9	**36.7**	35.0
chair	55.2	62.5	**63.3**	14.8	16.4	**16.9**
cow	36.4	40.1	**40.8**	12.9	17.7	**20.5**
table	50.6	49.6	**50.9**	14.9	19.8	**17.6**
dog	53.4	58.5	**59.3**	15.6	23.1	**23.5**
horse	53.7	66.3	**69.0**	37.6	41.0	**41.0**
motor	64.1	73.1	**74.8**	41.7	44.4	**46.3**
person	84.5	91.4	**91.7**	42.1	**45.6**	45.4
plant	36.1	41.8	**46.4**	6.5	**11.1**	10.1
sheep	60.1	64.8	**65.5**	29.4	**32.6**	29.7
sofa	49.1	52.8	**53.6**	22.3	**30.2**	28.3
train	79.2	84.7	**85.2**	36.5	39.3	**42.5**
tv	66.2	**70.6**	69.9	36.4	**38.3**	38.1
mAP	60.5	67.2(+11.07%)	**68.1(+12.56%)**	28.3	32.3(+14.13%)	**32.7(+15.55%)**

PASCAL VOC 2007, 2010 trainval/test datasets and SUN09 and compare with the state-of-the-art performance ever reported.

5.5.2 Mutual Contextualization

We first give the quantitative results for Context SVM on VOC 2010 train/val dataset in Table 5.1 with one iteration setting. The improved results for object classification and detection tasks demonstrate the effectiveness of Context SVM.

For VOC 2010 classification task, we obtain the mAP of 0.681, a relative improvement of 12.56% over the classification baseline (0.605), with the context information from the detection raw results. The classification result shows the most improvement at those categories which often occupy small amount of the image space, e.g. bottle, tvmonitor, etc. We list some sample images improved by the contextualization as shown in Figure 5.4. There are two rows showing the confidence change before and after the contextualization. The confidence has been normalized to $[0, 1]$. It is worth noting that the large changes are with those ambiguity samples whose original confidences are close to the 0.5. For example as shown in the first row, the third column of Figure 5.4, the motorbike image has been classified with a confidence value of 0.41, and then the detection has a positive response within this image, so the final contextualized classification score for motorbike is very high. The contextualization for the classification task shows that the detection can be utilized to increase the recall rate of classification since the local model used by the detection task can find the objects occupying small part of the images.

For VOC 2010 detection task, we obtain the mAP of 0.327, a relative improvement of 15.55% over the detection baseline (0.283), with the context information from the classification results. The detection result shows the most improvement at those categories which often occupy large amount of the image space with large appearance variance, e.g. dogs, tables, etc. We list some sample images improved by the contextualization in Figure 5.5. The role of the classification context model for

detection tasks is mainly reflected by the fact that (1) the detection often fails for those samples with large appearance variance and the classification model is better to model the appearance changes, and (2) the local model used by detection tasks generally has no scene level context. In those two cases, the classification context model can help (1) to identify those objects with better appearance modeling and (2) to eliminate those false alarms by using the high level global context model. For example, as shown in the first row, the first two columns of Figure 5.5, the classification context model helps to eliminate the false alarm detection of "tvmonitor" and further localize the true positive detection of "table".

Ambiguity Modeling Comparison: We give the quantitative results using different ambiguity modeling functions, i.e. **Linear Scaling Instantiation (LSI)** and **Ambiguity-guided Mixture Model (AMM)**. As shown in Table 5.1, both of these methods outperform the baseline methods with a large margin. Especially, AMM works better in terms of mAP. However, AMM does not outperform LSI at all 20 classes. Another observation is that for those classes with low AP accuracy, AMM performs similarly with LSI. It is reasonable since in that case the ambiguity modeling itself is not accurate. An analysis of AMM and LSI is as follows:

- The ambiguity modeling of LSI largely depends on the baseline prediction. It linearly scales the confidence obtained by the baseline and assigns higher values of $\{u_{r,i}\}$ to those ambiguous samples and lower values to those strong negative or positive samples. Then the learned context model q_r will act correspondingly.

- Unlike LSI, AMM models the data distribution as well as the baseline estimation. At the training stage of AMM, it incorporates the estimation error into the learning of the mixture model. The AMM learning concentrates on the data distribution of the ambiguous samples so that the learned mixtures better describe the complex decision boundary. The obtained $\{u_{r,i}\}$ corresponds

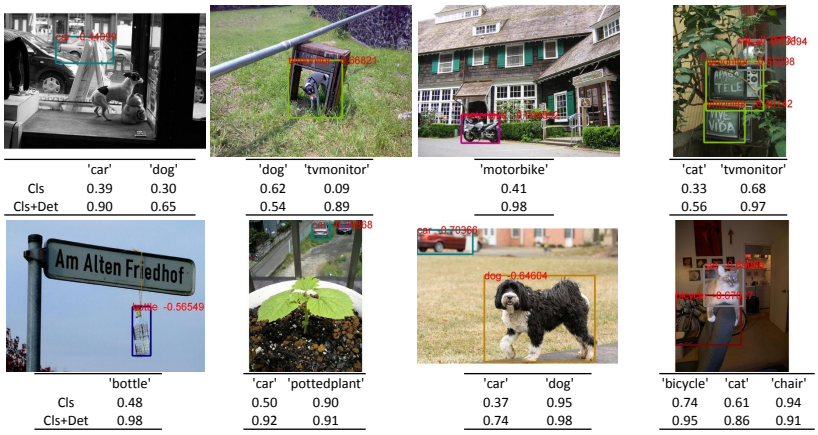

Figure 5.4: Representative examples of the baseline (without contextualization) and Context-SVM for classification task. The classification accuracy is promoted via detection contextualization. The first row of the table below each image shows the classes the image belongs to. The second row is the confidence of the baseline while the third row is the refined result after contextualization. For better viewing, please see original color PDF file.

to the posterior of sample i belonging to mixture r.

- The superiority of AMM over LSI probably comes from that (1) AMM considers the data distribution of ambiguous samples instead of only the baseline prediction in LSI, and (2) the number of mixtures in AMM is much larger than $R = 2$ in LSI. It is straightforward that a larger number of mixtures can better fit to the distribution of decision boundary, i.e. the ambiguous modeling. In all the experiments, we have fixed $R = 20$ in AMM as no obvious improvement can be observed when $R > 20$ from our offline experiments.

The Role of Contextualization: As shown in the results of VOC 2010 train/val dataset, the Context SVM shows great improvement over the baseline for object detection and classification. Through the analysis and the experiments described above, it can be observed that it is necessary to use context for both object classification and detection tasks.

- For object classification, the prevalent methods [15][101] use global features

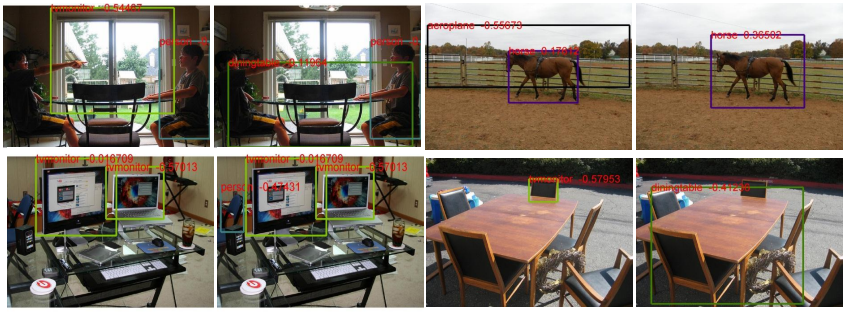

Figure 5.5: Representative examples of the baseline (without contextualization) and Context-SVM for detection task. The detection accuracy is promoted via classification contextualization. The left side image is the result before Context SVM and the right side image is the result after contextualization. For better viewing, please see original color PDF file.

and discriminative modeling to achieve the goal. Although the current state-of-the-art recognition pipeline uses sophisticated feature encoding and learning methods to extract image specific information which often reveals the object-specific contents, e.g. Fisher Vector Coding [14] and SVM classifier [95]. The methods used in classification task are often built with a top-down manner which use global information to infer the existence of a local object. On the other hand, the context from the detection model contains rich local information. It greatly enhances the classifier to learn the image-specific information. As shown in Figure 5.4, a lot of images containing small (or small-sized) objects are re-identified through contextualization.

- For object detection, usually the object detector models the object appearance [96] or object shape [3][19] through the annotated object training samples while discarding the context information defined by the object surroundings. The localized nature of object detector restricts the model to effectively differentiate the false alarm which occurs in those obviously different contexts. The context information from classification model helps to define the context of the object. As shown in Figure 5.5, it is helpful to eliminate the false alarm

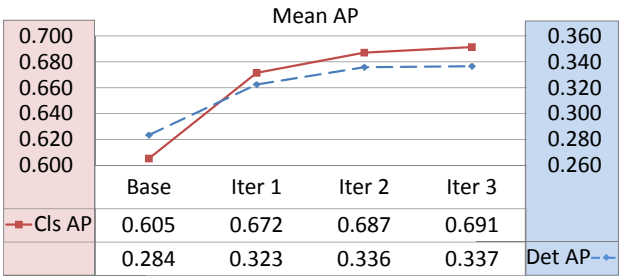

Figure 5.6: Mean AP values of 20 classes on VOC 2010 train/val dataset along iterative contextualization.

and promote the possible true positive.

- The ambiguity modeling enables that the learned context model concerns most on the ambiguous samples. The probabilistic motivation as introduced in Section 5.3.1 implies that it is desirable to learn the joint distribution of subject and the context feature instead of the independent learning as in [17][10]. We propose to use the ambiguity modeling as a bridge between the subject and context task so that joint learning is possible. The learned context model operated on the ambiguous samples is better than the other context modeling method as demonstrated in Section 5.5.4.

- Another key advantage of conducting contextualization for both object classification and detection is that we can further build a more accurate context model with better classifier and detector through mutual contextualization. This step can be iterative until no further useful information can be learned as demonstrated in later Section 5.5.3.

5.5.3 Iterative Performance Boosting

To evaluate the effectiveness of our proposed iterative and mutual contextualization process, we conduct three experiments on VOC 2010 "train/val" dataset. Firstly, we demonstrate the performance improvement measured by mean AP for all the

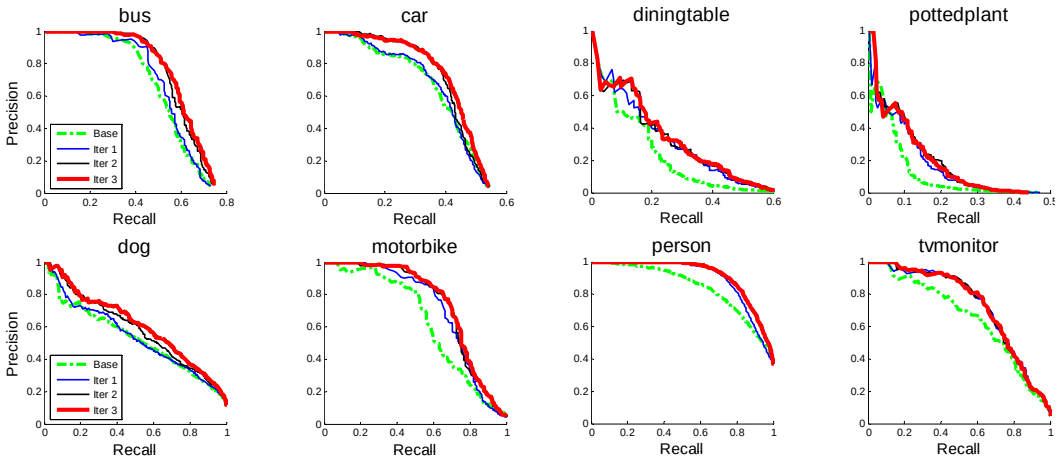

Figure 5.7: Illustration of performance improvement with comparison Precision-recall curves of object detection (upper row) and classification (lower row). The performance of baseline (without contextualization) and those of Context-SVM at iteration 1-3 are plotted.

20 classes in Figure 5.6. In this experiment, the mutual contextualization using LSI is conducted for 3 iterations, and obvious performance improvement is observed for the first and second iteration. As the improvement from the third iteration becomes trivial, we set the maximum iteration number, namely T_{max} to 3 for all the experiments in this work.

In the second experiment, we show exactly how the mutual contextualization process benefits each class by Precision-Recall curves of several representative classes in Figure 5.7, and also show the representative object detection and classification results in Figure 5.8 for the third experiment. As can be observed from Figure 5.7, great performance improvement can be achieved for the first two iterations and in the third iteration, certain amount of improvement can still be achieved for several classes such as "bus" and "dog". From Figure 5.8, it can be observed that the Context-SVM shows good stability in refining the classes even without accurate context such as "pottedplant". The example detection results demonstrate that the improvement of object detection is mainly achieved by effective removal of the ambiguous negatives while the object classification benefits from detection context

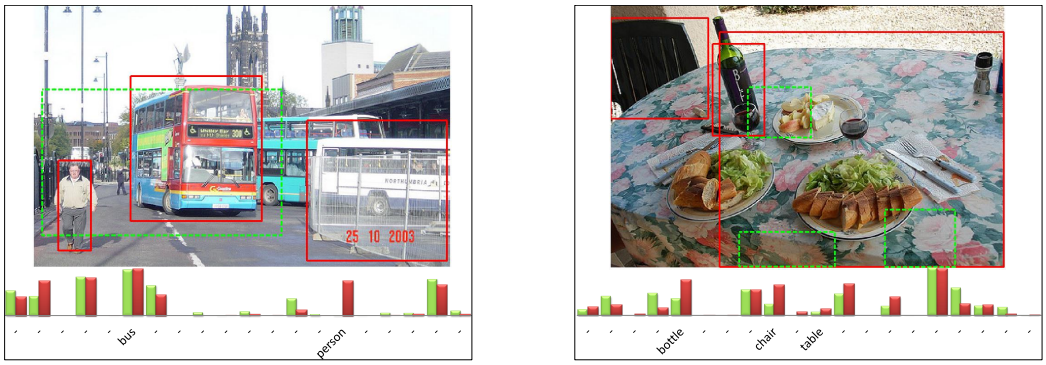

Figure 5.8: Representative examples of the baseline (without contextualization) and Context-SVM at iteration 3. The detections are shown via the detected bounding boxes on images (with proper threshold): the green boxes with dashed lines denote the false alarms from baseline, which are further removed by contextualization and red boxes denote the true detections of both methods. The classification results are compared by the confidences for each object category before (green) and after (red) contextualization. For better viewing, please see original color PDF file.

by calling back those missing objects, e.g. "person" and "chair" missed in the baseline results as shown in Figure 5.8.

5.5.4 Contextualization Methods Comparison

In this subsection, we compare our proposed iterative and mutual contextualization method with other mutual classification and detection contextualization models.

We perform experiments on PASCAL VOC 2010 "trainval/test" dataset and the results are shown in Table 5.2. We compare with the method proposed by Harzallah et al. [17] denoted as **Fuse**, which combines the confidences from several probabilistic models and is the most representative one among those confidence combination approaches [19] [9]. For object classification, Multiple Kernel Learning (**MKL**) [63] method used in [10] is also implemented for comparison, which is a general model fusion method and widely used to combine features in kernel form for object classification. An extra linear kernel is constructed for the context features from the object detection task, and then two kernels are combined with MKL. MKL performs badly for object detection task, and thus we do not report the

Table 5.2: Contextualization method comparison on the PASCAL VOC 2010 (trainval/test) dataset. "Det" and "Cls" respectively denote object detection and classification tasks. Three iterations of ContextSVM has been performed.

	Detection			Classification			
	DetFuse	CtxSVM_LSI	CtxSVM_AMM	ClsMKL	ClsFuse	CtxSVM_LSI	CtxSVM_AMM
plane	50.5	53.1	**54.6**	91.4	90.7	92.2	**92.8**
bike	49.8	52.7	**53.7**	76.6	74.0	77.7	**79.2**
bird	16.0	**18.1**	16.2	66.7	67.2	69.2	**70.9**
boat	10.4	**13.5**	12.5	72.3	73.9	75.7	**78.1**
bottle	30.4	30.7	**31.2**	53.1	53.8	53.5	**54.2**
bus	**54.3**	53.9	54.0	83.7	81.7	84.7	**85.2**
car	43.3	43.5	**44.2**	77.1	74.1	**80.9**	78.9
cat	38.3	**40.3**	40.0	75.3	73.6	76.1	**78.5**
chair	15.9	**17.7**	16.7	62.9	60.9	62.8	**64.4**
cow	30.0	31.9	**32.2**	59.8	59.8	**65.5**	64.5
table	24.1	28.0	**29.1**	57.1	60.5	63.1	**63.2**
dog	23.1	29.5	**30.1**	63.6	62.3	65.6	**68.7**
horse	47.8	52.9	**54.3**	76.5	75.1	79.6	**81.5**
motor	54.2	56.6	**57.2**	81.8	80.2	83.4	**84.5**
person	42.1	**44.2**	43.9	91.2	90.4	91.2	**91.3**
plant	11.8	**12.6**	12.5	44.1	45.8	47.5	**48.4**
sheep	33.5	**36.2**	35.4	64.1	61.7	**71.9**	65.0
sofa	27.5	28.7	**28.8**	48.4	56.0	55.2	**59.5**
train	47.3	50.5	**51.1**	84.0	85.9	86.3	**89.3**
tv	38.8	**40.7**	**40.7**	75.5	76.0	**76.7**	76.0
mAP	34.5	36.8	**36.9**	70.3	70.2	73.0	**73.7**

result of MKL for object detection task here. The main reason is that the context is fixed for all candidate windows within an image and the inaccurate context may severely affect the results for quite many candidate windows. The comparison results show that the proposed iterative and mutual contextualization method outperforms these two traditional contextualization methods for most object categories. We also notice that AMM is consistently better than LSI for object classification task while achieving similar performance on the object detection task.

5.5.5 Comparison with State-of-the-art Performance

We also compare the proposed contextualization method with the reported state-of-the-art object detection and classification approaches on VOC 2007, VOC 2010 and SUN09 datasets. The detailed performance comparison results are listed in Table 5.3, Table 5.6 and Table 5.7.

We compare with the best known VOC 2007 performance from several recent papers in Table 5.3. For object detection, the methods compared include [MIT_2010] by Zhu et al. [102] using latent hierarchical structural learning, [UCI_2009] by Desai et al. [103] using context of object layout, [INRIA_2009] by Harzallah et al. [17] fusing classification scores, and [UoC_2010] by Felzenswalb et al. [19] using part-based model with context of object co-occurrence. For the detection challenge of 2007, our method outperforms 13 classes out of 20 classes and the MAP outperforms the second best [UoC_2010] by 3.6%.

The well-known methods compared for VOC 2007 object classification task are: [INRIA_Genetic] [77], the winner of VOC 2007, [NEC_2010] [15] performing non-linear feature transformation on descriptors, [INRIA_2009] fusing detection scores, and [TagModal] [10] using extra tag information of VOC 2007 dataset. Our method significantly outperforms the competing methods for 12 classes out of 20 classes. Note that our mAP (AMM 0.713) achieves a leading margin by 6.90% to the result of [TagModal](0.667). It well validates the effectiveness of the proposed strategy in utilizing detection context for object classification.

For VOC 2010 dataset, we compare with the released results from the VOC 2010 challenge [18], which are all obtained through the combinations of multiple methods including mutual combination of detection and classification. Necessary postprocessing is also implemented in these methods. Therefore for a fair comparison, we refine the framework used by Chen et al. in their submission [NUSPSL] [105] with the following differences: 1) the combination of detection and classification is further refined by the proposed iterative Context-SVM and 2) we exclude the fusion of

other learning schemes used in [105], e.g. the kernel regression fusing, to verify the effectiveness of the Context-SVM.

The comparison results are shown in Table 5.6, from which we may observe that the classification results from our proposed method outperform the others in 16 classes out of 20 classes, and 6.46% in terms of mean AP over the second best VOC 2010 submission [NLPR_Context]. Note that the submission [NLPR_Context] combines the best-performed detection results in this challenge for classification. Our proposed method also outperforms the winner submission [NUSPSL] in 17 classes out of 20 classes and achieves the highest mean AP even without the fusion with other learning methods. The object detection results from our proposed method based on Context-SVM also outperform 7 classes out of 20 classes, and our method achieves the highest mean AP together with the winner submission [NLPR_Context], which outperforms 6 classes out of 20 classes in this competition.

We also conduct experiments on SUN09 dataset [80]. The 107 classes mAP results on SUN09 dataset for both object classification and detection tasks are listed in Table 5.7. The SUN 09 dataset contains over 200 object categories but only 107 classes are used in [80] since some categories contain insufficient training samples. The baseline detectors of [80] for some objects have poor quality even with additional set of annotations. The current state-of-the-art performance is achieved in [80] which reported 8.55 for detection task and 26.08 for classification. In [80], the authors used a tree-based model to explore the hierarchical context between different objects. Compared with its baseline, the improvement of the TreeContext model is 3.82% (promoted from 7.06 mAP to 7.33) for object detection task and 11.15% (promoted from 19,93 mAP to 17.93) for object classification task. It further incorporates additional global features, i.e. gist feature, and context feature, i.e. location information to achieve the state-of-the-art performance with mAP 8.55 on the detection task. The other top performance is DPMContext which also used different scale and location information as the context feature.

We used the baseline of the EMAS object detector which shows great efficiency for object detection problem with a large number of object categories. EMAS performs better than the DPM [19] with 7.27 mAP for 107 classes while DPM reaches 7.06. The overall detection mAP over all object categories is 8.39 for the LSI instantiation and 8.56 for the AMM instantiation which leads to a 17.74% improvement. Our baseline of object classification has the result of mAP 22.23 which is slightly better than the result of [104]. Using the Context SVM, the performance with AMM instantiation can be boosted to 31.43 which is a 41.39% improvement over the original recognition score. Our implementation shows that we can achieve comparable state-of-the-art result with only the context from the high level task.

5.6 Conclusions

In this chapter, we have proposed an iterative contextualization scheme to mutually boost performance of both object detection and classification tasks. We first propose the Contextualized SVM to seamlessly integrate external context features and subject features for general classification, and then Context-SVM is further utilized to iteratively and mutually boost performance of object detection and classification tasks. The proposed solution is extensively evaluated on both PASCAL VOC 2007, 2010 and SUN09 datasets and achieves the state-of-the-art performance for both tasks.

Table 5.3: Comparison with the state-of-the-art performance of object classification and detection on PASCAL VOC 2007 (trainval/test).

Table 5.4: Detection on VOC 2007

	MIT_ZL [102]	UCI_ICCV09 [103]	INRIA_2009 [17]	UoC_04 [19]	CtxSVM_LSI	CtxSVM_AMM
plane	29.4	28.8	35.1	31.2	38.6	**39.8**
bike	55.8	56.2	45.6	**61.5**	58.7	59.0
bird	9.4	3.2	10.9	11.9	18.0	**18.7**
boat	14.3	14.2	12.0	17.4	18.7	**18.9**
bottle	28.6	29.4	23.2	27.0	**31.8**	30.0
bus	44.0	38.7	42.1	49.1	53.6	**54.2**
car	51.3	48.7	50.9	**59.6**	56.0	57.2
cat	21.3	12.4	19.0	23.1	**30.6**	30.4
chair	20.0	16.0	18.0	23.0	**23.5**	**23.5**
cow	19.3	17.7	**31.5**	26.3	31.1	30.9
table	25.2	24.0	17.2	24.9	36.6	**38.2**
dog	12.5	11.7	17.6	12.9	**20.9**	20.7
horse	50.4	45.0	49.6	60.1	62.6	**63.8**
motor	38.4	39.4	43.1	**51.0**	47.9	48.8
person	36.6	35.5	21.0	**43.2**	41.2	41.5
plant	15.1	15.2	**18.9**	13.4	18.8	18.7
sheep	19.7	16.1	**27.3**	18.8	23.5	23.8
sofa	25.1	20.1	24.7	36.2	41.8	**42.5**
train	36.8	34.2	29.9	49.1	53.6	**54.8**
tv	39.3	35.4	39.7	43.0	**45.3**	44.9
mAP	29.6	27.1	28.9	34.1	37.7	**38.0**

Table 5.5: Classification on VOC 2007

	INRIA_Genetic [77]	SuperVec [15]	INRIA_2009 [17]	TagModal [10]	CtxSVM_LSI	CtxSVM_AMM
plane	77.5	79.4	77.2	**87.9**	82.5	84.5
bike	63.6	72.5	69.3	65.5	79.6	**81.5**
bird	56.1	55.6	56.2	**76.3**	64.8	65.0
boat	71.9	73.8	66.6	**75.6**	73.4	71.4
bottle	33.1	34.0	45.5	31.5	**54.2**	52.2
bus	60.6	72.4	68.1	71.3	75.0	**76.2**
car	78.0	83.4	83.4	77.5	**87.5**	87.2
cat	58.8	63.6	53.6	**79.2**	65.6	68.5
chair	53.5	56.6	58.3	46.2	62.9	**63.8**
cow	42.6	52.8	51.1	**62.7**	56.4	55.8
table	54.9	63.2	62.2	41.4	**66.0**	65.8
dog	45.8	49.5	45.2	**74.6**	53.5	55.6
horse	77.5	80.9	78.4	84.6	**85.0**	84.8
motor	64.0	71.9	69.7	76.2	76.8	**77.0**
person	85.9	85.1	86.1	84.6	91.1	**91.1**
plant	36.3	36.4	52.4	48.0	53.9	**55.2**
sheep	44.7	46.5	54.4	**67.7**	61.0	60.0
sofa	50.6	59.8	54.3	44.3	67.5	**69.7**
train	79.2	83.3	75.8	**86.1**	83.6	83.6
tv	53.2	58.9	62.1	52.7	70.6	**77.0**
mAP	59.4	64.0	63.5	66.7	70.5	**71.3**

Table 5.6: Comparison with the state-of-the-art performance of object classification and detection on PASCAL VOC 2010 (trainval/test).

Detection on VOC 2010

	NLPR [18]	MITUCLA [18]	NUS [18]	UVA [18]	CtxSVM_LSI	CtxSVM_AMM
plane	53.3	54.2	49.1	**56.7**	53.1	54.6
bike	**55.3**	48.5	52.4	39.8	52.7	53.7
bird	**19.2**	15.7	17.8	16.8	18.1	16.2
boat	**21.0**	19.2	12.0	12.2	13.5	12.5
bottle	30.0	29.2	30.6	13.8	30.7	**31.2**
bus	54.4	**55.5**	53.5	44.9	53.9	54.0
car	**46.7**	43.5	32.8	36.9	43.5	44.2
cat	41.2	41.7	37.3	**47.7**	40.3	40.0
chair	**20.0**	16.9	17.7	12.1	17.7	16.7
cow	31.5	28.5	30.6	26.9	31.9	**32.2**
table	20.7	26.7	27.7	26.5	28.0	**29.1**
dog	30.3	30.9	29.5	**37.2**	29.5	30.1
horse	48.6	48.3	51.9	42.1	52.9	**54.3**
motor	55.3	55.0	56.3	51.9	56.6	**57.2**
person	**46.5**	41.7	44.2	25.7	44.2	43.9
plant	10.2	9.7	9.6	12.1	**12.6**	12.5
sheep	34.4	35.8	14.8	**37.8**	36.2	35.4
sofa	26.5	30.8	27.9	**33.0**	28.7	28.8
train	50.3	47.2	49.5	41.5	50.5	**51.1**
tv	40.3	40.8	38.4	**41.7**	40.7	40.7
mAP	36.8	36.0	34.2	32.9	36.8	**36.9**

Classification on VOC 2010

	NLPR_Context [18]	NEC_Nonlin [18]	NUSPSL [18]	CtxSVM_LSI	CtxSVM_AMM
plane	90.3	93.3	93.0	93.1	**93.8**
bike	77.0	72.9	79.0	78.9	**80.5**
bird	65.3	69.9	71.6	73.2	**74.7**
boat	75.0	77.2	77.8	77.1	**78.3**
bottle	53.7	47.9	**54.3**	**54.3**	53.9
bus	85.9	85.6	85.2	85.3	**86.5**
car	80.4	79.7	78.6	80.7	**82.4**
cat	74.6	79.4	78.8	78.9	**80.3**
chair	62.9	61.7	64.5	64.5	**64.9**
cow	66.2	56.6	64.0	68.4	**72.8**
table	54.1	61.1	62.7	64.1	**65.7**
dog	66.8	71.1	69.6	70.3	**73.3**
horse	76.1	76.7	**82.0**	81.3	81.2
motor	81.7	79.3	84.4	83.9	**85.3**
person	89.9	86.8	91.6	91.5	**91.8**
plant	41.6	38.1	48.6	48.9	**50.2**
sheep	66.3	63.9	64.9	72.6	**72.9**
sofa	57.0	55.8	59.6	58.2	**61.6**
train	85.0	87.5	**89.4**	87.8	89.2
tv	74.3	72.9	76.4	76.6	**77.2**
mAP	71.2	70.9	73.8	74.5	**75.8**

Table 5.7: mAP results of 107 classes on SUN09 dataset for both object classification and detection tasks. The relative improvement of mAP over the baseline is also listed.

	Detection	Classification
Baseline_DPM	7.06	17.93
TreeContext [104]	7.33 (+3.82%)	19.93 (+11.15%)
TreeContext+loc+gist [104]	8.55 (+21.10%)	26.08 (+45.45%)
DPMContext [19]	8.34 (+18.13%)	23.79 (+32.68%)
Baseline_EMAS	7.27	22.23
CtxSVM_LSI	8.39 (+15.41%)	30.12 (+35.49%)
CtxSVM_AMM	8.56 (+17.74%)	31.43 (+41.39%)

Chapter 6

Large Scale Object Recognition: Efficient Maximum Appearance Search for Large-Scale Object Detection

In this chapter, we consider the problem of large scale object detection. General objects are believed to be detectable by combining appearance and shape cues. Most current object detection methods focus on shape modeling with rigid/deformable templates, however the study on enhancing the localized object appearance representation is not sufficient. In this chapter, we present an efficient appearance-based object detection model which is very suitable to large scale object detection, especially when there exists a large number of object categories.

We represent the image as an ensemble of densely sampled feature points with the proposed Pointwise Fisher Vector encoding, so that the learnt discriminative scoring function can be applied locally. Consequently the object detection problem is transformed into searching an image sub-area with maximum local appearance

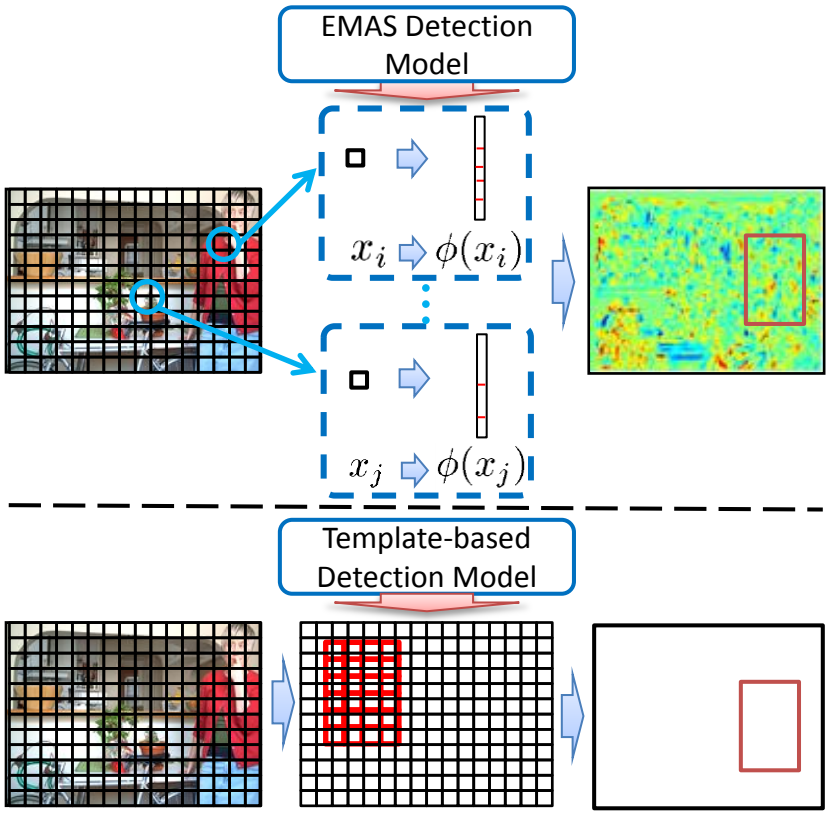

Figure 6.1: Upper part: the proposed EMAS detection. The model inference is operated on the local transformed feature followed by an efficient maximum subarray search. Lower part: the traditional template-based detection.

probability, which has much less complexity than traditional detection methods. The proposed model is suitable to incorporate the global object context with neglectable extra computational cost and multiple feature fusing, which greatly improves the performance in detecting multiple object categories. Our experiments show that the proposed algorithm can perform detection of 1000 object classes in less than one minute on the Image Net ILSVRC2012 datasets and 107 object classes in less than 5 seconds per image on the SUN09 dataset using a single CPU, while achieving comparable performance to state-of-the-art algorithms.

6.1 Introduction

Object detection is a fundamental vision problem which predicts where and which object categories are present in an image. Ongoing research [3, 19, 106, 107, 25, 24] is devoted to developing novel feature representations and classification algorithms as well as designing challenging datasets. To present, the best performing detection models are designed to discriminate object foreground from background on densely sampled sub-windows of images. The discriminative models are normally learnt on a large number of training examples annotated with object bounding boxes.

Most of state-of-the-art object detection methods focus on modeling the object shapes. Among them, the template-based approaches such as the popular Deformable Part Model (DPM) [19] use linear models constructed from a number of part templates of image gradient features. Since templates are sensitive to sampling scale and the pose of objects, inference of such models often entails exhaustively searching for the best template configuration regarding pose, scale, rotation, etc. Refinements to remedy this sampling problem brings extra computation cost, e.g DPM need search the template configuration for best part combinations. Most object detection systems based on the aforementioned methods work at seconds to tens of seconds per object model per image [106, 107], and hence are difficult to be applicable in detecting a large number of object categories.

In this chapter, we propose an Efficient Maximum Appearance Search (EMAS) framework which is efficient and effective in a multi-class object detection. As illustrated in Fig 6.1, we represent the image as an ensemble of densely sampled feature points with the proposed Pointwise Fisher Vector encoding. The learnt discriminative model can be applied to the enriched local representation unlike the state-of-the-art template-based model in which the learned model has to be applied to each testing window exhaustively. Consequently the object detection problem is transformed into searching an image sub-area with maximum local appearance

probability. The overall complexity of the proposed framework is much less than the traditional template-based detection methods as analyzed in Sec. 6.2.1 . The advantage of low computation complexity enables us to explore the large scale object detection problem with huge number of categories. We show in our experiment part that with the large number of categories, the large diversity of samples brings more challenging, our appearance-based approach shows better result than the traditional shape-based approach. Our contributions are the following ones:

- We propose an efficient maximum appearance search model for large scale object detection. Our proposed EMAS appliy the model locally to each transformed local points and the inference problem is transferred to searching the sub-window with maximum sum. As far as we know, this is the first model specifically designed for object detection with large number of categories which is different from the other efficient works concentrating on improving the efficiency of current DPM model [107, 108, 109].

- We propose the Pointwise Fisher Vector coding as the enriched local representation of our detection model. We argue that this local feature coding can enhance the discriminative power of the local feature and model the object appearance in a continuous local feature space. This is the key step to adopt maximum sub-window search and preserve the good performance. This representation can also generate the global feature (context) for the image with negligible cost. Thus it is easy to get the multi-class global classification model which is very useful to form global multi-class context in a large scale setting.

- We show state-of-the-art performance on two challenging datasets with large number of categories, i.e. SUN09 [80] and ILSVRC2012 [34]. Experimental evaluations show that the algorithm can perform detection of 1000 object classes in less than one minute on the Image Net ILSVRC2012 datasets and 107 object classes in about 5 seconds per image on the SUN09 dataset using

single CPU with comparable performance to state-of-the-art algorithms.

6.2 Related Works

6.2.1 General Object Detection

Recent shape-based object detection methods rely on discriminative shape templates using orientation histograms of image gradients. Initially, Dalal and Triggs . [3] used a single rigid template to build a detection model for pedestrians. Thereafter, the PASCAL VOC dataset [18] was released, comprising objects with more deformable shapes like animals and vehicles. Hence the single template model was extended to part-based models [19] by Felzenswalb et al. to handle small shape deformations. Although the deep convolution network [20] shows promising result on ImageNet, the part-based model methods [21, 22, 23] are still the best-performing methods on the practical detection datasets.

Previous research [24, 25, 26, 27, 28] have also explored the BoW model detection. The MKL object detection [24] which uses kernel-based models and spatial pyramid (SP) feature combination achieves promising results but the computation cost is very high. Efficient Subwindow Search (ESS) [25, 26, 27, 28] tries to speed up the VQ-based BoW model using a branch and bound technique but often with much poorer performance on standard datasets. The main disadvantage of VQ is that it encodes the local feature as one specific visual word index, thus no complex local discriminative model can be build upon this.

The BoW-based model has the advantage of efficiency if one linear model can be applied and the possible theoretical computation cost is much less than the template-based approach. Suppose we use the same low level feature for both models, e.g. HOG. For a template model with $m \times n$ cells, we need to compute $m \times n$ times convolution at each pixel for each category test searching over the image. The search complexity is $\mathcal{O}(mnP)$ where P is the searching space complexity for an

image. For a BoW model, the cost is separated into two parts, i.e. the local feature coding step and inference (dot-product) over the linear model. The cost of local feature coding step often increases with the codebook size K which is independent for each categorie. For multi-class object detection, the only cost addition is the inference cost which depends on the sparseness $\mathcal{E}$ of the coding. The sparseness is $1/K$ for hard Vector Quantization (VQ),and is around 3% for Fisher Vecotr coding (FV) [14] in our experiments. So the inference complexity is $\mathcal{O}(\mathcal{E}P)$ which is much less than the template-based approach ($mn \gg \mathcal{E}$).

6.2.2 Feature Encoding

Recent feature encoding approaches, such as Sparse Coding [11] and Locality-constrained Linear Coding(LLC) [12], introduce soft assignment for local feature quantization to substitute previous discrete quantization methods and can been seen as the gentle extension of Vector Quantization. For the recognition problem, these two coding methods benefit from large size codebooks as demonstrated in a recent comparison survey [13]. The large codebook size and the introduction of soft assignment narrow down the quantization error but also bring a lot of computation cost. Lately, the aggregation coding, e.g. Fisher Vector coding or Super Vector coding, demonstrated to greatly improve the discriminative power of local features [13]. Fisher encoding [14] tries to capture the average first and second order differences between local features and centers of a Mixture Gaussian Distributions learnt from general datasets while Super vector encoding [15] only focus on the first order difference. Recently, G. Csurka et.al [16] extend Fisher Vector coding to patch level for the semantic segmentation task and achieves good performance. We propose a pointwise extension and the scoring function is operated on the point level instead of the patch level scoring and back projection used in [16].

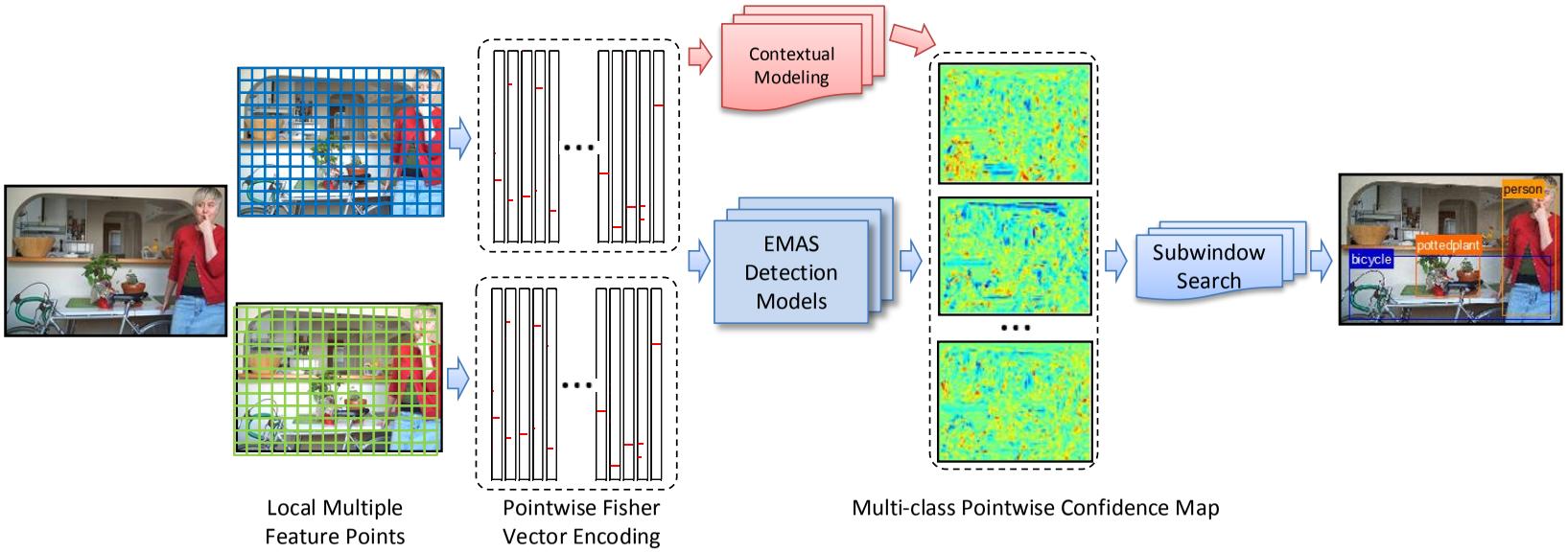

Figure 6.2: Framework illustration of Efficient Maximum Appearance Search.

6.2.3 Efficient Object Detection

In the past few years, various ways to reduce detection time have been explored to decrease the time cost in detection window sampling. The cascade part-based model [107] accelerates the part-based models [19] by learning stagewise thresholds to fast reject negative sampling windows. Some other methods try to improving the efficiency of current DPM model [108, 109]. The jumping windows method [106] generates sparse candidate windows by back-projecting Bag-of-Word image classification scores and assumes objects are more likely to be located by more positive discriminative words. ESS with branch-and-bound search [25] are proposed to reduce the cost in searching subwindow by finding bounds of subwindow scores.

6.3 Model

The proposed Efficient Maximum Appearance Search (EMAS) approach proceeds through four stages to perform large-scale object detection as shown in Fig. 6.2. During the first stage, we extract multiple complementary features; such as HOG, color moments, etc., for an image, these features are then used to encode the image with a pointwise feature representation during the second stage. In the third stage, we obtain the object confidence maps using a combination of appearance detection models and global context models to look for specific objects within a global context. Finally, the object confidence values are combined to find the highly confident object locations for each object category using maximum subarray search. In the following subsections, we explain in more detail the unique points of our approach, namely, the use of probabilistic prediction over a point ensemble, and the representation, model learning and model inference of the EMAS approach. We also extend our model into multi-class categories setting which enables a multi-class object context. Our system can easily adopt multiple feature fusing to boost the performance.

6.3.1 Probabilistic Prediction over Point Ensemble

Similar to Bag-of-Words like models, where the probabilistic prediction is conducted over the word ensemble contained by the inference body, the EMAS approach model also estimates the object probabilities using the point ensemble contained within an image area. In particular, let $P(X) = \prod_{i=1}^{n} p(x_i)$. The binary discriminative model is used for the figure-ground detection for each object category, which try to obtain the discriminative probabilities as:

$$\frac{P(X|l=1)}{P(X|l=-1)} = \prod_{i=1}^{n} \frac{p(x_i|l=1)}{p(x_i|l=-1)}, \tag{6.1}$$

where $l = 1$ denotes the foreground condition and $l = -1$ denotes the background condition. Using the linear discriminative models, e.g. SVM, the logarithm binary discriminative probability can be expressed approximately as:

$$\log(\frac{p(x_i|l=1)}{p(x_i|l=-1)}) = w^T \phi(x_i), \tag{6.2}$$

where w is the linear weighting vector and $\phi(x)$ denotes the feature expression for a single image point x. Therefore, Eqn. 6.1 can be formulated into the logarithm form as:

$$\log(\frac{P(X|l=1)}{P(X|l=-1)}) = \sum_{i=1}^{n} w^T \phi(x_i), \tag{6.3}$$

namely the log-likelyhood of an image area to be an object foreground depends on the sum of the pointwise inference in this area.

6.3.2 Representation: Pointwise Fisher Vector

The performance of the EMAS framework relies heavily on the design of pointwise featrure representation. In this work, we choose to extend the Fisher Vector (FV) feature coding method [14] to derive Pointwise Fisher Vector (PFV) coding. Similar to Fisher Vector coding method, the PFV coding uses a Gaussian mixture models

(GMMs) $U_\lambda(x) = \sum_{k=1}^{K} \pi_k u_k(x)$ trained on local features of a large image set using Maximum Likelihood (ML) estimation to describe image content. The parameters of the trained GMMs are denoted as $\lambda = \{\pi_k, \mu_k, \Sigma_k, k = 1, \cdots, K\}$, where $\{\pi, \mu, \Sigma\}$ are the prior probability, mean vector and diagonal covariance matrix of Gaussian mixture respectively.

For a local feature x_i extracted from an image, the soft assignments of the descriptor x_i to the kth Gaussian components γ_{ik} is computed by $\gamma_{ik} = \frac{\pi_k u_k(x_i)}{\sum_{k=1}^{K} \pi_k u_k(x_i)}$. The PFV for x_i is denoted as $\phi(x_i) = \{u_{i1}, v_{i1}, \cdots, u_{iK}, v_{iK}\}$ while u_{ik} and v_{ik} is defined as follows:

$$u_{ik} = \frac{1}{\sqrt{\pi_k}} \gamma_{ik} \frac{x_i - \mu_k}{\sigma_k}, v_{ik} = \frac{1}{\sqrt{2\pi_k}} \gamma_{ik} [\frac{(x_i - \mu_k)^2}{\sigma_k^2} - 1]. \tag{6.4}$$

where σ_k is the square root of the diagonal values of Σ_k. The representation $\phi(I, y)$ of image area y can also be generated by merging $\phi(x_i)$, i.e. $\phi(I, y) = \sum_{i=1}^{N} \phi(x_i)$. To summarize, we provide a brief analysis of the relationship between FV and PFV coding:

1. PFV extends Fisher Vector Coding [14] to the local feature point level. At each point, the local feature is mapped to GMMs with K Gaussians. The gradient vector with respect to the mean and standard deviation parameters serves as an enriched representation for this local feature. The pointwise representation can also be flexibly merged back to the Fisher Vector global image representation as aforementioned. Compared with VQ, PFV could provide much rich local representation. For VQ, each local feature is mapped to a codebook index while in PFV, x_i is mapped to each GMMs and the gradient vectors enable the local model learning.

2. The pointwise representation $\phi(x_i)$ is sparse since each feature point only has few non-zero GMMs component assignment values γ_{ik}. It means that the

model only needs to be applied to these non-zero components in the inference stage thereby making it very efficient. A statistic from SUN09 shows that each local feature is assigned, on average, to 3.5 GMMs components.

6.3.3 Model Learning

In the training procedure, we assume a series of training samples for one category with bounding boxes window $\{y_1, y_2, ..., y_{n_I}\}$ and corresponding labels $\{l_1, l_2, ..., l_{n_I}\}$. A max-margin formulation is used to learn the linear discriminative model w for each object figure-ground classification. In detail, we formulate the objective function as following:

$$\begin{aligned} w &= \arg\min_{w,\xi} \frac{1}{2}||w||^2 + C\sum_{i=1}^{n_I}\xi_i \\ s.t. \quad & l_i w^T(\frac{1}{Z_i}\sum_{m\in y_i}\phi(x_m^i)) > 1-\xi_i \\ & \xi_i \geq 0, \forall l_i \in \{1,-1\}, \end{aligned} \tag{6.5}$$

where $\phi(x_m)$ is the mth pointwise feature in the image area y and we use the ground truth object area as the positive training samples for $l = 1$ and use image areas which have less than 0.4 overlap ratio to the ground truth object areas as the negatve samples for $l = -1$. Normalization factor Z_i is applied to the sum of the pointwise features in order to fit to the SVM optimization. Hard negative mining is done for 3 rounds to enhance the discriminative capability of the model.

6.3.4 Model Inference

The goal of the EMAS inference step is to find the image area with maximum probability of containing the object,

$$\begin{aligned} \hat{y} &= \arg\max_{y} \log(\frac{P(X, y|l=1)}{P(X, y|l=-1)}) \\ &= \arg\max_{y} \sum_{m=1}^{n} w^T \phi(x_m) \\ &= \arg\max_{y} f(I, y, w) \end{aligned} \tag{6.6}$$

where $\phi(x_m)$ is the mth pointwise feature in the image area y. We denote an appearance-based detection model as $w = \{w_1^u, w_1^v, \cdots, w_K^u, w_K^v\}$ while w_k^u, w_k^v correspond to the weights for coding vector u_{ik}, v_{ik} respectively.

To apply the model on a given image area y in image I, we need to compute its inner product with the global representation of area y, denoted as $\phi(I, y)$. We show the model scoring function can be generated with the PFV representations $\phi(x_i)$ as follows,

$$f(I, y, w) = \sum_{i=1}^{N} \sum_{k=1}^{K} [(w_k^u)^T u_{ik} + (w_k^v)^T v_{ik}], \tag{6.7}$$

Namely, the scoring of an image area can be substituted by computing score sum of the feature points within the area.

To apply model w on the whole image I and detect high-scored areas, we first extract and encode dense and regularly sampled PFVs– $\phi(x_{ij})$, where $\{i \in [1, N_y], j \in [1, N_x]\}$, N_x and N_y are the sampling point numbers in the width and height direction and $N_x \times N_y = N$ is the total PFV number. Then by computing inner product to all PFVs with the model w, we can produce a rectangle score map M_I, where $M_I(i, j) = w^T \phi(x_{ij})$. In this work, we only consider locating object in rectangle areas $y = [t, b, l, r]$ denoted by the top, bottom, left and right coordinate of the rectangle. Consequently the object detection task is converted to the following op-

timization problem regarding the scoring function $f(I, y, w)$ in Equation 6.7. This optimization problem is called 2D maximum subarray sum search:

$$\begin{aligned} \hat{y} &= \arg\max_{y \in \mathcal{Y}} f(I, y, w) \\ f(I, y, w) &= \sum_{i=t}^{b} \sum_{j=l}^{r} M_I(i, j), \end{aligned} \tag{6.8}$$

where $\mathcal{Y}$ is the rectangle window set within image I. This problem has a number of efficient solutions [110, 28] as compared to simple exhaustive search which has a complexity of $\mathcal{O}(N^2)$. We adopt the method in [110, 112], which decomposes the search in one dimension to construct efficient dynamic programming problems and has the complexity of $\mathcal{O}(N^{1.5})$. In our experiment, the solution from [110] takes about several milliseconds to search for one confidence map, and the total subarray search for the 107 object categories of SUN09 [80] dataset costs less than one second on one images. Therefore, the computation cost in this subarray search is not a bottleneck of our proposed framework.

6.3.5 Contextual Detection

In this work, we propose a natural way to embed global contextual detection into our detection framework. As demonstrated in [19, 78], the object detection performance can be greatly enhanced using the knowledge of global context information in a multi-class setting. The global context is normally the probability values describing how likely the image contains certain object categories, which can provide a reference to the detection results. In our contextual detection, we obtain such probability values from global image classifications. We use the normalized Fisher Vector of the whole image (which can be easily produced from the PFVs) as features. Suppose, there are n_c class in the training dataset, we define the context feature for image I as $\phi_{ctx}(I) = \{d_1, c_1, \cdots, c_{n_c}\}$, where c_i are the object existence probability predicted by

the ith global classifier. Then, the contextual scoring function is defined as follows,

$$f(I, y, w) \quad = \quad w^T \phi(I, y) + w_{ctx}^T \phi_{ctx}(I), \tag{6.9}$$

It is worth noting that the contextual detection has several good properties: (1) Stability in the multi-class setting. Normally each context component can depict one attribute of the image, and the weight of the each attribute for detecting certain object can be learned from the training samples. Predictions using additional contextual information is more stable and accurate in problems with large number of object categories and clear object relations. (2) Highly efficient. Defining the global context as the union of classifier outputs is the most efficient way for most recognition frameworks since it requires little additional computation [19, 78]. In our work, the global context can be obtained immediately after running the global image classification.

6.3.6 Multi-Feature Fusing and Spatial Layout

To effectively model the object appearance, multiple features are often used due to their complementary nature, e.g. HOG or SIFT focus for modeling the local shape, Color Moment for modeling local color statistics, and LBP for modeling the local texture pattern. In the EMAS framework, it is easy to fuse multiple features to boost the detection accuracy as well as the effectiveness of the global classification model. We perform independent coding for each kind of local feature. During the training stage, multiple Fisher Vectors are concatenated and fed into the classifier learning. In the testing stage, multiple features are combined into one confidence map which is then searched efficiently.

We also consider addition of spatial constraints, such as Spatial Pyramid Matching (SPM), into our approach, which will certainly improve the detection accuracy. SPM can be easily added by applying more spatially-structured local models and the

maximum subarray search with more complex optimization algorithm. However, at this stage, we concentrate on how to improve the performance with low added-on cost and SPM will bring additional computation cost.

6.4 Efficiency Analysis

The whole detection process contains three steps, i.e. local feature extraction, PFV encoding, model inference. Here we would like to discuss the detailed efficiency analysis of the last two steps.

PFV encoding includes two parts: soft assignment calculation and the pointwise encoding. The soft assignment has $\mathcal{O}(KND)$ complexity, where N is the number of feature points, K is the number of Gaussians in the GMMs and D is the local feature dimension. The pointwise encoding takes $\mathcal{O}(\mathcal{E}(\gamma_{th})ND)$, where $\mathcal{E}(\gamma_{th})$ represents the average number of GMMs assignments with higher probability than threshold γ_{th} for each feature point. In our experiments, we set $\gamma_{th} = 0.01$ and obtain $\mathcal{E} = 3.5$ on the training image set of SUN09 without losing the performance. Hence the overall computation complexity for PFV coding is near $\mathcal{O}(KND)$ which is equal to the prevalently used Vector Quantization (VQ). For a single computation PFV computes exponential values and products and hence may take more time than square distance of VQ. However, the number of Gaussianse K in PFV is only about hundreds which is much smaller than the codebook size in VQ (from thousands to millions) with similar performance. After all, PFV is highly efficient considering both speed and performance.

The computation in the model inference contains three parts: pointwise confidence mapping, maximum subwindow search and contextual detection. For n_c class, the complexity of pointwise confidence mapping is $\mathcal{O}(n_c\mathcal{E}(\gamma)ND)$. It equals to n_c times inner product of the sparse PSV coding vector. And the maximum subwindow search we adopt has the complexity of $\mathcal{O}(N^{1.5})$ as aforementioned. Finally,

Table 6.1: Average running time(s) for 107 classes detection on SUN09.

Total	Fea Extract	PSV Encoding	Model Inference		
			Conf	MaxSearch	Context Det
4.7	0.4	0.7	2.6	0.8	0.2

compared to the other two parts, the contextual detection cost is trivial since it is only $\mathcal{O}(2n_cKD)$ complexity.

To be more clear, we demonstrate an example computation cost for EMAS in a large scale detection task. The task is performed on SUN09 [80] dataset which includes 107 classes. As shown in Tab. 6.1, the total cost for 107 classes detection is about 4.7 seconds on a Xeon 2.67GHZ (single core mode). For one object detector, per category model inference cost is around 0.03 seconds and 3.6 seconds totally for 107 categories. Namely the additional cost for one more detection model is only about 30ms. It proves that the proposed EMAS has high scalability in the number of object categories.

6.5 Experiments

6.5.1 Datasets and Metric

We evaluate our proposed EMAS framework on two popular datasets, i.e. ImageNet ILSVRC 2012 [34] and SUN09 [80]. ImageNet ILSVRC 2012 is a subset of ImageNet containing 1000 categories and 1.2 million images. In these 1.2 million images, more than 544K images are labeled with object bounding boxes. The validation and test data for this competition consists of 150,000 photographs, collected from flickr and other search engines, hand labeled with the presence or absence of 1000 object categories. A random subset of 50,000 of the images with labels is released as validation data included in the development kit along with a list of the 1000 categories. Our main result is conducted on this validation set since the organizer

didn't release the test set annotation after the challenge. The evaluation metric is top5 error rate defined by the ILSVRC organizer.

We also use the SUN 09 dataset introduced in [80] for object detection evaluation of 107 object categories, which contains 4,367 training images and 4,317 testing images. SUN 09 [80] has been annotated using LabelMe[33]. The author also annotated an additional set of 26,000 objects using Amazon Mechanical Turk to have enough training samples for the baseline detectors [19]. These detectors span from regions (e.g., road, sky, buildings) to well defined objects (e.g., car, sofa, refrigerator, sink, bowl, bed) and highly deformable objects (e.g., river, towel, curtain). The employed evaluation metric is Average Precision (AP) and mean of AP (mAP).

6.5.2 Implementation Details

We first normalize the image with the longest edge to 500 pixels. We extract two kinds of low-level features for all the experiments. The first one is dense SIFT feature from VL-Feat [51] using multiple scales setting (spatial bins are set as 4, 6, 8, 10) with 6 pixel step. The second one is the local color comment(CM) proposed in [14]. These two features show great complementary effect in the task of object classification [14]. Each SIFT and CM feature is reduced to $60D$ for noise removal. The number of mixtures in the GMMs model in PSV coding is set to 128 for SUN dataset and 256 for ILSVRC dataset. We sample 500,000 descriptors from the training images of ILSVRC and perform EM to obtain the GMMs. For all experiments, we only output the maximum subwindow for one image per class at testing stage, namely we use a precision-preferred detector. Multiple detections can be obtained by iteratively performing the EMAS on one image. All the experiments are conducted on a Xeon Server with 32GB memory using single core mode.

For model learning, we fix the parameter C of SVM as 1 for all experiments. The hard training constraint is mined with the same way of inference steps except that we restrict the number of output windows to 30 for one image with a further

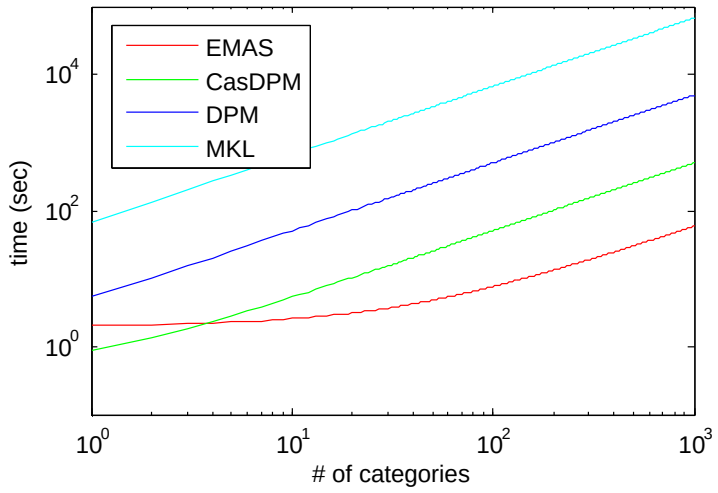

Figure 6.3: Rough cost comparison cost in a multi-class setting.

Non-Maximum Suppression step. The total training process usually takes about half an hour for one class.

6.5.3 Efficiency Comparison

Here, we compare the rough running cost of EMAS with three object detection models in a multi-class setting: 1) Multiple kernel learning for object detection (MKL) [24] using three-stage linear and non-linear detection, 2) Deformable Part Model [19] 3) Cascade DPM [107].

We first perform the full 1000 categories detection on ILSVRC 2012. The average running for one image is 58.4 seconds including 1.9 seconds for feature extraction and feature encoding, 56.5 seconds for 1000 categories model inference. So the added-on cost for each category is 56ms. For CasDPM and DPM, the feature pyramid for both method often takes 375ms, and needs 500ms, 5s respectively for model inference (rough estimate, changes for different setting). The cost for MKL reported in [24] is 67 seconds for one image. We can see the cost simulation for different approaches in Fig. 6.3 in a multi-class setting. It can be observed that our EMAS is not the fastest in the setting of few categories due to the feature encoding step cost. But it shows

Table 6.2: Object classification and detection results on ILSVRC 2012.

	XRCE/INRIA	Oxford_DPM	Oxford_Mix	ISI_CasDPM	EMAS
GMMs size	256	1024	1024	256	**256**
Multi-Fea+SPM	2 fea	2 fea	2 fea	4 fea+SPM	**2 fea**
$error_{cls}$	0.334	0.269	0.269	**0.261**	0.326
$error_{det}$	n.a.	0.529	**0.500**	0.536	0.554
acc_{det}	n.a.	0.644	**0.684**	0.628	**0.662**

at least one order of magnitude faster when the number of categories increases.

6.5.4 Performance Evaluation

Large Scale Object Detection on ILSVRC2012:

ILSVRC2012 is a large challenging dataset including 1000 object categories. We first perform the classification task to obtain the object context. For each category, we train a one-vs-all classifier using an implementation of stochastic dual-form SVM solver [113]. The top 5 error ratio ($error_{cls}$) using two features is 0.326 which is very close to the public result 0.334 from **XRCE/INRIA** in the challenge with similar setting. The result using single dense SIFT feature is 0.380. The complementary effect from CM improves the overall performance. It is worth noting that our performance can be further boosted with large GMM for FV. e.g. **Oxford** gets 0.269 when sets the size as 1024 which is 4 times larger than our implementation. We train our detection using the same SVM solver. The initialization of the detection model is trained using the object feature and a large amount of negative images. 3 round of hard sample mining is utilized.

For detection, we compare our results with the challenging entries [1]: (1)**Oxford_DPM** is the result from DPM detection over baseline classification scores. (2)**Oxford_Mix** used the detection result from DPM and retrain the foreground model with complicated classification model which also is the best result from **Oxford**. (3) **ISI_CasDPM**

[1] `www.image-net.org/challenges/LSVRC/2012/results.html`

is the result using cascade object detection with deformable part models, restricting the sizes of bounding boxes. We show the comparison results on ILSVRC2012 dataset in Tab 6.2. Our detection result $error_{det}$ reaches 0.554 top 5 error rate which is comparable to the DPM and CasDPM while the single feature result using SIFT only is 0.582. Moreover, it is worth noting that the detection result of ILSVRC2012 heavily relies on the performance of classification. Usually, detection will be performed to the top ranked image with high classification confidence, i.e. a combination of two steps: first classifier the right categories and then perform the localization. Thus the error rate can be approximately interpreted as $error_{det} = 1 - (1 - error_{cls}) * acc_{det}$ where the acc_{det} shows the real detection accuracy for each detection model. We show the acc_{det} in Tab. 6.2. It can be seen that our localization ability of our detection model is also comparable to the state-of-the-art model.

Object Detection on PASCAL VOC2007:

The detection results on VOC 2007 are listed on Table 6.3. MPI_ESS [25, 114] is the Efficient Subwindow Search entry participating VOC 2007. It extracted dense grid SURF [115] feature and salient points from the image. A BoW model with 3,000 codebook is constructed. Subsequently, all feature points in train and test images are represented by their coordinates in the image and the ID of the corresponding codebook entry. UOCTTI [18] is the winner of PASCAL VOC2007 using the initial version of the Part Model which was further enhanced in [19].

We first report the EMAS results on the 20 object categories and compare with the previous appearance-based model MPI_ESS. Our raw PFV-based detection performs at the mAP of 16.3%, which is already much higher than 10.1% of MPI_ESS. It does demonstrate that the PFV encoding can well represent the local feature and has much less quantization error than the VQ encoding used in MPI_ESS. Our final EMAS with context refinement further improve the performance to the

Table 6.3: Object Detection results (AP in %) on VOC 2007.

	plane	bike	bird	boat	bottle	bus	car	cat	chair	cow	
UOCTTI_2007 [18]	20.6	36.9	9.3	9.4	21.4	23.2	34.6	9.8	12.8	14.0	
Part Model [19]	28.7	**51.0**	0.6	14.5	**26.5**	**39.7**	**50.2**	16.3	**16.5**	**16.6**	
MPI_ESS [25]	15.2	15.7	9.8	1.6	0.1	18.6	12.0	24.0	0.7	6.1	
EMAS	**33.1**	25.2	**10.6**	**14.9**	4.5	29.0	27.6	**33.8**	1.5	10.1	
	table	dog	horse	motor	person	plant	sheep	sofa	train	tv	mAP
UOCTTI_2007 [18]	0.2	2.3	18.2	27.6	21.3	**12.0**	14.3	12.7	13.4	28.9	17.1
Part Model [19]	24.5	5.0	**45.2**	**38.3**	36.2	9.0	**17.4**	22.8	34.1	**38.4**	**26.6**
MPI_ESS [25]	9.8	16.2	3.4	20.8	11.7	0.2	4.6	14.7	11.0	5.4	10.1
EMAS	**25.9**	**18.6**	21.8	26.9	5.6	9.1	9.2	**23.0**	**35.0**	10.1	18.8

mAP of 18.8% which is comparable to average performance of shape-based models. Moreover, our EMAS framework outperform in 8 out of 20 categories over the Part Model method, and the most competitive performance is on those highly deformable categories, e.g. aeroplane, bird, cat, dog.

Multi-Label Object Detection on SUN09:

SUN09 is a very challenging datasets with rich contextual information. The concerned object categories span from regions (e.g., road, sky, buildings) to well defined objects (e.g., car, sofa, sink, bowl, bed) and highly deformable objects (e.g., river, towel, curtain). We first trained the global object classification model. Each class is trained independently using linear SVM. The mAP of the classifiers is about 29.6% for 107 classes on SUN09 dataset. The classification scores on the training set is obtained by 10-fold cross validation. We perform the proposed EMAS detection model on the 107 classes and compare with the DPM. We use the results of DPM on SUN09 released by the author of [80] which is 7.06% mAP for 107 objects. Further [80] refines this baseline result by modeling the co-occurrence and relative spatial relation of objects with a tree graphical model and obtain the improvement to 8.37% mAP. Our base detector without contextual training obtains 7.26% mAP

Table 6.4: Object detection result on Sun09(AP %).

	plane	bed	bkcase	building	closet	field	floor	grass	mountain	river	
DPM[19]	**35.1**	26.3	2.3	**14.4**	1.1	**19.8**	31.3	11.0	17.2	2.9	
EMAS	12.7	**34.1**	**14.8**	14.3	**12.8**	18.9	**38.1**	**12.3**	**25.6**	**12.4**	
	road	sea	shelves	showcase	sky	sofa	toilet	tree	wall	water	mAP
DPM[19]	33.2	28.7	2.6	0.0	55.3	11.5	**22.0**	10.9	14.7	1.5	17.1
EMAS	**34.9**	**35.0**	**13.6**	**11.9**	**61.9**	**12.7**	11.7	**12.4**	**21.9**	**15.1**	**21.4**

which is slightly better than the result of DPM and we obtain 8.44% mAP with our contextual detection. Our outperformed categories are also on the highly deformable objects. In Section. 6.5.4, we will provide a more comprehensive analysis on this feature of the EMAS framework.

Object Detection with Large Appearance Variance

Our appearance-based model is appealing for object detection with large variation of appearance. Here, we show 20 classes amorphous object detection result from SUN09 and compare with the DPM [19] in Table 6.4. These classes range from 1) regions (e.g. sky, building, road, river) and 2) objects with large shape variation (e.g. bed, sofa, shelves, aeroplane). The EMAS achieves better results. There are some interesting features of EMAS revealed by some example detection shown in Figure 5.8. The model is purely appearance-based, i.e with no shape constraint, thus the algorithm is good at handling truncated/occluded objects (Figure 6.4, 1st row, such as part of cars and bicycles), rare view objects (Figure 5.8, 2nd row, such strange view of cats, sofa, motorbikes) and detecting region objects (Figure 5.8, 3rd and 4th row, such as sky, buildings, trees, floor). But it also causes the problem that it can not distinguish one object from a cluster of objects (e.g. a cluster of horses, cars, cows, shown in Figure 6.4, 5th row).

We show some sample detection results from ILSVRC2012 in Fig. 6.5, the large number of categories creates large diversity in the object categories. It is interesting

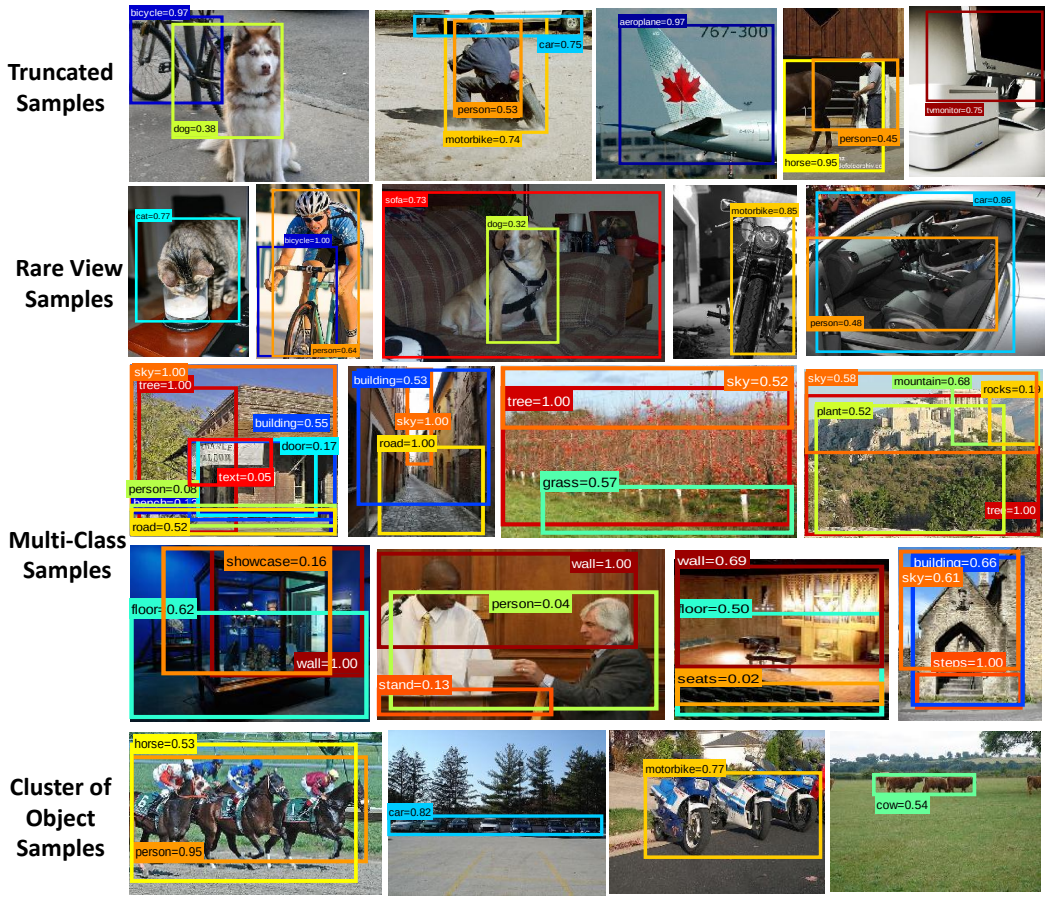

Figure 6.4: Sample results from SUN09

to see that the proposed detector can detect the object in the 1000 categories pool. We plot more results in the supplementary files.

6.6 Conclusion

In this chapter, we aim to do further study on the appearance-based approach with contextual information for the large-scale object detection problem. By means of an advanced coding scheme from a state-of-the-art large scale classification method and a 2D maximum subarray search algorithm, this work could get comparable top detection performance on various benchmarks but also with major computation efficiency gains. Moreover, with the "side effect" of the coding method, the proposed

EMAS could further integrate global and object co-occurrence contextual information into the detection model with little extra effort, which is very effective to handle multi-class and occluded object detection. And the approach of this chapter could also be treated as one complementary method for current shape-based methods or even surpass them on some benchmarks.

Figure 6.5: Sample results from ILSVRC2012

Chapter 7

Main Results and Conclusion

This thesis focuses on the problem of visual object recognition. Following the state-of-the-art pipeline of visual recognition (feature extraction, feature encoding, feature pooling and model learning), several key improvements have been made through different approaches. The key results obtained in this thesis are:

1. For the feature encoding part, a review of current popular encoding methods was first presented. Different encoding methods were analysed in a unified platform to evaluate the true performance. Based on the analysis, a combination coding method (SuperCoding) is proposed, namely, the combination of FisherKernel coding and the generalized GMM mean vector coding. The proposed SuperCoding shows excellent performance on the standard datasets and different recognition tasks.

2. For the feature pooling part, we introduced a generalized hierarchical matching (GHM) pooling method for object-central recognition. This general and flexible scheme allows us to embed any useful side information into the visual recognition framework. Two novel exemplar approaches for side information generation towards object-oriented recognition are presented, i.e. object confidence map and visual saliency map. Our extensive experimental results clearly

demonstrated that the proposed GHM together with designed varieties of side information could achieve state-of-art performance on diverse and popular visual recognition datasets.

3. For the model learning part, we proposed an iterative contextualization scheme to mutually boost the performance of both object detection and classification tasks. The Contextualized SVM is proposed to seamlessly integrate external context features and subject features for general classification, and then Context-SVM was further utilized to iteratively and mutually boost performance of object detection and classification tasks. The proposed solution was extensively evaluated on both PASCAL VOC 2007 and VOC 2010 datasets and achieved the state-of-the-art performance for both tasks.

4. Furthermore, to extend our works we aimed to study the problem of large scale object recognition. An appearance-based approach with contextual information was proposed. By means of advanced coding and novel pooling scheme from a state-of-the-art large scale classification method and a 2D maximum subarray search algorithm, it was found that this work could get comparable top detection and classification performance on various benchmarks but also with major computation efficiency gains. Moreover, with the "side effect" of the coding method, the proposed EMAS could further integrate global and object co-occurrence contextual information into the detection model with little extra effort, which is very effective for handling multi-class and occluded object detection. And the approach of this method could also be treated as one complementary method for current shape-based methods .

7.1 Main Results

We also conclude the quantitative results from this thesis in the following sections.

Table 7.1: Performance improvement on PASCAL VOC 2007 dataset.

	VQ (4K)	Coding	Coding+Pooling	Coding+Pooling +Context	Improvement
aeroplane	68.5	79.9	76.7	84.5	16.0
bicycle	49.6	67.6	74.7	81.5	31.9
bird	39.4	50.8	53.8	65	25.6
boat	60.8	70.9	72.1	71.4	11.3
bottle	20.7	29.3	40.4	52.2	31.5
bus	48.0	67.1	71.7	76.2	28.2
car	67.9	80.9	83.6	87.2	19.3
cat	45.2	61.7	66.5	68.5	23.3
chair	47.0	48.1	52.5	63.8	16.8
cow	31.8	48.5	57.5	55.8	25.7
diningtable	35.2	52.2	62.8	65.8	30.6
dog	40.8	46.1	51.1	55.6	14.8
horse	66.4	80.7	81.4	84.8	18.4
motorbike	51.8	68.2	71.5	77	25.2
person	79.6	85.7	86.5	91.1	11.5
pottedplant	23.6	31.8	36.4	55.2	31.6
sheep	35.1	51.7	55.3	60	24.9
sofa	42.9	48.8	60.6	69.7	26.8
train	67.1	79.2	80.6	83.6	16.5
tvmonitor	46.5	55.6	57.8	77	30.5
mAP	48.4	60.2	64.7	71.3	22.9

7.1.1 Results 1: Effectiveness Improvement

We give the performance improvement results for object classification task on PASCAL VOC 2007 dataset as shown in Table 7.1. We start the comparison with the baseline of VQ coding using SIFT feature which obtain the mAP 48.4 on VOC 2007 dataset. Then using the proposed coding method (SuperCoding) improves the result significantly to mAP 60.2. The object-central pooling further improves the performance to mAP 64.7. Finally, the context modeling combined with other coding and pooling methods achieves the mAP of 71.3 which is the state-of-the-art performance.

We further conclude the performance evaluation on the recent years' PASCAL VOC challenges during which we obtain the winner title of object classification

Table 7.2: Comparison with state-of-the-art performance at the PASCAL VOC 2007, 2010, 2011 Challenges.

	VOC2007		VOC2010		VOC2011	
	Winner	Ours	Other's best	Ours	Other's best	Ours
aeroplane	77.5	84.5	90.3	**93**	94.5	**95.5**
bicycle	63.6	81.5	77	**79**	82.6	**81.1**
bird	56.1	65	65.3	**71.6**	**79.4**	**79.4**
boat	71.9	71.4	75	**77.8**	80.7	**82.5**
bottle	33.1	52.2	53.7	**54.3**	57.8	**58.2**
bus	60.6	76.2	**85.9**	85.2	**87.8**	87.7
car	78	87.2	**80.4**	78.6	**85.5**	84.1
cat	58.8	68.5	74.6	**78.8**	**83.9**	83.1
chair	53.5	63.8	62.9	**64.5**	66.6	**68.5**
cow	42.6	55.8	**66.2**	64	**74.2**	72.8
diningtable	54.9	65.8	54.1	**62.7**	**69.4**	68.5
dog	45.8	55.6	66.8	**69.6**	75.2	**76.4**
horse	77.5	84.8	76.1	**82**	83	**83.3**
motorbike	64	**77**	81.7	**84.4**	**88.1**	87.5
person	85.9	**91.1**	89.9	**91.6**	**93.5**	92.8
pottedplant	36.3	**55.2**	41.6	48.6	56.2	**56.5**
sheep	44.7	**60**	66.3	64.9	75.5	**77.7**
sofa	50.6	**69.7**	57	**59.6**	64.1	**67**
train	79.2	**83.6**	85	**89.4**	90	**91.2**
tvmonitor	53.2	**77**	74.3	**76.4**	76.6	**77.5**
mAP	59.4	**71.3**	71.2	**73.8**	78.2	**78.6**

tasks for the years through 2010 to 2012. As listed in Table 7.2, we achieved the best performance for VOC 2007 dataset. We obtained the best performance on object classification tasks with mAP of 73.8 for year 2010 [1], and the improvement is mostly from the Context Modelling part. For year 2011 [2], we obtained the best performance on object classification tasks with mAP of 78.6 due to the sophisticated feature coding and pooling methods.

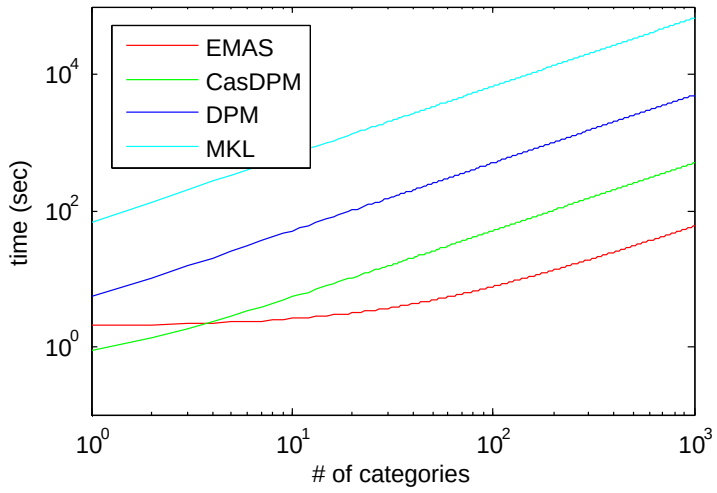

Figure 7.1: Computation cost in a multi-class setting.

7.1.2 Results 2: Scalability Comparison

Here, we compare the rough running cost of the proposed Efficient Maximum Appearance Search (EMAS) framework for large scale object detection problem with three object detection models as shown in Figure 7.1: 1) Multiple kernel learning for object detection (MKL) [24] using three-stage linear and non-linear detection, 2) Deformable Part Model [19] 3) Cascade DPM [107].

We first perform the full 1000 categories detection on ILSVRC 2012. The average running for one image is 58.4 seconds including 1.9 seconds for feature extraction and feature encoding, 56.5 seconds for 1000 categories model inference. So the added-on cost for each category is 56ms. For CasDPM and DPM, the feature pyramid for both method often takes 375ms, and needs 500ms, 5s respectively for model inference (rough estimate, changes for different setting). The cost for MKL reported in [24] is 67 seconds for one image. We can see the cost simulation for different approaches in Fig. 7.1 in a multi-class setting. It can be observed that our EMAS is not the fastest in the setting of few categories due to the feature encoding step cost. But it shows

[1] `http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2010/results/index.html`
[2] `http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2011/results/index.html`

at least one order of magnitude faster when the number of categories increases.

7.2 Conclusion

The overall system achieves in this study state-of-the-art performance considering two key factors, i.e. effectiveness and stability. On PASCAL VOC 2007 dataset, we promoted the recognition accuracy from 48% to 71.3% with a 48.5% relative improvement. In the past few years, numerous methods have been proposed to enhance the recognition rate on VOC 2007 [7, 11, 14]. However, to the best of our knowledge our system has achieved the best result. On Imagenet ILSVRC 2012 dataset, we accelerated the speed of 1000 object classes detection at least one order of magnitude faster than the current state-of-the-art method [19]. These two main results are obtained due to the fact that we made improvement at each step of the visual recognition pipeline. Compared with other works in the past few years, the works in this thesis concentrated on the separate stages of the recognition pipeline instead of one stage only. This makes the overall system obtain significant results. Furthermore, the methods proposed at each stage can be easily generalized to other similar framework. For example, we can use the SuperCoding to replace the coding method in ScSPM [11] to improve the result.

Although the overall system achieves significant results on the standard datasets, we notice the study has several limitations of this thesis. (1) We didnt make improvement over the "feature extraction" stage. The obtained results were based on the current popular hand-designed features, e.g. SIFT, HOG, LBP, instead of using feature learning. Despite the success of recent feature learning works, we find it is difficult to embed this kind of techniques into the overall framework, especially for the object detection tasks. (2) The recognition accuracy on the ILSVRC 2012 dataset is not satisfactory. The large scale object recognition problem is still on the going and needs to be thoroughly resolved. This problem is not unique to our study

as several groups in the world are working towards this direction. (3) This thesis focuses on the general problems of visual object recognition. Possible modification is needed to adapt to different application.

The built visual object recognition system has been demonstrated as practical and effective on the benchmark datasets. However, several directions can be further explored for visual object recognition. (1) Embed the feature learning part into the system. Iteratively learning the feature seems promising since it can naturally generate the feature which can best represent the data. (2) Explore the deep structure of the system. The current system can be viewed as three-layer architecture. A deeper structure should be designed and evaluated. One possible method is to construct iterative "coding-pooling" layer for the overall system. (3) One interesting question is raised after the system: what kind of possible application can be applied? and is it time to touch the further problem of visual recognition or artificial intelligence based on the current visual object recognition techniques, e.g. high level inference, decision planning? These directions are worthwhile to take both research and industry.

Bibliography

[1] J Sánchez and F Perronnin. High-dimensional signature compression for large-scale image classification. *Computer Vision and Pattern Recognition,*, 2011.

[2] Y LeCun, L Bottou, Y Bengio, and P Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[3] N Dalal and B Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition*, 2005.

[4] DG Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[5] T Ojala, M Pietikäinen, and D Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, 29(1):51–59, 1996.

[6] L Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *Computer Vision and Pattern Recognition*, 2005.

[7] S Lazebnik, C Schmid, and J Ponce. Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. In *Computer Vision and Pattern Recognition*, 2006.

[8] Kristen Grauman and Trevor Darrell. The pyramid match kernel: Discrimi-

native classification with sets of image features. In *International Conference on Computer Vision*, 2005.

[9] S K Divvala, D Hoiem, J H Hays, A A Efros, and M Hebert. An empirical study of context in object detection . In *Computer Vision and Pattern Recognition Workshops (CVPR Workshops)*, 2009.

[10] M Guillaumin and J Verbeek. Multimodal semi-supervised learning for image classification. In *Computer Vision and Pattern Recognition*, 2010.

[11] J Yang, K Yu, and Y Gong. Linear spatial pyramid matching using sparse coding for image classification. In *Computer Vision and Pattern Recognition*, 2009.

[12] J Wang, J Yang, K Yu, F Lv, and T Huang. Locality-constrained linear coding for image classification. In *Computer Vision and Pattern Recognition*, 2010.

[13] K Chatfield, V Lempitsky, and A Vedaldi. The devil is in the details: an evaluation of recent feature encoding methods. In *British Machine Vision Conference*, 2011.

[14] Florent Perronnin, Jorge Sanchez, and Thomas Mensink. Improving the Fisher Kernel for Large-Scale Image Classification. In *European Conference on Computer Vision*, 2010.

[15] X Zhou, K Yu, T Zhang, and T Huang. Image classification using super-vector coding of local image descriptors. In *European Conference on Computer Vision*, 2010.

[16] G Csurka and F Perronnin. A simple high performance approach to semantic segmentation. In *British Machine Vision Conference*, 2008.

[17] H Harzallah, F Jurie, and C Schmid. Combining efficient object localization

and image classification. In *International Conference on Computer Vision*, 2009.

[18] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A Zisserman. The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*, 2010.

[19] P F Felzenszwalb, R B Girshick, D McAllester, and D Ramanan. Object Detection with Discriminatively Trained Part-Based Models. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010.

[20] A Krizhevsky, I Sutskever, and G Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2012.

[21] L Bourdev. Poselets: Body part detectors trained using 3d human pose annotations. In *Computer Vision and Pattern Recognition*, 2009.

[22] Yi Yang and Deva Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *2011Computer Vision and Pattern Recognition* , 2011.

[23] Long Zhu, Yuanhao Chen, and Alan Yuille. Learning a Hierarchical Deformable Template for Rapid Deformable Object Parsing. *IEEE transactions on pattern analysis and machine intelligence*, 32(6):1029–1043, 2010.

[24] Andrea Vedaldi, Varun Gulshan, Manik Varma, and Andrew Zisserman. Multiple kernels for object detection. In *International Conference on Computer Vision (ICCV)*, 2009.

[25] C.H Lampert, M.B Blaschko, and T Hofmann. Beyond sliding windows: Object localization by efficient subwindow search. In *Computer Vision and Pattern Recognition*, 2008.

[26] Learning to Localize Objects with Structured Output Regression. In *European Conference on Computer Vision*, 2008.

[27] S An, P Peursum, Wanquan Liu, and S Venkatesh. Efficient subwindow search with submodular score functions. In *Computer Vision and Pattern Recognition*, 2011.

[28] Senjian An, P Peursum, Wanquan Liu, and S Venkatesh. Efficient algorithms for subwindow search in object detection and localization. In *Computer Vision and Pattern Recognition*, 2009.

[29] L Fei-Fei and R Fergus. One-Shot learning of object categories. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 28(4):594–611, 2006.

[30] G Griffin, A Holub, and P. Perona. Caltech-256 object category dataset. Technical report, 2007.

[31] Mark J Huiskes and Michael S Lew. The MIR flickr retrieval evaluation . In *Proceeding of the 1st ACM international conference*, pages 39–43, New York, New York, USA, 2008. ACM Press.

[32] Antonio Torralba, Rob Fergus, and William T Freeman. 80 Million Tiny Images: A Large Data Set for Nonparametric Object and Scene Recognition. *IEEE transactions on pattern analysis and machine intelligence*, 30(11), 2008.

[33] BC Russell, A Torralba, and KP Murphy. LabelMe: a database and web-based tool for image annotation. *International Journal of Computer Vision*, 77(1-3):157–173, 2008.

[34] Jia Deng, Wei Dong, R Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database . In *Computer Vision and Pattern Recognition*, 2009.

[35] Alexander Sorokin and David Forsyth. Utility data annotation with Amazon Mechanical Turk . In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPR Workshops)*, 2008.

[36] Alan F Smeaton, Paul Over, and Wessel Kraaij. Evaluation Campaigns and TRECVid. In *the 8th ACM international workshop*, pages 321–330, New York, New York, USA, 2006. ACM Press.

[37] S Branson and P. Perona. Strong Supervision From Weak Annotation: Interactive Training of Deformable Part Models. In *International Conference on Computer Vision*, 2011.

[38] Bangpeng Yao and Li Fei-Fei. Grouplet: A Structured Image Representation for Recognizing Human and Object Interactions . In *Computer Vision and Pattern Recognition*, 2010.

[39] Robert M Gray and David L Neuhof. Quantization. *Transaction on Information Theory*, 1998.

[40] David Forsyth, Philip Torr, and Andrew Zisserman, editors. Kernel Codebooks for Scene Categorization. *European Conference on Computer Vision*, 2008.

[41] K Yu, T Zhang, and Y Gong. Nonlinear learning using local coordinate coding. In *Advances in Neural Information Processing Systems*, 2009.

[42] H Jégou, M Douze, and C Schmid. Product Quantization for Nearest Neighbor Search. *IEEE transactions on pattern analysis and machine intelligence*, 33(1):117–128, 2011.

[43] Thomas Leung and Jitendra Malik. Representing and Recognizing the Visual Appearance of Materials using Three-dimensional Textons. *International Journal of Computer Vision*, 43(1):29–44, 2001.

[44] J Sivic and A Zisserman. Video Google: a text retrieval approach to object matching in videos. In *International Conference on Computer Vision*, 2003.

[45] D Nister and H Stewenius. Scalable recognition with a vocabulary tree. In *2006 Computer Vision and Pattern Recognition*, 2006.

[46] Shuicheng Yan, Xi Zhou, Ming Liu, Mark Hasegawa-Johnson, and Thomas S Huang. Regression from patch-kernel. In *Computer Vision and Pattern Recognition*, 2008.

[47] Herve Jegou, Florent Perronnin, Matthijs Douze, Jorge Sanchez, Patrick Perez, and Cordelia Schmid. Aggregating Local Images Descriptors into Compact Codes. *IEEE transactions on pattern analysis and machine intelligence*, 2011.

[48] David Picard and Philippe-Henri Gosselin. Improving image similarity with vectors of locally aggregated tensors . *International Conference on Image Processing (ICIP)*, 2011.

[49] Shuicheng Yan, Xi Zhou, Ming Liu, M Hasegawa-Johnson, and T.S Huang. Regression from patch-kernel. In *Computer Vision and Pattern Recognition*, 2008.

[50] B Schölkopf. The kernel trick for distances. In *Advances in Neural Information Processing Systems*, 2001.

[51] A Vedaldi and B Fulkerson. VLFeat: An open and portable library of computer vision algorithms. `http://www.vlfeat.org/`, 2008.

[52] L. Itti, C. Koch, and E Niebur. A model of saliency-based visual attention for rapid scene analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(11):1254–1259, 1998.

[53] C. Koch and S Ullman. Shifts in selective visual attention: towards the underlying neural circuitry. *Hum Neurobiol*, 4(4), 1985.

[54] A Oliva, A Torralba, M.S Castelhano, and J.M Henderson. Top-down control of visual attention in object detection. In *International Conference on Image Processing (ICIP)*, 2003.

[55] Antonio Torralba, Aude Oliva, Monica S. Castelhano, and John M. Henderson. Contextual guidance of eye movements and attention in real-world scenes: The role of global features in object search. *Psychological Review*, 113(4):766–786, 2006.

[56] F Shahbaz Khan and J van de Weijer. Top-down color attention for object recognition. In *Computer Vision and Pattern Recognition*, 2009.

[57] Christopher Kanan and Garrison Cottrell. Robust classification of objects, faces, and flowers using natural image statistics. In *Computer Vision and Pattern Recognition*, 2010.

[58] S Andrews and I Tsochantaridis. Support vector machines for multiple-instance learning. In *Advances in Neural Information Processing Systems*, 2003.

[59] Gang Wang and D Forsyth. Joint learning of visual attributes, object classes and visual saliency. In *International Conference on Computer Vision*, 2009.

[60] O Yakhnenko and J Verbeek. Region-Based Image Classification with a Latent SVM Model. Technical report, 2011.

[61] Y Chai, V Lempitsky, and A Zisserman. BiCoS: A Bi-level Co-Segmentation Method for Image Classification. In *International Conference on Computer Vision*, 2011.

[62] A Bosch, A Zisserman, and Xavier Munoz. Image classification using random forests and ferns. In *International Conference on Computer Vision*, 2007.

[63] A Rakotomamonjy, F Bach, S Canu, and Y Grandvalet. SimpleMKL. *Journal of Machine Learning Research*, 9:2491–2521, 2008.

[64] Terence Sim, S Baker, and M Bsat. The CMU pose, illumination, and expression database. *IEEE transactions on pattern analysis and machine intelligence*, 25(12):1615–1618, December 2003.

[65] P Welinder, S Branson, T Mita, C Wah, F Schroff, S Belongie, and P. Perona. Caltech-UCSD birds 200. Technical report, California Institute of Technology, 2010.

[66] KEA van de Sande, JRR Uijlings, T Gevers, and Arnold W M Smeulders. Segmentation as Selective Search for Object Recognition. In *International Conference on Computer Vision*, 2011.

[67] Fuxin Li, J Carreira, and C Sminchisescu. Object recognition as ranking holistic figure-ground hypotheses. In *Computer Vision and Pattern Recognition*, 2010.

[68] A Vedaldi and A Zisserman. Efficient Additive Kernels via Explicit Feature Maps. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (99):1, 2011.

[69] L Zhang, MH Tong, TK Marks, H Shan, and G W Cottrell. SUN: A Bayesian framework for saliency using natural statistics. *Journal of Vision*, 8:1–20, 2008.

[70] J H van Hateren and A van der Schaaf. Independent component filters of natural images compared with simple cells in primary visual cortex. *Proceedings of the Royal Society B: Biological Sciences*, 265(1394):359–366, March 1998.

[71] Kai-Sheng Song. A globally convergent and consistent method for estimating the shape parameter of a generalized Gaussian distribution. *Information Theory, IEEE Transactions on*, 52(2):510–527, 2006.

[72] M-E Nilsback and A Zisserman. A Visual Vocabulary for Flower Classification. In *Computer Vision and Pattern Recognition*, 2006.

[73] M-E Nilsback and A Zisserman. Automated Flower Classification over a Large Number of Classes. In *ICVGIP*, pages 722–729, 2008.

[74] A Olmos and E Kingdom. McGill Calibrated Colour Image Database.

[75] S Branson, C Wah, F Schroff, B Babenko, Peter Welinder, Pietro Perona, and Serge Belongie. Visual Recognition with Humans in the Loop. In *European Conference on Computer Vision*, 2010.

[76] Peter Gehler and Sebastian Nowozin. On feature combination for multiclass object classification. In *International Conference on Computer Vision* , 2009.

[77] M Marszałek, C Schmid, H Harzallah, and J van de Weijer. Learning object representations for visual object class recognition. In *Visual Recognition Challenge workshop, International Conference on Computer Vision* , 2007.

[78] Zheng Song, Qiang Chen, Zhongyang Huang, Yang Hua, and Shuicheng Yan. Contextualizing object detection and classification. In *Computer Vision and Pattern Recognition*, 2011.

[79] 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. *Computer Vision and Image Understanding*, 106(1):59–70, 2007.

[80] Myung Jin Choi, J.J Lim, and Torralba. Exploiting hierarchical context on a large database of object categories. In *Computer Vision and Pattern Recognition*, 2010.

[81] P Carbonetto, N De Freitas, and K Barnard. A statistical model for general contextual object recognition. In *European Conference on Computer Vision*, 2004.

[82] TL Berg and DA Forsyth. Animals on the web. In *Computer Vision and Pattern Recognition*, 2006.

[83] TL Berg, EC Berg, J Edwards, and DA Forsyth. Who is in the picture. In *Neural Information Processing Systems*, 2006.

[84] Geremy Heitz and Daphne Koller. Learning spatial context: Using stuff to find things. In *European Conference on Computer Vision*, January 2008.

[85] S Kumar and M Hebert. A hierarchical field framework for unified context-based classification. In *International Conference on Computer Vision*, 2005.

[86] L Wolf and S Bileschi. A critical view of context. *International Journal of Computer Vision*, 69(2):251–261, 2006.

[87] A Torralba. Contextual priming for object detection. *International Journal of Computer Vision*, 53(2):169–191, 2003.

[88] C Galleguillos and S Belongie. Context based object categorization: A critical survey. *Computer Vision and Image Understanding*, 114(6):712–722, 2010.

[89] A Oliva and A Torralba. The role of context in object recognition. *Trends in Cognitive Sciences*, 11:520–527, 2007.

[90] David A Forsyth, Jitendra Malik, Margaret M Fleck, Hayit Greenspan, Thomas Leung, Serge Belongie, Chad Carson, and Chris Bregler. *Finding pictures of objects in large collections of images*, volume 1144. Lecture Notes in Computer Science, Berlin, Heidelberg, August 1996.

[91] A Rabinovich and S Belongie. scenes vs. objects: A comparative study of two approaches to context based recognition. In *Computer Vision and Pattern Recognition* , 2009.

[92] B Yao and L Fei-Fei. Modeling mutual context of object and human pose in human-object interaction activities. In *Computer Vision and Pattern Recognition*, 2010.

[93] Alon Zweig and Daphna Weinshall. Exploiting Object Hierarchy: Combining Models from Different Category Levels. In *International Conference on Computer Vision*, 2007.

[94] Jia Deng, Alexander C Berg, and Li Fei-Fei. Hierarchical semantic indexing for large scale image retrieval,. In *Computer Vision and Pattern Recognition*, 2011.

[95] CJC Burges. A tutorial on support vector machines for pattern recognition. *Data Min Knowl Discovery*, 1998.

[96] Qiang Chen, Zheng Song, Rogerio Feris, Ankur Datta, Liangliang Cao, Zhongyang Huang, and Shuicheng Yan. Efficient Maximum Appearance Search for Large Scale Object Detection. In *Computer Vision and Pattern Recognition*, 2013.

[97] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.

[98] I T Jolliffe. *Principal Component Analysis*. Springer Verlag, October 2002.

[99] B Schölkopf, R Herbrich, and A Smola. A generalized representer theorem. *Computational learning theory*, 2111:416–41426, 2001.

[100] Koen E A van de Sande, T Gevers, and Cees G M Snoek. Evaluating Color Descriptors for Object and Scene Recognition. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1582–1596, 2010.

[101] F Perronnin, Z Akata, and Z Harchaoui. Towards Good Practice in Large-Scale Learning for Image Classification. In *Computer Vision and Pattern Recognition*, 2012.

[102] Long Leo Zhu, Yuanhao Chen, Alan Yuille, and William Freeman. Latent hierarchical structural learning for object detection . In *Computer Vision and Pattern Recognition*, pages 1062–1069. IEEE, 2010.

[103] C Desai, D Ramanan, and C Fowlkes. Discriminative models for multi-class object layout. In *Computer Vision and Pattern Recognition*, 2009.

[104] Myung Jin Choi, Antonio Torralba, and Alan S Willsky. A Tree-Based Context Model for Object Recognition. *IEEE transactions on pattern analysis and machine intelligence*, 34(2):240–252, 2011.

[105] Q Chen, Z Song, S Liu, X Chen, X Yuan, TS Chua, S Yan, Y Hua, Z Huang, and S Shen. Boosting Classification with Exclusive Context. In *The PASCAL VOC Challenge Workshop*, 2010.

[106] O Chum and A Zisserman. An Exemplar Model for Learning Object Classes. In *Computer Vision and Pattern Recognition*, 2007.

[107] PF Felzenszwalb and RB Girshick. Cascade object detection with deformable part models. *Computer Vision and Pattern Recognition*, 2010.

[108] Hyun Oh Song, Stefan Zickler, Tim Althoff, Ross Girshick, Mario Fritz, Christopher Geyer, Pedro Felzenszwalb, and Trevor Darrell. Sparselet Models for Efficient Multiclass Object Detection . In *European Conference on Computer Vision*, 2012.

[109] C Dubout and F Fleuret. Exact acceleration of linear object detectors. In *European Conference on Computer Vision*, 2012.

[110] Jon Bentley. *Programming Pearls (2nd Edition)*. Addison-Wesley Professional, 2 edition, October 1999.

[111] P. Belhumeur, D. Chen, S. Feiner, D. Jacobs, W. Kress, H. Ling, I. Lopez, R. Ramamoorthi, S. Sheorey, S. White, and L. Zhang. Searching the World's Herbaria: A System for Visual Identification of Plant Species. In *European Conference on Computer Vision*, 2008.

[112] V Lempitsky and A Zisserman. Learning to count objects in images. In *Advances in Neural Information Processing Systems*, 2010.

[113] Cho-Jui Hsieh, Kai-Wei Chang, Chih-Jen Lin, S. Sathiya Keerthi, and S. Sundararajan. A dual coordinate descent method for large-scale linear SVM. In *International Conference on Machine Learning*, 2008.

[114] C.H Lampert, M.B Blaschko, and T Hofmann. Efficient Subwindow Search: A Branch and Bound Framework for Object Localization. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(12):2129–2142, 2009.

[115] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding*, 110(3):346–359, June 2008.

[116] Y. Su and F. Jurie. Improving Image Classication using Semantic Attributes. *International Journal of Computer Vision*, 100, 1 (2012) 59-77, 2012.

[117] Perronnin, F and Dance, C. Fisher Kernels on Visual Vocabularies for Image Categorization. In *Computer Vision and Pattern Recognition*, 2007.

[118] J. Xiao, J. Hays, K A. Ehinger, A. Oliva and A. Torralba. SUN Database:

Large-scale Scene Recognition from Abbey to Zoo. In *Computer Vision and Pattern Recognition*, 2010.