The importance of spatial information on object recognition

Kristo

2016

Kristo. (2016). The importance of spatial information on object recognition. Master’s thesis, Nanyang Technological University, Singapore.

https://hdl.handle.net/10356/66396

https://doi.org/10.32657/10356/66396

NANYANG TECHNOLOGICAL UNIVERSITY

THE IMPORTANCE OF SPATIAL INFORMATION ON OBJECT RECOGNITION

KRISTO

SCHOOL OF ELECTRICAL & ELECTRONIC ENGINEERING

2016

THE IMPORTANCE OF SPATIAL INFORMATION ON OBJECT RECOGNITION

KRISTO

School of Electrical & Electronic Engineering

A thesis submitted to Nanyang Technological University
in partial fulfillment of the requirement for the degree of
Master of Engineering

2016

Acknowledgments

Praise to God Almighty, Father, Son, and Holy Spirit, who have allowed the author to embark on his doctoral journey. Praise for all His providence, blessings, and comfort throughout the process of this study. All glory, honor, and blessings belong to Him and Him alone.

The author would also like to express his appreciation to Associate Professor Chua Chin Seng for his counsel throughout his research. It was he who inspired the author to contribute to the research on this topic. His insight, experience, and expertise have been crucial in producing this work.

The author would also like to extend his gratitude to Assistant Professor Wang Gang for the opportunity to join his Vision Group. The same gratitude is also extended to all members of the group - Dr. Lu Jiwen, Mr. Amir Shahroudy, Mr. Wang Xingxing, Ms. Abrar Abdul Nabi, Mr. Liu Ting, Ms. Zuo Zhen, Mr. Shuai Bing, and Mr. Rahul Rama Varior - for the discussions, questions, answers, ideas, or simple jokes that have been traded over the past three years. The author's work would not have been the same without them.

The author dedicates this work to his family: his father, who gave him encouragement, his brother and sister-in-law, who brought out the best in him, and his niece, who brought a smile even in the hardest days of this work. The author would also like to express his gratitude to all friends and staff in both Intelligent Robotics Laboratory and Computer Vision Laboratory for their kind support and help.

The author would like to express his utmost gratitude to the entire FES IM and CLPC family for the prayers, support, and encouragement that enabled him to move forward even in the hardest times.

Finally, the author would like to express his sincerest gratitude to Nanyang Technological University, especially NTU President Professor Bertil Andersson, for awarding him the Nanyang President Graduate Scholarship and providing the research facilities that enabled the completion of this work.

Contents

Abstract

The "bag-of-words" (BoW) model is a staple in the field of computer vision, spanning applications within object, scene, and action recognition. An assumption inherent in the "bag-of-words" model is that each patch involved in the process is independent and unordered. As a result, BoW naturally neglects the important information of spatial locations and arrangements, leading to several drawbacks. Spatial Pyramid Matching (SPM) is the most popular framework used in incorporating spatial information into the BoW model. The model redefines an image as a pyramid consisting of several layers made of copies of the same image. Each layer $l \in \{0, 1, ..., L-1\}$ is divided into $2^l \times 2^l$ sub-windows/spatial windows, and from each sub-window a BoW descriptor is extracted. These descriptors are then concatenated, creating the SPM image descriptor.

SPM therefore offers a simple and efficient way to approximate spatial arrangements within the previously unordered collection of codeword histograms. Due to its simplicity, it has been very successful in many applications, and is even used in non-BoW methods. Very few works have questioned the effectiveness of this approach. The efficiency of spatial pyramids as an image descriptor and the appropriateness of SPM construction are simply taken for granted. This work will present a detailed investigation of the importance of spatial information in object recognition, and challenge the traditional SPM arrangement.

This thesis is divided into two parts. The first part presents an argument for the necessity of such knowledge, by showing how spatial information can significantly improve recognition systems. The Hierarchical Dirichlet Process (HDP) for image recognition is used to show this. The HDP suffers from the "rich-get-richer" effect caused by the way sampling is carried out. The first part of this thesis shows that spatial information can alleviate this issue, considerably improving the reliability of the HDP. We show that spatial information in the form of the cardinality coefficient and approximate shape masks is not only able to produce overall improvement in terms of accuracy ofobject recognition, but is also able to mitigate the detrimental effect inherent to the HDP.

With the realization that spatial information plays an important role in image

recognition, the second part offers a systematic investigation of the architecture of SPM. This study is done to show that SPM representation is sub-optimal, and at the same time present possible ways for improvement. In doing so, this thesis presents a few novel paradigms. From the second part, two novel paradigms are presented based on our investigation of the optimality of traditional SVM. Overlapping spatial windows (OWSPM) and circular spatial windows (CWSPM) present a new way of constructing the spatial pyramids, strengthening the discriminability of SPM representations by adding a broader context to each spatial window.

While OWSPM and CWSPM come from investigating the process of crafting SPM representation, the investigation of the arrangement of SPM led to our introduction of optimal spatial window arrangements. This comes in the form of Optimal Window SPM (OA-SPM) and a linear approximation of it in the form of LA-SPM. Combined, these proposed models were tested using various dataset and compared with several baseline methods such as ScSPM, LLC, Object Bank, and Deep Learning. A consistent and significant increase in performance, up to 4.38% with a lesser memory cost of nearly 40%, was reported, showing that the traditional spatial window arrangement of SPM is indeed inefficient.

The thesis will present the conclusion that SPM is sub-optimal on multiple fronts. In terms of structure, the disjointed window arrangement of traditional SPM actually performs poorly, and can be improved by the overlapping window arrangement. Furthermore, usage of overlapping windows enabled us to further explore the topic of optimality of SPM. The usage of all spatial windows inside a spatial pyramid proved to be more damaging than beneficial, as it hampers the discriminability of image representation, and adds unnecessary cost to the training and testing process.

List of Tables

List of Figures

List of Abbreviations

BoW "bag-of-words" model. 3, 4, 7, 10, 13–16, 31, 42, 63, 74, 75, 95, 111, 118

CNN Convolutional Neural Network. 20, 21, 75, 111, 119

CRF Chinese Restaurant Franchise. 36

CRP Chinese Restaurant Process. 36

CWSPM Overlapping circular windows SPM. 7, 8, 64, 70–76, 79–81, 83, 88, 96–98, 100–103, 116, 117

DHDP Dependent Hierarchical Dirichlet Process. 7, 32–34, 37, 40, 41, 48–50, 56, 58–60, 115, 116

DP-MM Dirichlet Process Mixture Model. 26, 27

FV Fisher Vector. 16, 96, 118

HDP Hierarchical Dirichlet Process. 4, 5, 7, 28, 29, 32–34, 36, 37, 41, 62, 115, 116

HOG Histogram of Oriented Gradients. 19, 20, 72

IWSPM Interleaved window SPM. 84, 87–94, 97, 98, 117

LA-SPM Linearly-optimized Arrangement SPM. 7, 8, 109, 111, 112, 117

LDA Latent Dirichlet Allocation. 10

LLC Locality-constrained Linear Coding. 12, 65, 67, 72–74, 81, 82, 97

MCMC Markov Chain Monte Carlo. 27

OA-SPM Optimized Arrangement SPM. 6–8, 96, 99, 100, 102–107, 109, 111, 112, 117–119

OWSPM Overlapping rectangular windows SPM. 7, 8, 64, 67–76, 79–81, 83, 86–94, 96–98, 100, 116, 117

PCA Principal Component Analysis. 16, 19, 49, 50

ScSPM Sparse Coded Spatial Pyramid Matching. 12, 64, 65, 67, 72–75, 80–82, 87, 89–91, 94, 97, 104, 118

SIFT Scale Invariant Feature Transform. 16, 19, 20, 43, 45, 49, 50, 67, 73

SPM Spatial Pyramid Matching. 3–8, 12, 14–16, 21, 63–68, 70, 72–76, 79–81, 83–88, 92–99, 101–103, 105–107, 109, 111–113, 116, 117

SURF Speeded Up Robust Features. 19

SVM Support Vector Machine. 63, 115

List of Symbols

A Subset of G. 25

C Number of classes in a database.. 69

$C(\cdot, \cdot)$ Acquaintance Coefficient. 37, 49, 52

$D(\cdot, \cdot, \cdot)$ Cardinality Coefficient. 37, 40, 52, 59, 60

G Probability distribution over a space Ω. 24–26

G_0 Probability distribution over a space Ω. 25, 26, 28, 34, 35

G_i Probability distribution over a space Ω. 28, 35

H Probability distribution over a space Ω. 28, 34

I An image. 50, 51, 68

K Total number of clusters. in chapter 3, denotes the total number of dishes circulating in the CRF. in other chapters, denotes the number of entries inside dictionary $\mathbf{V}$. This difference is due to the fact that in chapter 3, the clusters are being merged and split during the iterations, hence it's not a constant.. 11, 38, 68, 87, 88, 95

$M(x, y)$ Approximate shape mask value at position x and y. 46, 48, 51, 52, 61

N The number of patches inside an image I. 11

$R(\mathbf{P})$ Recognition accuracy of pyramid configuration $\mathbf{P}$. xiii, 99, 102, 108

S The number of images inside a dataset.. 100

T_j Number of tables occupied by customers at the j^{th} restaurant in CRF. 37

W The number of candidate windows to be optimized. 6, 99

Δ_i Two window adjustment factor for LA-SPM.. 107, 108

Λ Linkage between two patches in DHDP.. 36, 40, 41

Ω Symbol for the space used in Dirichlet Process throughout this thesis. 25, 34, 35

α_0 Dirichlet Process concentration parameter. 25, 26

β_k probability of drawing θ_k according to stick breaking construction. 35

ϵ Approximate adjustment factor for LA-SPM applicable to set of three window and higher.. 107, 108

γ Dirichlet Process concentration parameter. 28, 34

$\mathbf{B}$ The collection of all mid-level image descriptor from a specific dataset of size S.. 100

$\mathbf{P}$ Pyramid configuration used in learning classifier. v, xiii, 79, 92, 93, 99, 102, 108, 118

$\mathbb{U}(\mathbf{P}, w)$ Operation to add window w into the configuration $\mathbf{P}$.. 99

$\mathbf{U}$ A matrix obtained by putting together $\mathbf{u}_i$, just like how $\mathbf{X}$ is constructed. 11, 12, 67

$\mathbf{V}$ Codewords dictionary. 11, 12, 49–51, 67, 68, 84, 87

$\mathbf{W}$ The set of all sub-windows inside a spatial pyramid. 99

$\mathbf{X}$ A $D \times N$ matrix containing all features in an image. 11, 34, 51, 84

$\mathbf{Z}$ The descriptor of a particular image. Also called as mid-level image descriptor by some literature. 68, 69, 100

$\mathbf{u}_i$ The code for patch $\mathbf{x}_i$. 11, 84

$\mathbf{v}_k$ Dictionary entry inside $\mathbf{V}$. 12

$\mathbf{x}_i$ The feature descriptor of patch i. 11, 51, 84

$\mathbf{z}$ The descriptor of a particular sub-window. Also called as mid-level feature by some literature. 84

ω Overlap parameter. ix, xii, xiii, 66, 69–72, 74, 84, 86, 88, 102

ϕ_{jt} The t^{th} table of the j^{th} restaurant in CRF. 36, 37

π_k Infinite sequence used in stick breaking construction. 25

σ_{ϕ_k} Unit mass at point ϕ_k. 25

$\mathbf{F}(\mathbf{x}, \theta)$ Distribution of $\mathbf{x}$ given θ. 29

θ Parameter for Dirichlet Process drawn from base distribution H. Corresponds to latent themes shared between multiple image categories. 34

θ_k Intermediate variable used in DPMM, refers to the pointer to k^{th} menu in Chinese Restaurant Metaphor, acting as the k^{th} latent theme learned from the process. 34, 35, 37

φ_{ji} The i^{th} customer at the j^{th} restaurant in CRF. 36

h_s Height of a sub window at a particular layer. 71

$j(w)$ The number of times window w appears in image I_j.. 38

L Total number of layer inside SPM. 14, 63, 79, 83–85, 95, 96, 100

l The l^{th} layer inside SPM. $0 \leq l \leq L - 1$. xii, 3, 15, 63, 71, 86

n_{jt} Number of seater in ϕ_{jt} in CRF. 37, 39

$p(\mathbf{x}_j | \theta_j)$ Posterior probability matrix learned from DHDP/modified DHDP. 49, 51

w_s Width of a sub window at a particular layer. 71

E1 Testing of modified DHDP with no approximate shape mask. ix, xi, 51–56, 58–61

E2 Testing of modified DHDP with approximate shape mask. ix, xi, 51, 52, 55, 58–60

Chapter 1

Introduction

1.1 Motivation

The visual capability of a human being is a wondrous creation. It is easy to take this ability for granted, simply because we have grown accustomed to it in our daily routines. The human visual system enables us not only to see, but also to comprehend the information contained in what we see. The human visual system can distinguish between individual objects based on their shape and prior knowledge of the nature of objects. It is able to recognize the type of the object, from a very general (it is a round object) to a very specific classification (it is an IKEA meatball). Furthermore, it can adapt very well to variations in its surroundings: light, intensity, different points of view, color changes, occlusion, and moving objects. It is a wonder that we can do all these tasks in a split-second.

However, this is not the case with current man-made computer visual systems. Computer vision, as researchers call this area of study, is limited by the computational power of the system as well as current scientific understanding of how it works. While humans can integrate all the above tasks into a single action, current advances in computer vision research only allow us to do a few tasks a time. Often, we are forced to confine ourselves to a specific task:

tracking a moving object, localizing and segmenting the object of interest in an image, or recognizing what the person in the image is doing, or what kind of object is within the image.

Object recognition and categorization is a fundamental problem in image visualization. It remains a challenging task given both the variability of images that objects from the same class can produce, and the substantial expenses incurred from providing high-quality image annotations to train the detectors [4]. Advancements in this area can greatly contribute to our society: take for example the release of the Microsoft Kinect™. The idea behind it is in fact very simple: the camera only has to detect the existence of one or two people inside its field of view, segment their bodies, and track their movements to make a new breed of human-computer interface [5]. Why not extend this beyond the bounds of the human body to the detection of any item that the person is holding on to?

Other examples of usage come from the bloom of social networking websites such as Facebook, Google+, and Twitter. Thousands of images are uploaded to the internet via these social networks, and users are given a choice to tag these pictures or provide the location where they were taken. This also applies to picture-sharing services provided by websites such as Instagram. Currently, it is the user that needs to provide all this information, often with a great deal of trouble that comes from hash-tagging. With an improved detector, we can create an automated system to assist the user. Therefore, the data generated can also be used to train a more advanced system, propagating progress in this field.

Take, for example, the now growing topic of self-driving cars. The realization of such technology will involve a comprehensive visual system inside an automobile, in terms of both driving assistance and decision making based on the situation present on the road. These are just some of the many possible uses that this area of study can contribute to, not to mention the more sophisticated application of such advancements in areas like artificial intelligence, robotics, automation, and manufacturing.

Thus, it is not surprising that one of the critical tasks in this field is to find a representation of the images involved. Such a representation needs to be informative, and able to provide information on the contents and contexts of the image, and at the same time be compact and efficient, to enable a real system to process it swiftly.

Several works have presented the idea that the visual perception of an object extends beyond the object itself [6–9]. Preliminary work in this area tended to confine the visual extent to within the object's silhouette, by segmenting the object from the image [10]. This led to the idea that an object should be correctly segmented before it is itself recognized. With time, the use of powerful descriptors, increased numbers of datasets for learning, and the advances of statistical learning have circumvented the necessity of identifying the object's location before classifying it, such as in the "bag-of-words" model (BoW) representation.

From the BoW representation, the work on crafting image representation flourished, but Spatial Pyramid Matching (SPM) might be the most commonly used technique throughout the literature. SPM is a staple of object recognition techniques. Ever since its initial conception, it has been attractive to many researchers due to its simplicity and ease of use. SPM models the spatial information in a rough fashion, by dividing an image into $2^l \times 2^l$ disjoint windows of equal size at each pyramid level l. These spatial windows are then used for the pooling step to create the final representation of the image.

Although this method seems simple, it has functioned very well in many practical applications. Its ability to be implemented in various existing approaches also makes it desirable. SPM has therefore been widely accepted as an essential component in many computer vision techniques. Few works have questioned the effectiveness of SPM; however, this thesis will discuss how SPM is actually not optimal, and how it can be further improved without introducing significant additional cost to image representation.

This work was motivated by two factors. The first is the confidence that spatial information, even in its crudest form in SPM, contains important information that is invaluable for ensuring discriminability in image representation. The second factor is the fact that SPM is able to evaluate this spatial information, but is highly inefficient in doing so. This work therefore proposes to improve SPM by providing a highly discriminable but compact image representation.

This work will show how spatial information can enhance image representations that do not traditionally contain them, such as BoW representations, by introducing them into the Hierarchical Dirichlet Process (HDP). After establishing that spatial information is integral to image representation, the thesis will touch on improving SPM by optimizing its very core: the partitioning of an image into spatial pyramids.

1.2 Objectives

As discussed in the previous section, the objectives of this research are

1. Investigating the effects of spatial information on object recognition.
2. Formulating a novel image representation that is highly discriminative and compact.
3. Evaluating the optimality of Spatial Pyramid Matching (SPM).
4. Optimizing the SPM model based on the evaluation.

To achieve these objectives, we define the following scope for our research activities:

1. Finding a representation of an image to be processed by the system, including the selection of features inside the image, describing it appropriately, and/or simplifying its description.
2. Derivating suitable techniques to enable the system to learn, given the image database.
3. Using the learnt knowledge to find suitable classification methods.

4. Optimizing each methods and moving beyond it to tackle the weaknesses.
5. Implementing/adapting the idea into various current trends in the field.

1.3 Contributions

The following contributions have been made in this research:

1.3.1 Minimizing the "rich-get-richer" effect of HDP

The Hierarchical Dirichlet Process (HDP) involves breaking and merging clusters of data to produce a distribution of latent themes from the resulting clusters. However, the computation of such distribution favors clusters with a large membership size which tend to absorb smaller clusters. This effect is aptly named the “rich-get-richer” effect, and in the worst-case scenario, we will end up with one large cluster which will absorb all data points from the whole sample. Obviously, such a scenario will render classification impossible.

This thesis proposes to modify the learning method HDP to ease this effect using two techniques: (a) strengthening meaningful data by creating a linkage between clusters that occur frequently in a class, and (b) weakening background clutter by approximating the object mask for an image. These two approaches modify the representation of each image being fed to the HDP to minimize the snowballing of the “rich-get-richer” effect.

This contribution is discussed in detail in Chapter 3.

1.3.2 Usage of overlapping spatial regions to improve SPM

Viewing an image as several disjointed groups, as is done in SPM, is rarely done in real life. When we identify an image, we evaluate what we see over several overlapping areas that share information between them. Imitating them on SPM will strengthen image representations due to the sharing of information between windows. In view of this, this thesis proposes to improve SPM by introducing the overlapping spatial windows of two shapes: rectanges and circles. These two schemes have been proven to be beneficial for improving the discriminability of SPM image representation, allowing us to achieve up to a 3% increase in recognition accuracy compared to classical SPM.

This contribution is discussed in detail in Chapters 4 and 5.

1.3.3 Optimization of spatial window arrangement in SPM

The introduction the overlapping spatial window improved the discriminability of each spatial window to the extent that it was unnecessary for the whole set of windows to be included into the image representation to achieve the performance level of the traditional SPM. This fact leads us to question not only the setup of these spatial windows, but also the arrangement of windows in the traditional SPM. It was also found that including all spatial windows led to a lower overall classification accuracy, compared with smaller subsets of possible arrangements.

This means that there exists an optimal arrangement for SPM, and if we can search for an arrangement that can maximize classifier performance, we will not only be able to achieve better accuracy, but also lower memory and computational cost. A greedy method called Optimized Arrangement SPM (OA-SPM) is proposed to find such an arrangement, and experiments show that almost half the number of spatial windows in SPM are not needed for an optimal arrangement.

The optimization process of OA-SPM allows us to obtain a highly efficient image representation that is able to outperform the traditional SPM while having a shorter representation (which leads to lower memory and computational cost in training the classifier). However, the optimization involves a process that is of $O(W^2)$ complexity, where W is the number of candidate windows. We pushed OA-SPM further by formulating a linear approximation of the OA-SPM process, resulting in an optimization process that performs similarly to OA-SPM but runs in linear complexity. We call this the Linearly-optimized Arrangement SPM (LA-SPM).

This contribution is discussed in detail in Chapter 6.

1.4 Organization of thesis

This thesis is divided into three main parts. First, we show that spatial information is important in order to understand the objects contained in the image for the bag-of-words model. This is achieved using a modified Dependent Hierarchical Dirichlet Process (DHDP). Following this, the second part will show that there is much room for improvement in SPM, and discuss the definition of the spatial window, introducing two novel spatial window models: Overlapping rectangular windows SPM (OWSPM) and Overlapping circular windows SPM (CWSPM). Finally, the third part investigates systematic approaches to learning the optimal arrangement of spatial windows in OA-SPM and LA-SPM.

The thesis is organized as follows:

Chapter 2 reviews the background of the research related to this work. This chapter begins by describing the existing datasets for object recognition, followed by a review of the BoW model and SPM. The chapter then proceeds to discuss feature extraction and the description of the image, followed by a review of the non-parametric learning paradigm. The chapter will also cover object segmentation/localization techniques, and end with a review of the recognition algorithm itself.

Chapter 3 describes the first part of this thesis, which is the usage of the cardinality of codewords to assist in the learning process of the HDP. The chapter describes the original algorithm of [1] and integrates the cardinality coefficient into the algorithm. Furthermore, the proposed solution is expanded by integrating approximated shape mask generation to assist in feature selection and learning. At the end of the chapter, the improvement in recognition performance is discussed in detail.

Chapter 4 explains the second part of this thesis. This chapter presents overlapping windows as a means to include more holistic context in the SPM model. The idea leads to two novel paradigms for SPM arrangement: the overlapping rectangular window (OWSPM) and circular overlapping window (CWSPM). The chapter describes the construction of both arrangements, and shows that the two proposed methods are better than the traditional SPM.

Chapter 5 covers the third part of this thesis. The chapter discusses the suboptimality of the spatial window arrangement of SPM by using an Interleaved Window arrangement as an example (referred to as the IW scheme), and discusses the findings in detail, explaining why it is possible to conclude that the arrangement of SPM is highly redundant.

Chapter 6 continues the discussion from the previous chapter by introducing the algorithm for finding the optimal arrangement of SPM in the form of OA-SPM. OA-SPM was limited by having $O(n^2)$ complexity, which made it slow to perform with increasing window candidates. A linear approximation to OA-SPM, in the form of LA-SPM is proposed to alleviate this issue. Both approaches are evaluated, and the results indicate that our claim of SPM suboptimality is true.

Chapter 7 presents the conclusions of this thesis and suggestions for future work.

Chapter 2

Literature Review

2.1 Existing datasets for object recognition

Currently, there are a huge number of vast image databases for object recognition, available for free. With the proliferation of the internet today, and increasing interest in research on object recognition, these image databases are easily accessible, and the results obtained by different researchers are then compared with each other.

Caltech datasets (Caltech-4, Caltech-101 [11], and Caltech-256 [12]) comprise several classes of objects (the dataset Caltech-C means that there are C classes inside the dataset), centered within the image with little or no clutter in the background. Each class contains 30 to 400 images.

The 15-Scene [13] dataset contains scene images categorized into 15 classes. This is a dataset of fifteen natural scene categories with around 200-300 images in each class.

The PASCAL Visual Object Challenge Database [14] is updated yearly, with each dataset containing typically 20 classes (since the 2007 version) with the purpose of providing a standardized image dataset for object class recognition. Most work is based on either the 2007 or 2008 version of the dataset.

CIFAR-10 is an established computer-vision dataset used for object recognition. It is a subset of the 80 million tiny image dataset and consists of 60,000 32x32 color images containing 1 of 10 object classes, with 6000 images per class. It was collected by Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton [15].

Another good source of images is the internet. Websites such as Picasa, Instagram, Google Image Search or even Facebook can used as virtually limitless sources of images in a very broad context. Tagging and categorizing, however, are different issues as they can be freely labeled by users. The labeling process tends to involve a great amount of noise. Image quality poses another problem, as it tends to be inconsistent across many images in terms of both size and resolution. Recently, an active area of research has been the creation of databases by mining images from the internet, categorizing them using the information from the tags and the image itself, propagating them into a large database. Researchers have been reporting success with this approach and demonstrating its application in object recognition [16].

Naturally, a hybrid database comprising both is also possible. Starting from an existing database, one can develop a system to propagate and obtain more images for its database. In addition, due to progress in the area of object recognition, the problem has been intensified to tackle more detailed categorization, such as distinguishing between different species of birds or trees [17]. Yao et al. used the Caltech-UCSD Birds dataset [18] for fine-grained image categorization of birds in [19].

2.2 The bag-of-words model

The “bag-of-words” (abbreviated as BoW from here on) model has been receiving a lot of interest from the research community in the area of object recognition. This model has existed since 1954 [20], and was originally used for topic detection in methods such as Latent Dirichlet Allocation (LDA) [21], until re-

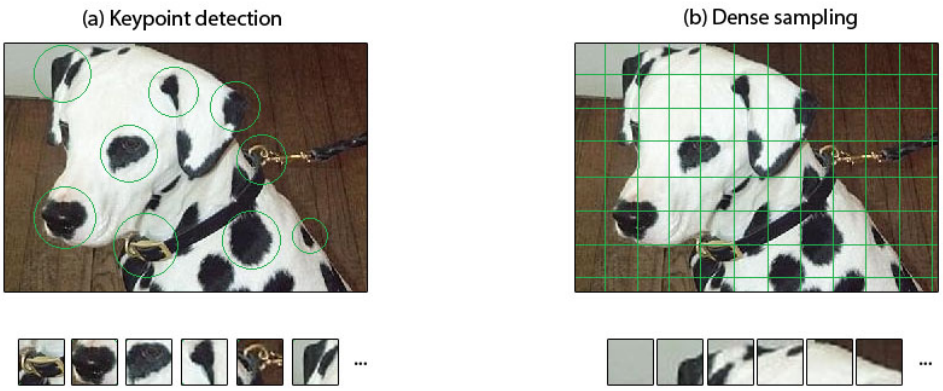


Figure 2.1: Illustration of two keypoint extraction method. (a) Using keypoint detection, salient image patches are located and extracted. Algorithms described in chapter 3 utilize this method. (b) Image patches are extracted over dense grid covering the entire image. Grid sizes are usually set such that patches extracted are overlapping with each other. In this figure, the spacing of grids are enlarged for clarity. Algorithms discussed in chapter 4 to 6 utilize this method.

searchers such as Zhang et al. pioneered its usage in the imaging domain [22]. Analogous to its counterpart for documents, we extract patches from an image using a variety of possible approaches, like keypoint extraction [1, 3] or dense sampling [23–25]. These patches are then categorized as codewords, obtained from a specific codeword dictionary. The dictionary is learned from a collection of patches using techniques such as K-means clustering.

The process of categorizing patches into codewords is known as the coding step. In early BoW models, a patch was only associated with its nearest codeword center in the dictionary. If $\mathbf{u}_i$ was the code for patch $\mathbf{x}_i$, then $\mathbf{u}_i$ would have exactly one element (also referred as the membership cooefficient) with a value of 1 and 0 everywhere else. Let $\mathbf{X}$ be a matrix where each column is a collection of N patch descriptors with D dimensionality, i.e., $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N]$ ($x_i \in \mathbb{R}^{1\times D}$, $\mathbf{X} \in \mathbb{R}^{N\times D}$). Let $\mathbf{V} \in \mathbb{R}^{D\times K}$ be the codeword dictionary with K entries learnt from collecting random patches from datasets, and associate $\mathbf{x}_i$ to the entries of $\mathbf{V}$. In practice, each column of $\mathbf{V}$ stores the cluster center of the codewords. Associating patches to dictionary entries is achieved with a simple

K-means algorithm by optimizing the following:

$$\begin{aligned} &\min_{\mathbf{U},\mathbf{V}} \sum_{n=1}^{N} \|\mathbf{x}_n - \mathbf{V}\mathbf{u}_n\|^2 \\ \text{subject to } &\{\mathbf{u}_{ni} = 1, \mathbf{u}_{nj} = 0, \forall i \neq j\} \\ \text{and } &\|\mathbf{u}_n\| = 1, \mathbf{u}_{ni} \geq 0, \forall n = \{1, 2, ..., N\}. \end{aligned} \tag{2.1}$$

$\mathbf{U}=[\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_N]^T \in \mathbb{R}^{N \times K}$ is the cluster membership indicator and $|| \cdot ||$ denotes the $L2$ norm. In other words, the membership indicator will have exactly one element with a value of 1 and 0 everywhere else. This encoding scheme is called *hard assignment*.

Membership assignment evolved in further works to allow for multiple assignments of dictionary entries, called *soft assignment*. The objective function to be optimized is now written as:

$$\begin{aligned} &\min_{\mathbf{U},\mathbf{V}} \sum_{n=1}^{N} \|\mathbf{x}_n - \mathbf{V}\mathbf{u}_n\|^2 + \lambda \|\mathbf{u}_n\| \\ \text{subject to } &\|\mathbf{v}_k\| \leq 1, \forall k = 1, 2, ..., K. \end{aligned} \tag{2.2}$$

where $\mathbf{V}=[\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_K]$ where $\mathbf{v}_k$ is the dictionary entry of the k^{th} codewords. Under this scheme, the assignment coefficient $\mathbf{u}_n$ is forced to be sparse by adding the $L2$-norm into the objective function. The constraint $\|\mathbf{v}_k\| \leq 1$ is added to avoid trivial solutions and λ is a constant to modify the degree of sparsity. Sparse assignment proves to be beneficial in object recognition, enabling performance boosts when applied to SPM [13]. The encoded patches are then pooled together to create the image representation.

This soft coding approach shows a powerful understanding on how to encode patches, which gives rise to powerful encoding methods such as Sparse Coded Spatial Pyramid Matching (ScSPM) [25], Locality-constrained Linear Coding (LLC) [23], Laplacian [24], or Dictionary Learning: Commonality and Particularity (DL-COPAR) [26]. Other encoding methods that have been able to achieve large success includes Fisher Vector [27, 28], Vector of Locally Aggregated Descriptors (VLAD) [29], and t-embedding [30, 31]. As an alternative to these, Poselets [32, 33] and Deformable Part Model (DPM) [34] have also been used, mainly in the field of human detection. More detailed explanations on the optimization of $\mathbf{U}$ and $\mathbf{V}$ can be found in [25, 31]. Liu et al. give a thorough description on the benefit of soft coding for object recognition in [35].

Each of the encoded patch will then be aggregated into a single feature vector by a pooling operation. This step is known as the pooling step. Koniusz et al. presented a detailed report on various pooling methods in [36]. Among the options, two pooling schemes that are commonly used are average pooling and maximum pooling. Let $\mathbf{z}_w$ be the image representation and $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_N]$ be the set of all $\mathbf{u}_i$ belonging to that image. The average pooling is given as:

$$\mathbf{z} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{u}_i \tag{2.3}$$

On the other hand, the maximum pooling can be given as:

$$\begin{gathered}\mathbf{z} = [z_1, z_2, ..., z_k]^T = [\max_j u_{1j}, \max_j u_{2j}, ..., \max_j u_{Nj}]^T \\ \text{where } \mathbf{u}_i = [u_{i1}, u_{i2}, ..., u_{iK}]^T \text{ and } j = 1, ..., N\end{gathered} \tag{2.4}$$

$\mathbf{z}$ is the BoW image representation and is used to learn the classifier. Figure 2.2 illustrates the bag-of-words model.

In the earlier stages of object recognition research, it was widely accepted that before we started taking information from the image, we first needed to localize the object. Some researchers used silhouette information [37] while others used edge detection techniques combined with other methods [38]. However this segmentation task is another problem in itself, as it is very difficult to precisely locate the object. The BoW model is introduced as an alternative approach. Basically, it shows that background clutter does not affect detection accuracy; its presence may even improve the detection result [10]. In this model, an image is represented simply as a collection of patches from detected points

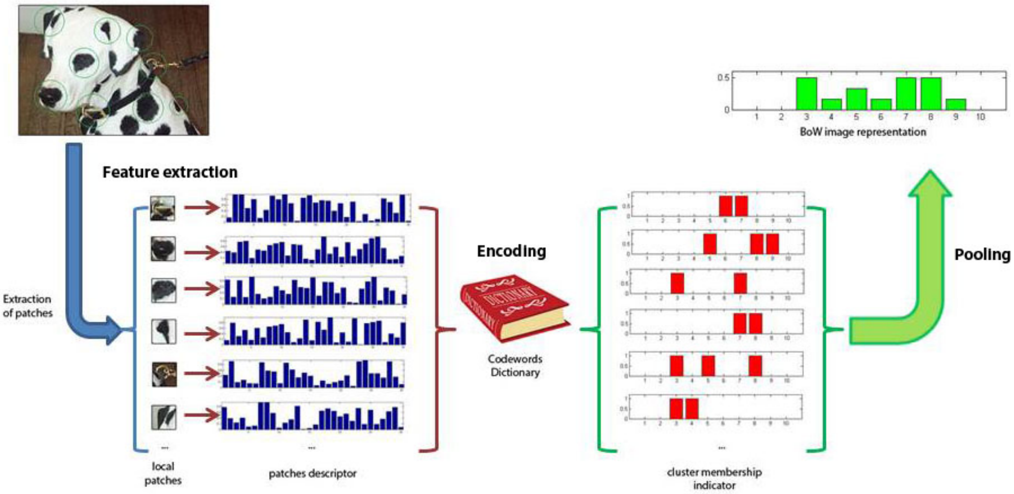


Figure 2.2: Illustration of the “bag-of-words” model.

of interest.

The concept of patch locations and their relation towards other patches are neglected, due to the assumption of conditional independence imposed on the documents (images) [39]. This is due to the fact that we consider each image to be only a collection of patches. Hence, we assume that these patches are independent from each other. It should also be noted that this assumption may discard useful information for learning and recognition.

Another problem is that the BoW model treats each patch as being equal to the others, although we can never be sure that even the most advanced salient point detector will return patches from only the object itself. In real life, we may have multiple objects in a scene, or some backgrounds that stand out extremely well.

There has been a significant amount of research done to improve the bag-of-words model, whether in feature descriptors and encoding [23,25,40], dictionary learning [26, 41–45], classifiers used [46–48], or the pipeline in general [13, 46, 49]. Liu and Wang presented a tool called Restricted Support Region Set Detection in [35], to visualize what the classifier had learned from a specific algorithm. Other works focus their efforts on reducing the dimensionality of the feature learned [50, 51].

2.3 Spatial Pyramid Matching

As mentioned in the previous section, one challenge of the BoW model comes from the exclusion of spatial information in the final image representation. Lazebnik et al. presented a solution to this problem in [13] by incorporating a rough spatial arrangement of patches into the image representation. This approach is called wthe Spatial Pyramid Model (abbreviated as SPM).

In SPM, an image is now duplicated L times such that we have multiple layers of the same image, creating a pyramid. Each layer $l \in \{0, 1, 2, \ldots, L-1\}$ is divided into $2^l \times 2^l$ equal-sized, disjoint regions. The BoW representation is applied to each of this spatial windows. This means that a single patch can be pooled more than once based on their spatial window membership. At the end of the process, multiple codeword histograms are obtained, one for each spatial window. The final image representation is obtained by concatenating the histograms into a single vector.

SPM might seem overly simplistic at a first glance. However, it is able to offer a considerable improvement to the results of the learnt classifier. The simplicity and power of SPM have attracted many researchers, putting it as a mainstay in most computer vision pipelines. Many state-of-the-art performances have been achieved with SPM as an integral part of the image representation [23–25, 28, 52–59].

It is worth noting, however, that the SPM approach is used with very little to no modification. While coding and pooling schemes have received much attention in the research community, the SPM model itself is challenged by very few works. Most works focus their attention on finding suitable pooling functions, neglecting the spatial arrangement used by SPM. Jia and Huang proposed a scheme to adaptively learn receptive fields (spatial regions) based on a collection of possible spatial windows in [60]. Yan et al. proposed to use all possible spatial window arrangements (covering a spatial window's aspect ratio, size, and location), and to apply Principal Component Analysis (PCA) for feature

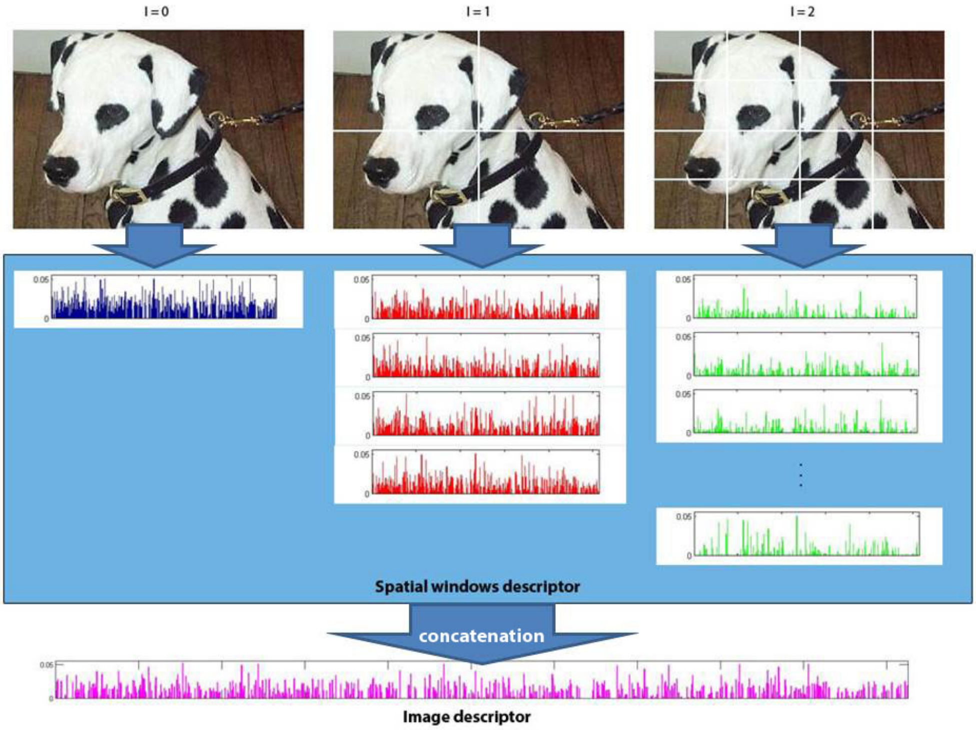


Figure 2.3: The spatial pyramid matching model on an image from "dalmatian" class with $L = 3$.

selection [58]. Krapac et al. have shown in [28] that using the Fisher Vector (FV) as an appearance descriptor not only allows a smaller codeword dictionary (a property that is very important in FV), but also when combined with spatial pyramids, it is able to achieve state-of-the-art performances with only two layers of the pyramid. This is particularly interesting, as it indirectly asserts that the current SPM architecture is sub-optimal.

2.4 Key-point detectors

In the BoW model, patches are normally obtained using three methods: (a) random sampling, (b) dense sampling, or (c) key-point extraction based on salient regions. An illustration for (b) and (c) can be found in figure 2.1. The first two methods are straightforward, even though it is possible to yield significant results using these methods. There are, however, several ways to execute the last method.

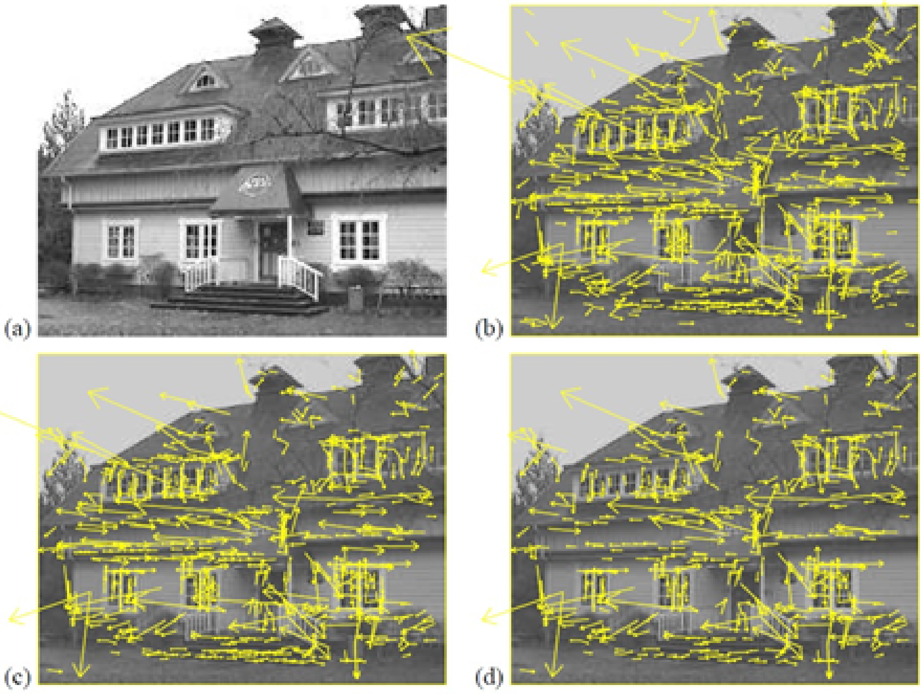


Figure 2.4: SIFT detection result with step-by-step filtering of key-points (image taken from [2]).

Various techniques have been developed to extract these salient points that can be regarded as being representative of the object(s) inside an image. For a start, the Scale Invariant Feature Transform (SIFT) algorithm, while primarily a feature descriptor, provides a way to collect key points before being described. Through a combination of scale-space pyramids, the key points are filtered using their local extrema, followed by localization and elimination of edge responses [2]. An image of 200×200 pixels may produce up to 500 key-points.

Kadir and Brady [61] proposed to detect interest points based on the saliency within the image being detected. This algorithm, known as the KB saliency-scale detector, searches for a visually salient region over different image scales based on the image's appearance (geometric features, rarity and local complexity), creating measures of intra- and inter-scale entropies which are used to detect key points. One can control the number of key points detected by adjusting the threshold where entropy is considered a salient point.

The Harris-Laplace detector [62] provides another alternative for extracting regions of interest. It is invariant to scale transformations, and detects points

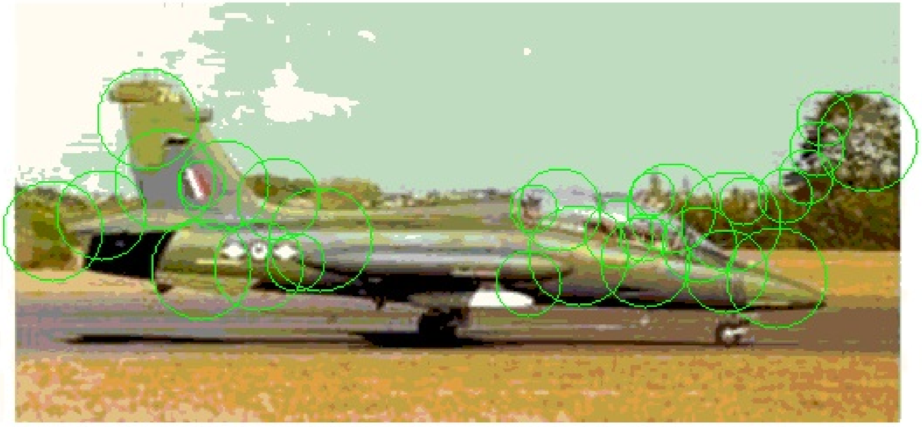


Figure 2.5: Kadir-Brady detection result.

that correspond to corner-like regions. The outputs of the detector are circular regions at certain characteristics of scale. Another similar type of detector is the Laplacian detector [63] which extracts blob-like regions from the image. The Harris-Laplace detector tends to produce comparable results with fewer detections, as compared to the Laplacian detector. However, it might be necessary to use the Laplacian detector especially when the test image is small in size.

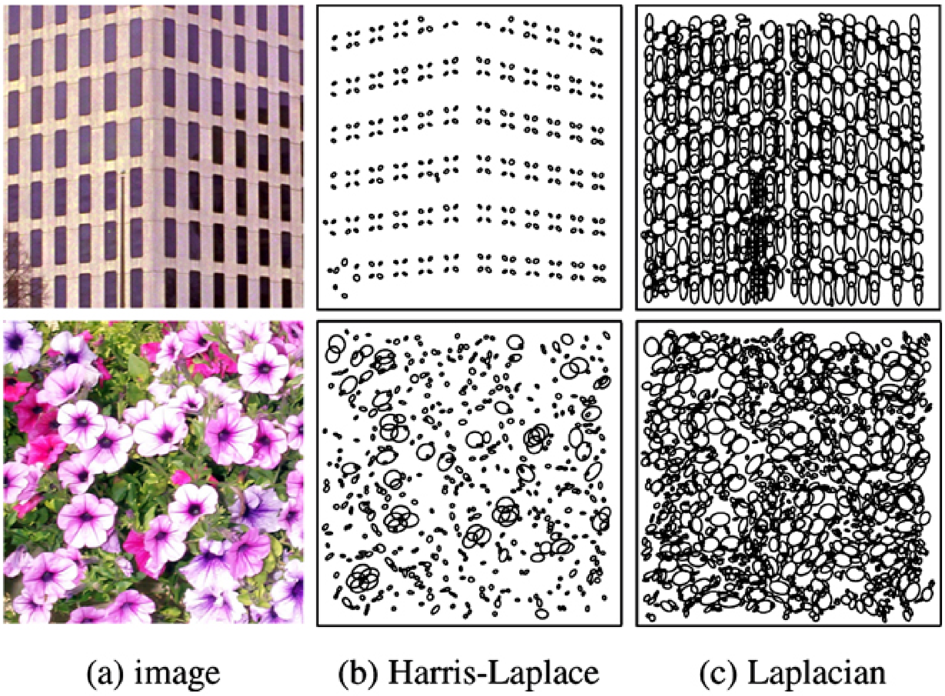


Figure 2.6: Harris-Laplace and Laplacian detection result (Image taken from [3]).

2.5 Feature descriptor

Mapping image patches directly onto the bag-of-words model is costly and non-beneficial. The collected patches need to be expressed in another representation $\mathbf{x}_i$ that has more efficiency in utilization. This representation is commonly known as a feature descriptor, as it describes the feature points within the images that represent these patches.

Such descriptors need to be designed to match the needs of object recognition tasks. Considering that images may be taken from different distances and perspectives, the descriptor should be invariant to these changes. In other words, it needs to be at least scale-invariant (due to variations in object size or distances from which the image is taken) and rotation-invariant (due to variations in object orientation or perspective).

One of the best and popular descriptors used in recent times is the SIFT [2]. Patches of a specific scale are divided into 4×4 sub-regions (the number 4 can be substituted with other numbers as well). An edge detection is done to each of the sub-region, and the resulting magnitude/orientation pair from each pixel position is aggregated in an 8-dimensional vector. The aggregation is done by pooling the magnitudes based on their orientation with respect to 8 general direction. Hence, we transform the representation of the patches into a 128-dimension vector (depending on the number of sub-regions and the number of the gradient bin, in this case $4 \times 4 \times 8 = 128$). Although the dimensionality of the descriptor could be very high, we can also utilize the PCA to reduce its size.

Speeded Up Robust Features (SURF) [64] is a robust feature partly inspired by SIFT. SURF is several times faster than SIFT, and is more to different transformations compared to SIFT. It makes use of the sum of 2D Haar-wavelet responses around the point of interest with the aid of integral images.

The Histogram of Oriented Gradients (HOG) [65] is another example of a widely used feature descriptor. First publicized by Dalal and Triggs, HOG describes local object appearances and shapes based on the distribution of intensity gradients or edge directions. Similar to SIFT, during implementation, HOG divides the patch into small connected regions (termed cells), obtains the histogram, and combines them into the descriptor. Since HOG operates on localized cells, the method upholds invariance to geometric and photometric transformations, with the exception of orientation.

2.6 Non-parametric Bayesian Learning

One milestone in the development of Artificial Intelligence is the acceptance of uncertainty and inductive reasoning as primary concerns within the field. While the term “uncertain” seems to convey an opposite message from “intelligent”, it was Judea Pearl who managed to shift that opinion. Early AI researchers tended to focus on mimicking the deductive capabilities of human intelligence. This changed in post-Pearl research, which accepted the uncertainty surrounding a realistic environment, and tried to explicitly represent these uncertainties so as to mitigate their effects. It might well be that the only way to bridge the gap between systems of limited and robust intelligence is by embracing uncertainty. Computationally, it involves two aspects: explicit representations of uncertainty and the algorithmic manipulations of these representations to reduce uncertainty. Pearl showed that these two aspects are intimately related [66] obtaining a compact representation of uncertainty will lead to an efficient algorithm for marginalization and conditioning, which in turns leads to reducing uncertainty.

Uncertainty about an environment can also be reduced by observing the environment, i.e., learning from a collection of data, which has been an early focus of deduction in AI. However, many researchers of machine learning do not wish to make the assumption that the learner needs to maintain an explicit probabilistic model of the environment. Many learning algorithms involve some degrees of algorithmic procedures that are not necessarily interpretable as computations of conditional probability. Their unconditional performance is used over and over on various datasets as justification of these procedures.

It is worth noting that statistics involves the interplay of both the conditional (Bayesian) and the unconditional (frequentist) perspectives that underline much development in AI research. Ever since Pearl's work in the 1980s, there has been a trend to blend reasoning and learning: one does not need to learn from the data that which one can infer from the model, and vice versa. Thus, learning and reasoning interact. The most difficult problems in AI are currently being approached with methods that blend reasoning with learning. There remain, however, several limitations of probabilistic and statistical approaches.

It is generally accepted that to use probabilistic methods in AI, one is forced to write out a list of assumptions. While this is often a helpful exercise, some of the assumptions are not well motivated. Assumptions of independence, for example, are often imposed for reasons of computational convenience, and not because they are deemed to be true in the environment. Also, some choices of convenience influence the adoption of various assumptions.

The non-parametric Bayesian learning pursues a different approach to expressive probabilistic representations, and a less assumption-laden approach to inference. The idea is to move beyond the simple fixed-dimensional random variables that have been generally used in graphical models, and to consider a wider range of probabilistic representations. With the usage of flexible data structures that can expand and contract as needed, e.g. trees, list and collections of sets, it is possible to produce an efficient algorithm. The existing field of stochastic processes essentially provides this kind of flexibility. Within the general theory of stochastic processes, it is quite natural to define probability distributions on spaces of probability distributions to yield an appealing recursivity; or to apply it to trees, lists and collection of sets.

One way to use stochastic processes in inferences is by taking a Bayesian perspective and replacing the parametric distributions in classical Bayesian analysis with stochastic processes. For example, we could consider a model in which the prior distribution is a stochastic process that ranges over trees of arbitrary depth and branching factors. By combining it with the likelihood, we obtain a posterior distribution (which is also a stochastic process) that ranges over trees of arbitrary depth and branching factors. Bayesian learning amounts to updating one flexible representation into another flexible representation (prior to the posterior). This idea is called Bayesian nonparametrics.

The word ‘nonparametrics’ does not mean ‘no parameters’, in fact, many stochastic processes have been described in many (or even infinitely many) parameters. It instead means ‘not parametric’, in the sense that inference is not restricted to objects whose dimensionality stays fixed over increasing amounts of data, thus giving flexibility of data structures, where representations can grow as needed. As such, the Bayesian nonparametric approach is less assumption-laden than classical Bayesian parametric learning.

2.6.1 De Finetti's theorem

De Finetti's theorem [66] provides a natural point of departure for the discussion of non-parametric Bayesian learning, as it provides as one of the pillars of Bayesian inference.

De Finetti's theorem. *Suppose* $\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3, ...$ *is an infinite exchangeable sequence of Bernoulli random variables. Then* $\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3, ...$ *are conditionally independent and identically distributed given some random variable* $\mathbf{Y}$ *with probability distribution* $m \in [0, 1]$.

A random variable $\mathbf{X}$ has a Bernoulli distribution if Pr$(\mathbf{X} = 1) = p$ and Pr$(\mathbf{X} = 0) = 1 - p$. De Finetti's theorem states that the probability distribution of any infinite exchangeable sequence of Bernoulli random variables is a "mixture" of the probability distributions of independent and identically distributed sequences of Bernoulli random variables. "Mixture" in this sense means a weighted average, but this not necessarily means a finite or countably finite weighted average, it can also be an integral rather than a sum. This theorem suggests the need of prior distributions in a statistical model and directly implies the consideration of stochastic processes as Bayesian priors.

Consider an infinite sequence of random variables $(\mathbf{x}_1, \mathbf{x}_2, \dots)$, assumed to be discrete. We say that such a sequence is infinitely exchangeable, if the joint probability distribution of any finite subset of those random variables is invariant to permutation. De Finetti's theorem states that $(\mathbf{x}_1, \mathbf{x}_2, \dots)$ are infinitely exchangeable, if and only if, the joint probability distribution of any finite subset can be written as a marginal probability in the following way:

$$p(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N) = \int_{-\infty}^{\infty} \prod_{i=1}^{N} p(\mathbf{x}|G)P(dG) \qquad (2.5)$$

As such, the theorem can be interpreted as stating that exchange-ability implies the existence of an underlying parameter as well as a prior distribution on that parameter, hence it is often viewed as the foundational support for the Bayesian paradigm. There are also no restrictions that G should be a finite dimensional object.

2.6.2 The Pólya urn model

In Pólya urn model, objects of real interest are represented as colored balls in an urn or other container. Suppose the urn contains x white and y black balls. One ball is drawn randomly from the urn and its color observed and is then returned into the urn. An additional ball of the same color is added to the urn, and the selection process is repeated. In this way, every time a particular value is observed, it becomes more likely to be observed again. Additionally, successive acts of measurement over time has less and less effect on future measurements.

The model provides an example in which the count of balls in the urn is not concealed. A Bayesian analysis of the observer's uncertainty about the urn's initial content can be made, using a particular choice of prior distribution. This basic Pólya urn model has been enriched in several ways, and in this thesis we are particularly interested in Dirichlet Process. Suppose that we start with an urn of α black balls. If we draw a black ball, put the ball back along with a new ball that is non-black, randomly generated from a uniform distribution over an infinite set of available colors, and consider the generated color as the value of the draw. If a non-black ball is drawn, put the ball back into the urn along with a ball of the same color (instead of generating a new color), as with standard Pólya urn scheme.

This modified Pólya urn model gives a simple example of the realization of infinite-dimensional G and the stochastic process P. While the model defines a distribution on labels, it can also be used to induce distribution on partitions. Such a model leads to the Chinese restaurant process [67].

Chinese Restaurant Process. *Imagine a Chinese restaurant with an infinite number of circular tables. Assuming that each table have infinite capacity, the first customer of the restaurant is seated at an unoccupied table with probability* 1. *At time* $n+1$ *a new customer visits the restaurant and chooses at random to sit at the possible* $n+1$ *places: to the left of the* n *customer already seated at an occupied table, or seat at a new, unoccupied table.*

2.6.3 The Dirichlet Process

Consider a distribution $\pi = (\pi_1, \pi_2, ...)$ on positive integers. We can view them as a sequence of a non-negative numbers that sum to one. To obtain a random sequence that sums to one, the "*stick-breaking*" sequence is introduced: Define an infinite sequence of independent random variables

$$\beta_k \;\sim\; \text{Beta}(1, \alpha_0), \;\; k = 1, 2, ... \tag{2.6}$$

where $\alpha_0 > 0$ is a parameter. The beta distribution follows the following probability density function:

$$\begin{gathered} f(x; \alpha, \beta) = \frac{1}{B(\alpha, \beta)} x^{\alpha-1}(1-x)^{\beta-1} \\ B(x, y) = \int_0^1 t^{x-1}(1-t)^{y-1} \mathrm{d}t \text{ for } \text{Re}(x), \; \text{Re}(y) > 0 \end{gathered} \tag{2.7}$$

Now define an infinite random sequence

$$\begin{gathered} \pi_1 = \beta_1 \\ \pi_k = \beta_k \prod_{l=1}^{k-1} (1 - \beta_l) \end{gathered} \tag{2.8}$$

Clearly, the sum of the numbers is equal to one.

We can exploit this construction to generate a large class of random distributions on sets other than these integers. Consider an arbitrary measurable space Ω, and let G_0 be a probability distribution over that space. In this thesis, the term *measure* in the context of Dirichlet Process is referring to *probability measures*.

Probability measures. *a function μ is a probability measure on a probabilty space if (1) μ return results in the unit interval $[0, 1]$, returning 0 for empty set and 1 for the entire space, and (2) μ must satisfy the countable additivity properti that for all countable collections $\{E_i\}$ of pairwise disjoint sets $\mu(E_1 \cup ... \cup E_n) = \mu(E_1) + ... + \mu(E_n)$.*

Draw an infinite sequence of points $\{\phi_k\}$ independently from G_0. Now define

$$G = \sum_{k=1}^{\infty} \pi_k \sigma_{\phi_k} \tag{2.9}$$

where σ_{ϕ_k} is a unit mass at the point ϕ_k. Clearly, G is a measure, and indeed for any measurable A subset of Ω, $G(A)$ adds up the values π_k for those k where $\phi_k \in A$. This process also satisfies the countable additivity needed in the definition of a measure. Since $G(\Omega) = 1$, G is a probability measure. In this definition, G is a stochastic process where the indexing variables are the measurable subsets of Ω; since for any fixed A subset of Ω, $G(A)$ is a random variable ranging over subsets $\{A_1, A_2, \dots, A_k\}$, hence the joint distributions of the collections of random variables $G(A_i)$ are consistent with each other. If we specialize the sets $\{A_1, A_2, \dots, A_k\}$ as a partition of Ω, the random vector $\{G(A_1), G(A_2), \dots, G(A_k)\}$ can be shown to have a finite-dimensional Dirichlet distribution of

$$\{G(A_1), G(A_2), ..., G(A_k)\} \sim \text{Dir}\{\alpha_0 G_0(A_1), \alpha_0 G_0(A_2), ..., \alpha_0 G_0(A_k)\} \tag{2.10}$$

from which the required properties for consistency follow immediately from the properties of the Dirichlet distribution. Hence, the stochastic process described by equation 2.9 is known as the Dirichlet Process, while equation 2.10 shows that the Dirichlet process has a Dirichlet marginal.

Further inspection on the Pólya urn model will show that the Dirichlet process is the De Finetti mixing distribution underlying the Pólya urn. Equation 2.10 is usually expressed as follows:

$$G \sim \text{DP}(\alpha_0, G_0) \tag{2.11}$$

Here we say that the Dirichlet Process has two parameters: the concentration parameter α_0 (proportional to the probability of obtaining a new color in the Pólya urn) and the base measure G_0, which is the source of the "atoms" ϕ_k.

2.7 Dirichlet process mixture model

The Dirichlet process defines a prior on partitions of objects, and this prior can be used to develop a Bayesian non-parametric approach to clustering. As discussed earlier, with this non-parametric approach, one does not have to fix the number of clusters a priori. Let $(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)$ be a sequence of random vectors whose realizations we want to model in terms of an underlying set of clusters. We treat these variables as exchangeable and conditionally independent (as suggested by De Finnetti), given an underlying random element G. Drawing G from a Dirichlet process, we define a Dirichlet Process Mixture Model (DP-MM) [67] as

$$\begin{aligned} G &\sim \text{DP}(\alpha_0, G_0) \\ \theta_i &\sim G,\ i = 1, ..., N \\ \mathbf{x}_i &\sim p(\mathbf{x}|\theta_i),\ i = 1, ..., N \end{aligned} \tag{2.12}$$

where $p(\mathbf{x}_i|\theta_i)$ is a cluster-specific distribution. The use of the intermediate variable θ_i is simply an expanded way to write the factor $p(\mathbf{x}_i|G)$. In particular, G is a sum across atoms, and thus θ_i is one of the atoms in G, chosen with a

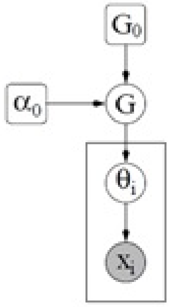


Figure 2.7: Graphical model representation of Dirichlet Process Mixture Model.

probability equal to the weight assigned to that atom. The graphical model of the DP-MM is shown in Figure 2.7.

2.7.1 Inference for Dirichlet process mixtures

We briefly describe on Markov Chain Monte Carlo (MCMC) inference procedure for the DP-MM in this thesis, which is attributed to Escobar [68]. It should be noted that there are various procedures existing in other forms, as described by Neal in [69].

Consider the equation

$$p(\theta, \mathbf{x}) = p(\theta_1, \theta_2, ..., \theta_N) \prod_{i=1}^{N} p(\mathbf{x}_i|\theta_i) \tag{2.13}$$

We note that the equation induces a Pólya urn marginal distribution on $\theta = (\theta_1, \theta_2, \ldots, \theta_N)$. Equation (2.13) shows the joint distribution on θ and $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ where the first factor is the Pólya urn model. This can be viewed as the product of the prior and its likelihood.

The variable $\mathbf{x}$ is deemed as fixed through inference (as observed data) with the goal to sample θ. A Gibbs sampler is developed to achieve this purpose. Using the property of exchangeability, the joint probability of $(\theta_1, \theta_2, \ldots, \theta_N)$ is invariant to permutation, meaning that we can permute the vector to move θ_i to the end of the list. By integrating the product of the urn representation and the likelihood, we can obtain the conditional distribution of θ_i for all i.

2.7.2 Hierarchical Dirichlet Process

In a hierarchical Bayesian model, the joint distribution of all the variables in the model is obtained as a product over conditional distributions, where each conditional may depend on other variables in the model. The literature on this graphical model has focused almost entirely on parametric hierarchies, where each of the conditionals is a finite-dimensional distribution. However, it is also possible to build hierarchies where its components are stochastic processes. Applying this to the Dirichlet process, we obtain the Hierarchical Dirichlet Process (HDP).

Recall that a Dirichlet process $G_i \sim \text{DP}(\alpha_0, G_0)$ is a random measure G_i, that has a 'parameter' G_0 that is itself a measure. If we treat G_0 itself as a draw from a Dirichlet process and let the measures $G_1, G_2, \ldots, G_m$ be conditionally independent for a given G_0, we obtain the following hierarchy:

$$\begin{aligned} G_0|\gamma, H &\sim \text{DP}(\gamma, H) \\ G_i|\alpha, G_0 &\sim \text{DP}(\alpha_0, G_0) \end{aligned} \tag{2.14}$$

Where γ and H are concentrations and base measure parameters at the top of the hierarchy. This construction yields an interesting kind of 'shrinkage'. Recall that G_0 is a discrete random measure, with its support on a countable infinite set of atoms. Drawing $G_i \sim \text{DP}(\alpha_0, G_0)$ means that each G_i will also have its support on the same set of atoms. The weights are obtained via conditionally independent stick-breaking processes.

The HDP is useful when we want to tackle multiple clustering problems, in applications where we wish to relate the groups formed to each other, as in this thesis. To achieve this, the hierarchical Dirichlet process mixture model is

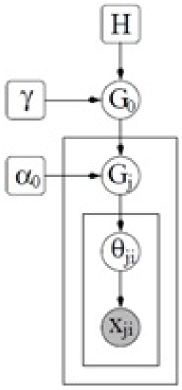


Figure 2.8: Graphical model representation of Hierarchical Dirichlet Process.

described as

$$
\begin{aligned}
G_0|\gamma, H &\sim \text{DP}(\gamma, H) \\
G_i|\alpha, G_0 &\sim \text{DP}(\alpha_0, G_0), \;\; i = 1, ..., m \\
\theta_{ij}|G_i &\sim G_i, \;\; j = 1, ..., N_i \\
\mathbf{x}_{ij}|\theta_{ij} &\sim \text{F}(\mathbf{x}_{ij}, \theta_{ij}), \;\; j = 1, ..., N_i
\end{aligned}
\tag{2.15}
$$

$\text{F}(\mathbf{x}, \theta)$ is the distribution of $\mathbf{x}$ given θ. The graphical model of HDP is shown in Figure 2.8.

2.8 Localization Technique

In the problem of object detection, the possibilities of having cluttered real-world scenes are very high. In such cases, it is not only necessary to assign the correct category label to an image, but also to firstly find the objects and to separate them from the background. Historically, this step of figure-ground segmentation has long been seen as an important, and even necessary, precursor for object recognition, as discussed in the previous section. However, researchers have generally been faced with the failure of achieving task-independent segmentation. Coupled with the success of appearance-based methods to provide recognition without segmentation, the two areas have since diverged.

The criteria for measuring localization accuracy have evolved over time.

Agarwal and Roth [70] evaluated the center-point of an object and classified its localization as correct when the marked point was considered in close neighborhood with the actual center of the object. Modern localization methods should go even further, to return additional information like poses, viewpoints, articulations and aspects.

A few methods that interweave object detection and segmentation have been recently developed by researchers such as Leibe et.al. [71], Fusseneger et.al. [72], and Liu et.al. [37]. Liu et al. proposed a scheme of unsupervised segmentation, where object detection and object segmentation help each other to produce better results in segmentation. This has been shown to be beneficial in the work of Marzalek and Schmid [3], by using training information to produce an approximate image mask to filter out (or reduce the effect) of codeword clutter. This work tried to tackle the problem with the same approach in Chapter 3, to see the effects of such segmentation on non-parametric learning.

2.9 Convolutional Neural Network

More recently, several works have shifted their focus onto unsupervised feature learning by the means of Deep Convolutional Neural Network (CNN) [73]. In contrast to crafted features like SIFT and HOG, unsupervised learning utilizes the neural network with a non-linear layer on each stage to learn a suitable feature descriptor. The resulting features are inexpensive to compute, and are able to model the latent features shared by the patches. Some works [55–57, 74] have succeeded in producing results on par with results from state-of-the-art crafted features, or performing even better. Also called Deep Learning, with its introduction invariant features have learned to adapt to intra-class image variations [75, 76].

Deep Learning has been picking up paces swiftly since 2013. Deep Learning first appearance can be traced as far back as back as 1980s in the form Convo-

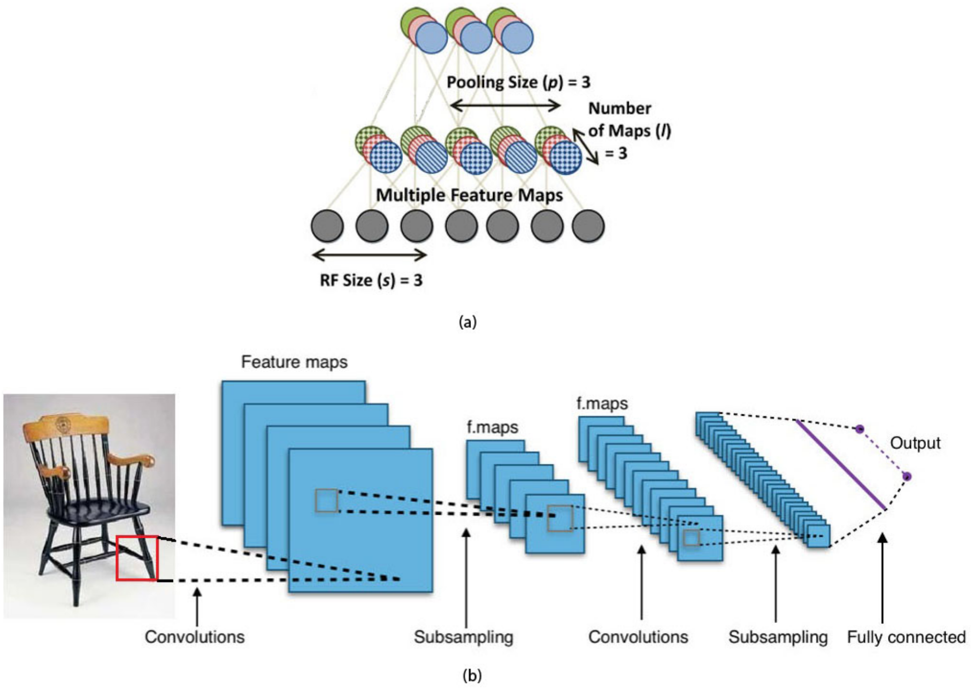


Figure 2.9: Common structure of a Convolutional Neural Network. (a) a typical CNN consists of a convolutional layer which is depicted by the bottom connected network, and a non-linear layer (usually accompanied by pooling step) depicted by the top connected network. This pooling layer is also referred to as subsampling layer. (b) Deep CNN stacks multiple CNNs into one single, deep network. In this figure, two CNN units are stacked together.

lutional Neural Network. The design was then improved in the period of 1998 to 2003, and started to be noticed in 2006. It was around 2013 when some good results on computer vision can be seen, although it was still being outperformed by the bag-of-words approach. It was not until early 2014 that CNNs started to outperform bag-of-words and sparse coding approach.

With the advent of deep convolutional neural networks, we are witnessing a rapid, revolutionary change in the vision community. Deep learning-based approaches have shown substantial improvement in current technologies of image classification, object detection, and various other recognition and non-recognition tasks. A single layer unit in a CNN mainly consists of two parts: convolutional layers (connected sparsely) and fully connected layers following it. The convolutional layers operate in the manner of sliding windows, which give feature map outputs that represent the spatial arrangement of activations.

The second part involves inputs of fixed size that represent non-linearity and pooling. Several layer unit can be stacked together to create deep architecture, thus the name Deep Learning.

This thesis will not cover CNNs and Deep Learning extensively, as chapter 3 does not utilize them in any way, and the contributions from Chapter 4 to 6, was mostly tested on bag-of-words methods. The reasoning against inclusion of an extensive review on CNNs is that it leads to swamping the thesis with information that does not directly relate to the work in this thesis. As groundbreaking and important the works on CNNs and Deep Learning are, this work is not on object recognition methods but improving SPM that is basically a common element among most object recognition algorithm. This work, however, will translate its proposed method in chapter 6 to 8 into CNNs, and show that the improved SPM paradigm is beneficial not only to bag-of-words approach but to Deep Learning as well.

SPM, however, is used widely on all family of methods, be it bag-of-words, regionlets, sparse-coding, or deep learning. Some of these algorithms do not mention SPM by name, but a quick look to the pooling mechanism will enable us to make the association to SPM. In the case for convolutional neural network, SPM is present in the form of the fully connected layer. In this layer, responses from previous layer are aggregated by pooling based on their spatial location and memberships with respect to several sub-windows. The architecture of such pooling mechanism are identical to SPM, without calling SPM specifically by name. As one of this thesis goal is to show that SPM is sub-optimal, it will also touch on how the proposed model in this thesis can be translated to instances of SPM in Convolutional Neural Network.

Chapter 3

Region Cardinality and Approximate Shape Mask

3.1 Introduction

The BoW (bag-of-words) approach is one of the most popular approaches for image representation within object categorization. The usage of such a model normally follows the four basic steps: extraction of patches, feature description, vocabulary construction (encoding), and image representation (pooling/aggregation) [77]. These steps are performed independently with respect to the intended object classes for detection. This model is therefore considered a bottom-up approach.

Normally, the BoW model assumes that the patches of an image are independent. This assumption considerably simplifies the complexity of the model. However, it is evident from real-world experience that there can be connections between components of an object. These connections enable our vision system to easily categorize the object. The assumption of independence between patches will therefore results in the discarding of useful information contained in the dependencies between the image patches.

Let us consider two objects, a table and a wooden bed frame (as shown in

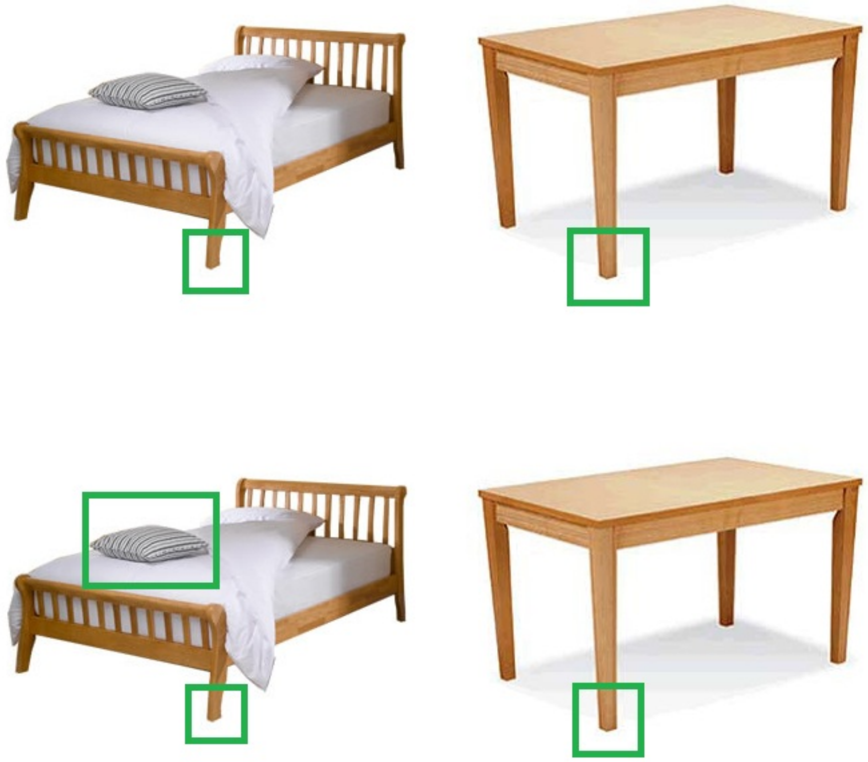


Figure 3.1: Dependent patches as assistive information for object recognition. A “wooden leg” can explain both “table” and “bed” as illustrated in the top row. However, the existence of a “pillow” might enhance the recognition performance.

Figure 3.1), to illustrate this phenomenon. For the wooden bed frame, the parts “wooden leg” and “pillow” tend to occur in the same image. However, if we tackle both vocabularies independently, we might misclassify the bed frame as a table because of its “wooden leg”. If we know that there is a “pillow” near a “wooden leg”, and include that information in our recognition process, we can easily distinguish between the two objects. This concept is the foundation of the Dependent Hierarchical Dirichlet Process (DHDP) proposed by Wang et al. [1].

The Hierarchical Dirichlet Process (HDP) is a non-parametric Bayesian model that extends the Dirichlet Process to multiple levels [67]. Since the process is layered hierarchically, it is possible for us to fit our model into the HDP to share latent themes between multiple object categories. The DHDP extends the HDP by introducing linkages between patches to measure the strength of their dependencies. This model aims to learn the theme distribution of the objects and train a classifier for object categorization.

Even with carefully set parameters for salient scale detectors, it is unavoidable that some patches collected would not be help achieve our goal of object categorization. This may happen when the object has bad contrast with the surroundings, or if there are many objects cluttered in the background. Let us call such patches noise patches. Noise patches tend to cluster together in the DHDP training process as they often have similar descriptors. This in turn creates a large group of noise patches from all image categories.

The Dirichlet Process (and consequently HDP and DHDP) tends to pull smaller groups of patches into a group of patches, which in turn has a large number of associated patches (referred to as the "rich-get-richer" effect). This effect is strengthened in DHDP, as the linkage strength always multiplies the distribution in favor of the stronger category. With a large number of training images, a large number of noise patches will be accumulated. This collection of noise patches will in turn merge the other latent themes to itself, which is very detrimental to recognition performance as discriminative parts may be merged along with it. One particular solution is to prevent this from happening by setting appropriate settings for the detector. However, this is not an easy task to perform, and the settings could differ depending on the application as well. This work proposes a new approach to tackle this problem, which involves incorporating the cardinality of the patches into the training and recognition steps of the algorithm.

Going back to the example of the “table”, we may observe that a table tends to have three or four “wooden leg” vocabularies, depending on the viewpoint of the image. Different viewpoints may give rise to the occlusion of some parts, but the number of parts will not deviate too far from a certain value. The same also applies to images og a “car” and “motorcycle”, which we know have four and two “wheels” respectively. This work proposes to integrate this information, in addition to the patches and their dependencies, into the learning and recognition system, to improve overall performance and reduce the effect of noise patches.

3.2 Basic notations

An image is modeled as a collection of patches. Each patch is represented by a codeword selected from a dictionary of codewords. These patches are assigned to a single latent theme, and it is possible for a theme to be shared by multiple patches. We want to ensure that patches that are both dependent and relevant to each other share the same themes. The posterior distribution for each class is sampled in the training process to obtain a probability matrix, as well as the theme distribution of the class.

The following notations are used throughout this chapter:

- A *patch* $\mathbf{x}$ is described by membership in the visual dictionary of codewords.
- An *image* is represented as a group of patches, denoted by $\mathbf{X}= \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$.
- A *category* is a collection of images.

3.3 Dependent and relevant Hierarchical Dirichlet Process

Figure 3.2 below shows the graphical model of the DHDP, in contrast to the HDP model by [67]. We have a probability measure H over a measurement space Ω and a positive real number γ. θ is a parameter that takes values in the measurement space with prior H, i.e. $\theta_k|H \sim H$. θ_k corresponds to the latent themes shared with multiple image categories. A Dirichlet Process

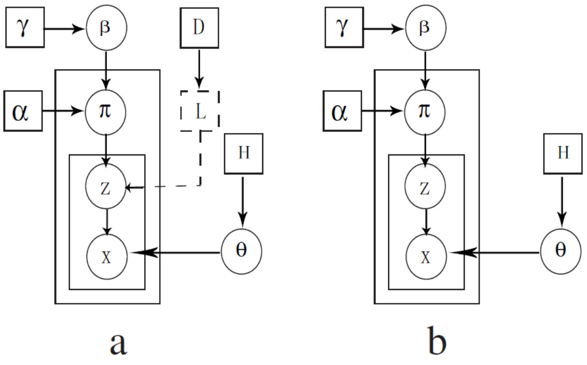


Figure 3.2: The graphical model of (a) DHDP and (b) HDP. The node "L" signifies the linkage between the patches, and their dependency.

$G_0 \sim \text{DP}(\gamma, H)$ is a distribution over measures Ω, which can be constructed as

$$\begin{aligned} G_0 &= \sum_{k=1}^{\infty} \beta_k \theta_k \\ \beta_k' | \gamma, H &= \text{Beta}(1, \gamma) \\ \beta_k &= \beta_k' \sum_{l=1}^{k-1} (1 - \beta_l') \end{aligned} \tag{3.1}$$

G_0 is an unobservable variable in the model, and β_k denotes the probability of drawing θ_k. We associate each image with another Dirichlet Process G_i, acting as the prior of the mixture models in different images. Equation 3.1 is in fact the stick-breaking construction discussed in Section 2.6.3. Since we want latent themes to be shared between different images, we force G_i to be drawn from G_0:

$$\begin{aligned} G_i &= \sum_{k=1}^{\infty} \pi_{jk} \delta_{\theta_k} \\ \pi_j | \alpha_0, \beta &\sim \text{DP}(\alpha_0, \beta) \end{aligned} \tag{3.2}$$

Denote $\mathbf{x}_{ji}$ as the i^{th} patch in the j^{th} image. The component drawn from the mixture is denoted by z_{ji} and is drawn from π_j, which is also affected by the linkage L:

$$z_{ji} | \pi_j, L \sim (\pi_j, L) \tag{3.3}$$

Then, $\mathbf{x}_{ji}$ is generated by the following likelihood:

$$\mathbf{x}_{ji}|z_{ji}, \theta_k \sim F(\mathbf{x}_{ji}, \theta_{ji}) \tag{3.4}$$

The original HDP sampling is illustrated by the Chinese Restaurant Franchise (CRF) metaphor. Note that the normal DP is illustrated by Chinese Restaurant Process (CRP) metaphor. Assume that for a set of restaurants, each restaurant has its own set of tables, and each table orders one dish. These dishes are identical throughout the set of restaurants, i.e. the dishes are shared between different restaurants. As with the CRP metaphor, in CRF, the restaurant represents the image, while the dishes represents the latent themes. Customers (corresponding to the image's patches) in each restaurant pick their table and can either order what has been ordered for that table, or choose a new table. The dependency arises as customers tend to be seated together and thus order the same dishes as other customers they know, according to the *Acquaintance Coefficient*, signifying the strength of dependencies between two patches.

In the original works of Wang et al., the linkage Λ is composed only of the strength of the dependency between two codewords from the dictionary. This work proposes to extend the linkage Λ to take the cardinality of the patches inside each image into account as well. Consider a new coefficient called the *Cardinality Coefficient*, which will redefine the strength of dependencies depending on the known category of the patches being considered. We would therefore intend to seat the customer along with other customers whom he/she is both acquainted with (high dependency) and relevant to (closer cardinality to the known category).

Let φ_{ji} denote the i^{th} customer at the j^{th} restaurant. Each φ_{ji} is associated with one ϕ_{jt}, denoting the t^{th} table of the j^{th} restaurant. In this model, this table could be regarded as the intra-image mixture component. Naturally, there can be multiple customers associated with one table, and the number of customers at a table is denoted by n_{jt}. Each ϕ_{jt} is associated with one θ_k, which is the k^{th} dish of the entire franchise ordered by the table. Similarly, multiple tables can be associated with one dish, and the number of tables ordering that dish is denoted by m_k. Denote m_jk as the number of tables in the j^{th} restaurant ordering θ_k. In all, this illustrates the “Chinese Restaurant Franchise” metaphor for a two-level HDP as mentioned by [67], which is sampled using Gibbs sampling.

Wang et al. incorporated the dependency criterion into the sampling scheme of the HDP, and also introduced the DHDP. Using the same rationale, it is possible to incorporate the cardinality relevancies into the learning process. That is, given $\varphi_{j1}, \varphi_{j2}, \ldots, \varphi_{j(i-1)}$, we can choose a table for φ_{ji} using

$$\Lambda(j, \phi_1, \phi_2) = C(\phi_1, phi_2) D(j, c(j), \phi_2) \tag{3.5}$$

$$\phi_{ji}|\phi j1, \phi j2, ..., \phi j(i-1), \alpha_0, G_0 \ \sim \ \sum_{t=1}^{T_j} n_{jt} \prod_{q=1}^{n_{jt}} 1 + \Lambda(j, \phi_{ji}, \phi^q_{\varphi_{jt}})\sigma_{\varphi_{jt}} + \alpha_0 G_0 \tag{3.6}$$

in which T_j is the number of tables occupied with customers, and $C(\cdot,\cdot)$ denotes the acquaintance coefficient between the customer that is to be seated and the customer already seated at the table. $D(\cdot,\cdot,\cdot)$ denotes the cardinality coefficient of the customer already seated at the table. $c(j)$ denotes the actual class of image j. By modifying the equation in this manner, we force the incoming customer to sit at the same table with customers that are more dependent as well as more relevant to it.

These dependencies also affect the sampling of dishes for each table. That is, given $\varphi_{11}, \varphi_{12}, ..., \varphi_{21}, ..., \varphi_{j(t-1)}$, a table φ_{jt} will sample its dishes with the following likelihood:

$$\begin{gathered} \varphi_{jt}|\varphi 11, \varphi 12, ..., \varphi 21, ..., \varphi j(t-1), \gamma, H \ \sim \\ \sum_{k=1}^{K} m_k \prod_{p=1}^{P} \prod_{q=1}^{m_k} 1 + \Lambda(j, \phi_{\varphi^p_{jt}}, \phi^q_{\theta_k})\sigma_{\theta_k} + \gamma H \end{gathered} \tag{3.7}$$

in which K denotes the number of dishes that have been ordered by previous customers. The probability matrix and the theme distribution are trained for every category using the Markov Chain Monte Carlo sampling scheme. For a given patch $\mathbf{x}$, the scheme will sample a theme. These samples, in turn, are used to train the posterior probability matrix. The theme distribution is modeled as the ratio of the number of times it is sampled over the total number of samples during the training process.

The acquaintance coefficient that is used for sampling is defined as

$$C(w_1, w_2) = \frac{R(w_1, w_2, I)}{R(w_1) + R(w_2)} - \frac{R(w_1, w_2, I)}{R(w_1) + R(w_2)^{\tau}} \tag{3.8}$$

In this equation, $R(w_1, w_2, I)$ denotes the number of times two codewords appear in the same image I, while $R(w_i)$ is the total number of times the codeword w_i appears in the corpus. τ is an experimental parameter, set at 'greater than one' as a penalty factor to prevent patches that rarely appear from becoming highly dependent.

Let $j(w)$ be the number of times the codeword w appears in an image I. The cardinality coefficient given $j(w)$, is modeled as a Gaussian:

$$D(j, c, w) = \exp(-\frac{(j(w) - \mu_{w|c})^2}{2\sigma_{w|c}^2}) \tag{3.9}$$

in which $\mu_{w|c}$ and $\sigma_{w|c}^2$ denote the mean and variance of the occurrences of codewords w appearing throughout class c, respectively. $j(w)$ is the number of times a patch belonging to codeword w appears in image j.

The Gaussian distribution is chosen to model the cardinality coefficient since it is centered on an expectation value. This is due to the fact that said coefficient reaches its highest value when the cardinality fits the knowledge basis of the image class. Additionally, with a smooth distribution, the cardinality coefficient will still return values that are less than the values obtained at the expected point, which may happen in some special cases (e.g. an image of a car taken directly from the side will yield 2 "wheel" codewords instead of the usual 3 or 4). If a distribution based solely on training data is used, it could result in cases of inaccurate data models. Modeling the cardinality coefficient as a Gaussian may not be the only suitable method. However, this thesis will only use a Gaussian to model this coefficient.

Rather than dealing with φ_{ji} and ϕ_{jt} directly, HDP samples index variables t_{ji} and k_{jt} respectively, as the index of the table and the dishes.

t is sampled by observing (3.7). In the case of sampling an unoccupied table from H, a new sample is generated from it, denoted by $k_{jt_{\text{new}}}$, using

$$\begin{aligned} k_{jt_{\text{new}}}|k &\sim \sum_{k=1}^{K} m_k\sigma_k + \gamma\delta k_{\text{new}} \\ \delta k_{\text{new}} &\sim H \end{aligned} \tag{3.10}$$

After the new dish is obtained, the index t_{ji} is sampled directly from (3.6) as

$$\begin{cases} a_0 f(\mathbf{x}_{ji}|\theta_{k_{jt}}) & \text{if } t = t^{new} \ , \\ n_{jt}^{-i} \prod_{q=1}^{n_{jt}^{-i}}(1+\Lambda(j,x_{ji},x_t^q)) \cdot f(x_{ji}|\theta_{k_{jt}}) & \text{otherwise} \ . \end{cases} \tag{3.11}$$

in which n_{jt} denotes the number of customers seated at the t-th table of the j-th restaurant. The superscript $-i$ means that customer i is excluded from consideration. If the t sampled is t^{new}, then $k_{jt_{new}}$ is inserted as a temporary value into the data structure; otherwise, it is discarded.

Similarly, we follow (3.7) to sample k by first generating a new mixture parameter $\sigma_{k_{new}} \sim H$ to anticipate the event when the sampling of k produces a new dish. The sampling of k is then done with the likelihood of

$$\begin{cases} \gamma \prod_{i:t_{ji}=t} f(\mathbf{x}_{ji}|\theta_k) & \text{if } k = k^{new} \ , \\ m_k^{-i} \prod_p \prod_q^{m_k}(1+\Lambda(j,x_{jt}^p,x_k^q)) \prod_{i:t_{ji}=t} f(x_{ji}|\theta_k) & \text{otherwise} \ . \end{cases} \tag{3.12}$$

3.4 Reducing rich-get-richer effect via $D(\cdot, \cdot, \cdot)$

If we take a closer look at equations (3.6) and (3.7), the linkage factor is signified by the $(1 + \Lambda)$ term, in which Λ is the Acquaintance Coefficient (in the original DHDP) or the product of the Acquaintance Coefficient and Cardinality Coefficient in our modified DHDP. Observe that the two $(1 + \Lambda)$ terms in (3.6) and (3.7) are tied with one or two product sequences, and are multiplied as many times as there is a customer sitting at a specific table or ordering a specific menu. The "rich-get-richer" effect is very strong in DHDP, since $(1 + \Lambda) > 1$, and when a large cluster is formed, the product term will exponentially increase, producing a huge imbalance in the sampling probability. This results in the new customer being co-opted into said cluster, which in turn leads to an even higher probability.

With increasing amount of patches evaluated in Chinese Restaurant Franchise, ensuring that customers be spread out evenly with respect to both tables and restaurants becomes a challenging problem. However, we can't simply ignore this issue, as it would eventually render the resulting probability matrix useless due to all customers being sampled into one table and one restaurant, leading to a scalar probability matrix with value 1. In fact it does not take many iterations for this to happen when the number of patches evaluated are very large. The cardinality coefficient $D(\cdot, \cdot, \cdot)$ strives to reduce this particular effect by acting as an adjusting factor to the $(1 + \Lambda)$ term by reducing the value of Λ closer to 0. In turn this dampens the growth of $(1 + \Lambda)$ factor in equations (3.6) and (3.7), which reduces the rich-get-richer effect.

To be more detailed, let us consider the case of finding a single patch belonging to the "wheel" codeword inside an image as we are considering a "Car" category. If we know that the "Car" category in our dataset tends to have between 3 or 4 "wheels" then the statistic of this image does not agree with the class being considered. In this case the cardinality coefficient will return a value closer to 0, lowering the value of Λ, and dragging $(1 + \Lambda)$ closer to 1. This will effectively slow down the rich-get-richer effect because for this particular patch (customer), its linkage would no longer affect the computation. At the very least, this particular patches will be less significant compared to those patches coming from other codewords with agreeable statistics. While it does not completely negate the rich-get-richer effect, it does considerably dampens the rate at which it happens. If HDP offers a model where a latent theme can be shared among different classes of image, then DHDP improves that model by allowing dependency between patches, but at the cost of a faster "rich-get-richer" effect. The proposed approach makes it possible to have this dependency without the adverse effect.

This property is very important in downplaying the noise patches within an image, as noise patches are random in terms of both occurrence and frequency (the number of patches in an image). At the same time, codewords that agree with the statistics are more likely to belong to the object itself. In this case, the cardinality coefficient will make sure that its contribution to the sampling of t and k in equations (3.6) and (3.7) predominates.

We call this DHDP with cardinality coefficient the modified DHDP throughout this thesis.

3.5 Approximate shape masks

Several works have shown that the visual extent of an object extends beyond the object itself [6–9]. Interestingly, in the early days of computer vision research, it was thought that the useful components of an object were only confined within its silhouette. This eventually led to the paradigm that objects should be correctly segmented from an image before they could be recognized. However, the general task of finding this location of an object, as bounded by its contours, is in fact very difficult. As time passed, and with the increasing popularity of the BoW approach, this approach was found to be unimportant for object recognition. Nevertheless, there remains support for research that uses such segmentation techniques for object recognition.

With the increase in computational power over recent years, powerful descriptors and large databases are now accessible to researchers. Along with the advances in statistical pattern recognition, the need to achieve localization before recognition has been circumvented in favor of using local region descriptors in a specific spatial arrangement [70, 78, 79]. This approach led researchers to identify an object based only on its discriminative features, which in turn led to the introduction of the bag-of-words method that was discussed in the previous chapter [80]. The method was then generalized by Csurka et al. [81], who removed its spatial verification, thus making the method rely solely on the interest point detector to extract visual "codewords" from the object. Furthermore, it was then found that the quantity of visual "codewords" was more important than the quality of the location of these visual "codewords" [82, 83], which shifted the extraction of "codewords" from salient points to a dense, regular grid. This development divorced the concept of object location from its method, mixing context and object indiscriminately.

Marzalek and Schmid's approach to generate the approximate shape mask presented in [3] is adopted in this phase of our work. Using the Kadir-Brady Saliency Detector [61], as in the previous section, salient keypoints are extracted from the image and described using the SIFT descriptor. Aggregating all keypoints from the database, one can construct the visual words dictionary using a simple K-means algorithm. Each of these key-points corresponds to a codeword, which is used to describe images.

In contrast to the previous section, a user-defined segmentation mask of the training images is provided. This binary segmentation mask separates the background and foreground within the objects, and will be used as a reference point in creating an approximate mask for a queried image. Each key-point is then assigned rectification parameters to complement the invariant description of local image features, in this case SIFT.

The design of the SIFT is such that it is invariant to both scale and rotational transformations [2], which means that the rectification parameters need to take these two transformations into account (it has been shown that it can undergo affine transformation as well, which is not incorporated into the design of this experiment). To express this mathematically, consider a feature descriptor $\mathbf{x} = d(\mathbf{p})$ of a patch $\mathbf{p}$. It is known that the descriptor is invariant to a transformation $T(\mathbf{p}, \rho)$ with ρ as the parameter of the transformation. Then, for each local image region $\mathbf{p}_i$, the parameter of this transformation $\rho_i \in D_\rho(T)$ is included in its rectification $r_i \in \mathbf{R}$. This can be written as

$$\mathbf{R} = \prod_{T \in S} D_\rho(T) \tag{3.13}$$

in which $\prod$ in this equation denotes the Cartesian product, $D_\rho(T)$ is a domain of transformation, and S is the set of transformations to which the description is invariant:

$$S = \{T : \forall \rho, r \;\; d(\mathbf{p}) = d(T(\mathbf{p}, \rho))\} \tag{3.14}$$

As the SIFT normalizes the local image region before computing the description, the rectification parameters express the normalization of the image.

Therefore, assume that we have two features $\mathbf{x}_i$ and $\mathbf{x}_j$ that are declared a match after their descriptors are considered. Both features possess their own rectification parameters, which are expressed in the transformation matrices r_i and r_j respectively. We can thus align them with a transformation matrix P_{ij}, computed as

$$r_{ij} = r_i^{-1} r_j \tag{3.15}$$

Fig. 3.3 shows an example, in toy form, of the alignment of shape masks between two images, as described by Marzalek and Schmid in [3]. Consider the

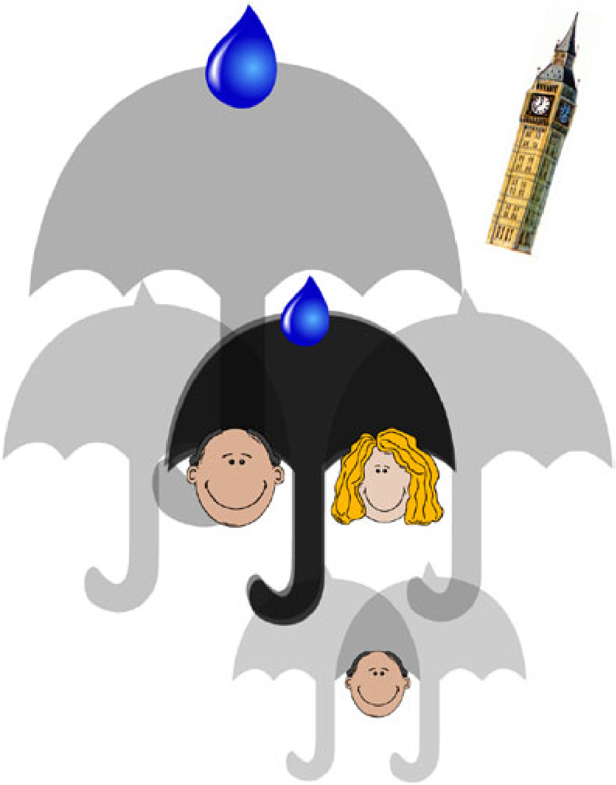


Figure 3.3: Visualization of shape mask alignment as depicted in [3].

two heads and water droplet as features, each associated with the shape mask of an umbrella. The droplet is associated with the top of the umbrella, and the heads are associated with being located either on the right or the left side under the umbrella. The presence of a droplet or a head gives us the probable location of the associated umbrella masks, depending on the position, scale, and orientation of the features. It should be noted that the size of the umbrella varies with respect to the size of the features (droplet and heads); as the size of the features changes, so will the size of the umbrella, in corresponding fashion. The "Big Ben" feature represents the background.

As shown in this example, one droplet and one head can give rise to multiple shape masks, which can be considered mistakes, but the two heads and droplet in the center of Figure 3.3 agree on the position of the shape mask, giving us the best approximate shape mask of the image, due to them being aligned with each other.

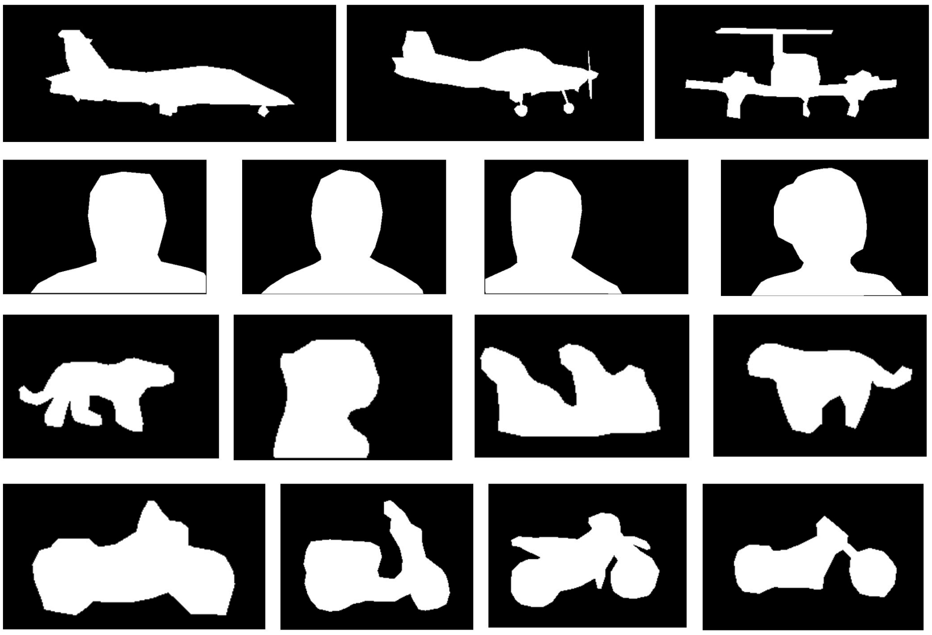


Figure 3.4: Example of user-defined segmentation.

Observe that from Figure 3.3, the alignment of shape masks can be used to produce approximate object segmentations. That is, given a set of features, each feature may produce a set of hypotheses about the localization of the object. Even if the quality of the resulting approximate segmentation varies, those masks usually tend to focus on the object.

To construct an approximate segmentation, the following steps are performed:

3.5.1 Compute sparse local features

For a given image, compute a set of sparse local features using the saliency detector and describe them using SIFT. A user-defined set of segmented training images is provided as a ground truth to help the system learn the shape mask from a training viewpoint. In addition, as we have the data for the ground truth, it is possible to filter out the background features in the training image. The system will then learn the rectification parameters of each feature.

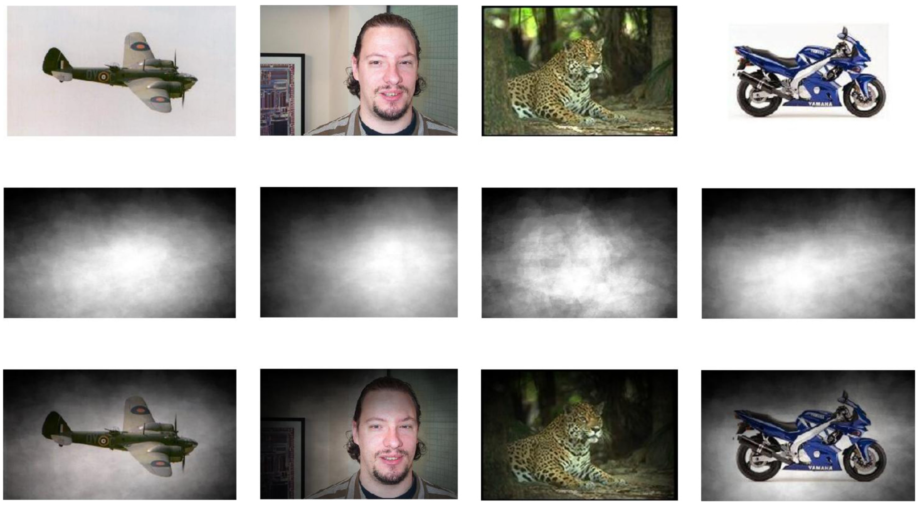


Figure 3.5: Example of the approximate mask generating process. The first row shows the original image, the second row shows the approximate shape mask, and the last row shows the multiplication of the two. The results were based on 100 training shape mask datasets created manually.

3.5.2 Cast hypotheses

Pairs of similar features obtained from Section 3.5.1 generate hypotheses about possible object locations, by applying the rectification parameter to the shape mask that correlates with the features saved in the training database. These ground truth masks will then be scaled, rotated, or shifted accordingly, depending on the pair's rectification parameters.

3.5.3 Stack hypotheses

As a large number of closest features is collected, numerous hypotheses will be formed. The system will then try to aggregate these hypotheses to create the approximate shape mask. Each hypothesis is summed together, weighted by a Gaussian function (with zero mean and $\sigma = 0.15$ standard deviation) of the distance between the training and test features. The sums of all hypotheses are then normalized to unity, to create the approximate segmentation mask $M(x, y)$.

Figure 3.5 shows some examples of results generated using this algorithm. With the ground truth data from the shape masks, it is possible to filter out any detected points of interest that lie outside the shape mask. As such, this process can focus on specific points of interest, with fewer noise patches. There will be no changes, in terms of sampling and computation of distribution, to the Dirichlet Process (equations (3.6) and (3.7)).

3.6 Object Recognition

The previous process trains the posterior probability matrix $p(\mathbf{x}_j|\theta_j)$ for every object category. Let us say that for an image I containing the set of patches $\mathbf{X}$ to be categorized, it will have $|X|$ patches extracted from the detector. In contrast to the original recognition method by Wang et al., this proposed method treats these patches individually, instead of aggregating their probabilities as independent events. To incorporate knowledge of cardinalities into the recognition steps, a new recognition paradigm is proposed. First, we calculate the probability $p(\mathbf{x}_k|c)$ for each class using

$$p(\mathbf{x}_k|c) = \sum_i p(\mathbf{x}_k|\theta_i)p(\theta_i|c) \tag{3.16}$$

A voting system is then introduced. For each class c, define a favored weight $V(c)$ based on the probability in equation (3.16) and the cardinality coefficient given in equation (3.8):

$$V(c) = \sum_{k=1}^{|\mathbf{X}|} p(\mathbf{x}_k|c)D(I, c, w_{\mathbf{x}_k}) \tag{3.17}$$

In equation (3.17), $w_{\mathbf{x}_k}$ denotes the codeword membership of $\mathbf{x}_k$. Equation (3.17) is expanded further to take the generated shape mask into account. Let $M(x, y)$ be the shape mask that gives a score towards the localization value of each feature, given its position (x, y) in the image. The voting system in equation (3.17) is modified into

$$V(c) = \sum_{k=1}^{|\mathbf{X}|} p(\mathbf{x}_k|c)D(I, c, w_{\mathbf{x}_k})M(x, y) \tag{3.18}$$

in which x and y denote the position of $\mathbf{x}_k$. All the components needed to calculate the voting coefficient are available from the previous process.

The categorization comes from finding the category which yields the highest voting coefficient

$$\text{class detected} = \arg\max_c V(c) \tag{3.19}$$

Incorporating the cardinality into the linkage for DHDP helps improve the model as it accounts for the fact that some categories would have similar numbers of codewords. The theme distribution $p(\theta_i|c)$ is modeled as the ratio of the frequency of the theme sampled to the total number of samples.

3.7 System implementation

As in Wang et al.'s work [1], each image is represented as a collection of patches, as discussed in Section 3.2. The feature point is extracted using the Kadir-Brady scale-saliency detector [61], in such a way that the best 30-40 patches per image are extracted. It should be noted that the image is resized such that all images have exactly the same height for uniformity. Each patch is resized into a 48×48 pixel window, and divided into four 24×24 sub-regions. A SIFT-like descriptor is used by utilizing an 18-bin orientation histogram. Hence, each feature is described by a 72-dim vector. It is also desirable to reduce computational cost by reducing the dimensionality to a 15-dim vector using PCA. The PCA basis is obtained from the Caltech-101 "Background" category by sampling patches at regular intervals. For vocabulary construction, K-means clustering is used to group these vectors to form a large "Codewords Dictionary". Classification of feature vectors is done based on the center-points found by the K-means

Algorithm 1 Learning process

```
Input: Image database
Output: Probability matrix $p(\mathbf{x}_j|\theta_j)$
 1: for all image class $c$ do
 2:     for all image $I_{jc}$ do
 3:         extract $\mathbf{X}_{jc}$ from $I_{jc}$
 4:         if using approximate shape mask then
 5:             get user-defined mask $M_{jc}(x, y)$
 6:             filter $\mathbf{X}_{jc}$ using $M_{jc}(x, y)$
 7:         end if
 8:     end for
 9: end for
10: run K-Means to form dictionary $\mathbf{V}$
11: assign $\mathbf{x}_{jck} \in \mathbf{X}_{jc}$ to an entry $\theta_k \in \mathbf{V}$
12: precompute $C(\cdot, \cdot)$ using (3.8)
13: set max_iter
14: for iter = 0 to max_iter do
15:     for all $j$ and $i$ do
16:         sample $t$ following eq. (3.10) and (3.11)
17:     end for
18:     for all $t$ and $j$ do
19:         sample $k$ following eq. (3.12)
20:     end for
21: end for
22: calculate $p(\mathbf{x}_j|\theta_j)$ from the samples
```

algorithm.

Algorithm 1 summarizes the Learning part of our modified DHDP that we have discussed so far:

- *Extracting local regions* (Steps 1 to 9) is done using the Kadir-Brady scale-saliency detector. All hits and their scores are collected, with the top 30 to 40 patches picked depending on their scores. These patches are then described using SIFT, and their dimensionality reduced using PCA. If we want to use approximate shape mask, steps 4 to 7 will be run. In these steps, the known ground truth of the object's shape and location from the dataset is used to discard all local regions that fall outside the shape mask.
- *Cluster extracted regions to form dictionary* (Step 10) is done by clustering the gathered patches using the K-means algorithm. The learned

Algorithm 2 Recognition process

```
Input: Image $I$, dictionary $\mathbf{V}$ and $p(\mathbf{x}_j|\theta_j)$
Output: Assigned class $c_r$
1: extract $\mathbf{X}$ form $I$
2: assign all $\mathbf{x}_i \in \mathbf{X}$ to a codewords in $\mathbf{V}$
3: if using approximate shape mask then
4:     compute $M(x, y)$
5:     assign coefficient to each $\mathbf{x}_i$ based on $M(x, y)$
6: end if
7: recognize class $c_r$ using eq. (3.16) - (3.19)
```

cluster centers are used as codewords.

- *Represent regions as codewords* (Step 11) is then performed on each of the gathered local image regions/patches. Hard-assignment is used to associate each patch with exactly one cluster center.
- *Computing "Acquaintance Coefficient"* (Step 12) is then possible since we now have complete knowledge of patches' identities from the previous step. $C(w_1, w_2)$ can be pre-computed prior to DHDP sampling and be represented as a matrix. Note that the cardinality coefficient is dynamic and constantly changes over the course of DHDP sampling.
- *Learn probability matrix using DHDP and "Cardinality Coefficient"* (Steps 13 to 22) is done by utilizing equations (3.10)-(3.12). Set every patch $\mathbf{x}_i$ in image j as the customer φ_{ji}, each seated at a table $\phi_j t$ ordering $\theta_{k_{jt}}$, initialized as their codeword membership from $\mathbf{V}$. Having fully established the needed identities for the Chinese Restaurant Franchise process, we iterate sampling for a certain number of times, utilizing Gibbs sampling in the form of equations (3.10)-(3.12) for all customers and tables within the set, for each iteration. The final membership for tables and dishes is computed as the probability matrix.

Having done the learning step, we have trained the classifier to help us recognize the class of an input image I. For the recognition process, Algorithm 2 describes the steps in detail:

- *Extracting local regions* (Step 1) is done in the same way as in Algorithm

1. The only difference is that in Recognition process, extraction is done only for the input image, and we do not have the user-defined shape mask to assist in filtering noisy patches.

- *Represent regions as codewords* (Step 2) is then performed on the extracted patches using the dictionary learned in Algorithm 1 using hard-assignment.
- *Compute approximate shape mask and assign score* (Steps 3-6). This is the approximate shape mask generation step as explained in Section 3.5 (finding $M(x, y)$). Based on the computed shape mask, a coefficient is assigned to each patch as weight for the last step, as in equation (3.18).
- *Recognition of class c_r* (Step 7) is done using equation (3.16) to (3.19).

3.8 Experiment and results

3.8.1 Experimental settings

The experiment was carried out on Caltech-4 datasets. This dataset contains the classes "Airplanes", "Faces", "Leopards", and "Motorcycles". Each class consists of 800, 435, 200 and 798 images respectively. We tested our proposed method with two settings: without approximate shape mask (hereby denoted as E1) and with approximate shape mask (E2).

For E1, the first 100 images from each class were used to train the system to construct a dictionary of 1200 codewords, while another 100 images from each class were chosen randomly, to test recognition performance. The parameters of α_0,γ, and τ were set as 0.1, 1, and 1.2 respectively, while the sampling was iterated 100 times.

For recognition of salient regions, cases with patches that did not belong to an object but were included in the training and recognition process (as illustrated in Figure 2.2) were considered. In Figure 2.2, the patches coming from the trees in the background are included in the detected patches. However, the keyword "trees" is not a descriptor of an airplane. By taking note of the occurrence of the keywords across training images, we can assign the weight of relevance to each keyword from a given category using $D(j, c, w)$.

For E2, the number of iterations in the Modified DHDP is set to be considerably smaller (20 compared to the 100 of E1) as the initial number of codewords is significantly smaller (1200 without approximate shape mask vs 300 with the mask). Hence, it will take fewer iterations for the training step to be disturbed by the rich-get-richer effect. The experiments were run multiple times, and it was found that a number in the region of 20-30 iterations was acceptable.

User-defined segmentations of training images were collected manually using MATLAB, with users asked to specify the boundary box of all images in the database (2233 images in total), as a training dataset and ground truth for evaluating the approximate segmentation. The three information sets of $C(\cdot, \cdot)$, $D(\cdot, \cdot, \cdot)$, and $M(x, y)$ were combined, to assist the training and recognition as depicted in the previous sections.

3.8.2 Theme distribution

In E1, the sampling of Modified DHDP was run 100 times with the first 1200 codewords from the dictionary as the initial "dish". The Modified DHDP merged these codewords into 118 latent themes to be shared among the four object categories. While it would have been possible to continue with further iterations to reduce the number of latent themes, this was not done in this experiment due to the fact that as the iterations proceed, the "rich-get-richer" effect is more likely to occur.

It was found that even though it is possible to push the number of latent themes to a low number (as low as 20), the "rich-get-richer" effect will be in play. Latent themes with large numbers of members assigned to them tend to absorb other themes into themselves, especially in the sampling k step of the Modified DHDP (as described in Section 3.3). This leads to one latent theme

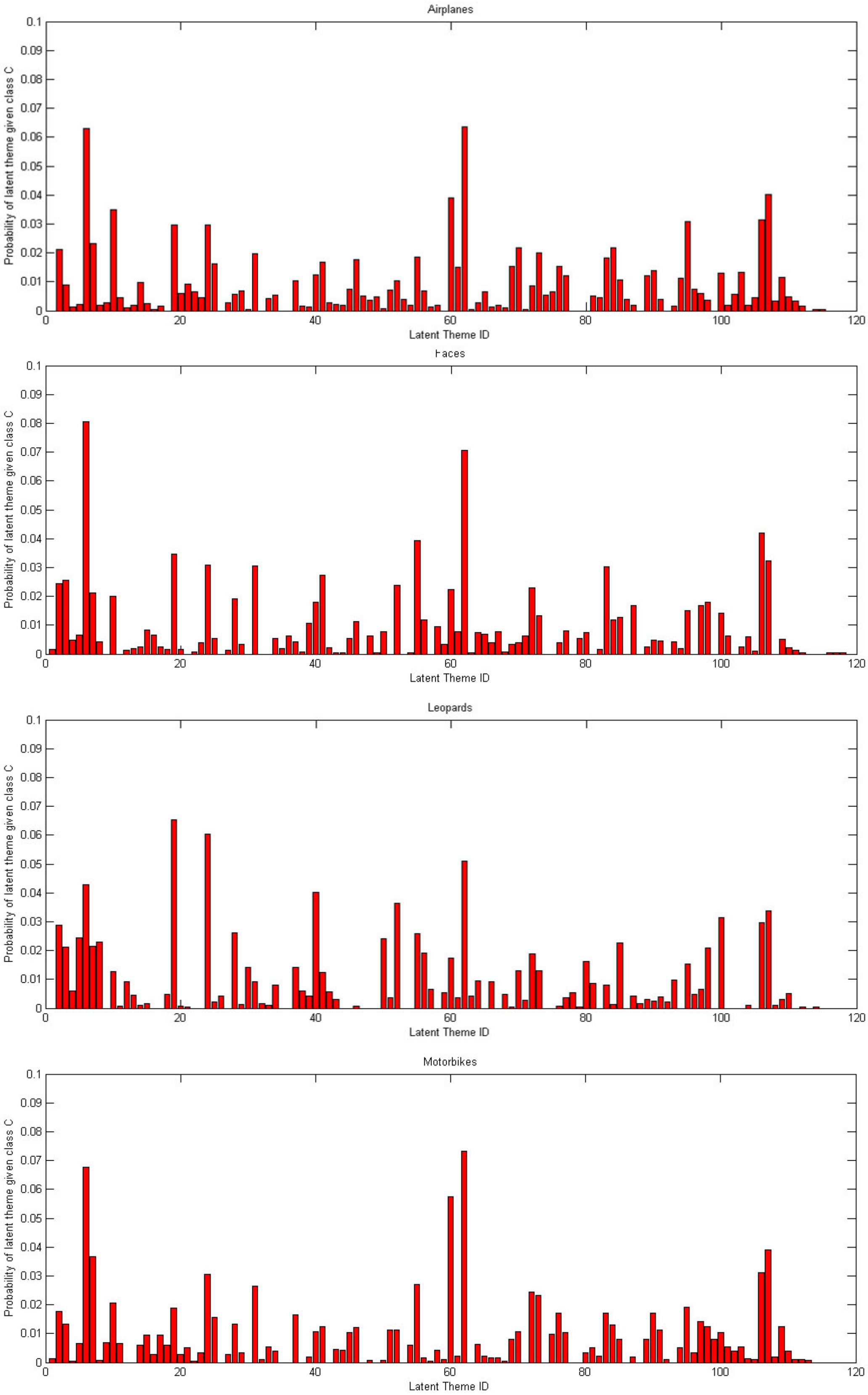


Figure 3.6: Theme distributions of the four classes in Caltech-4 from E1 test case with 118 latent themes.

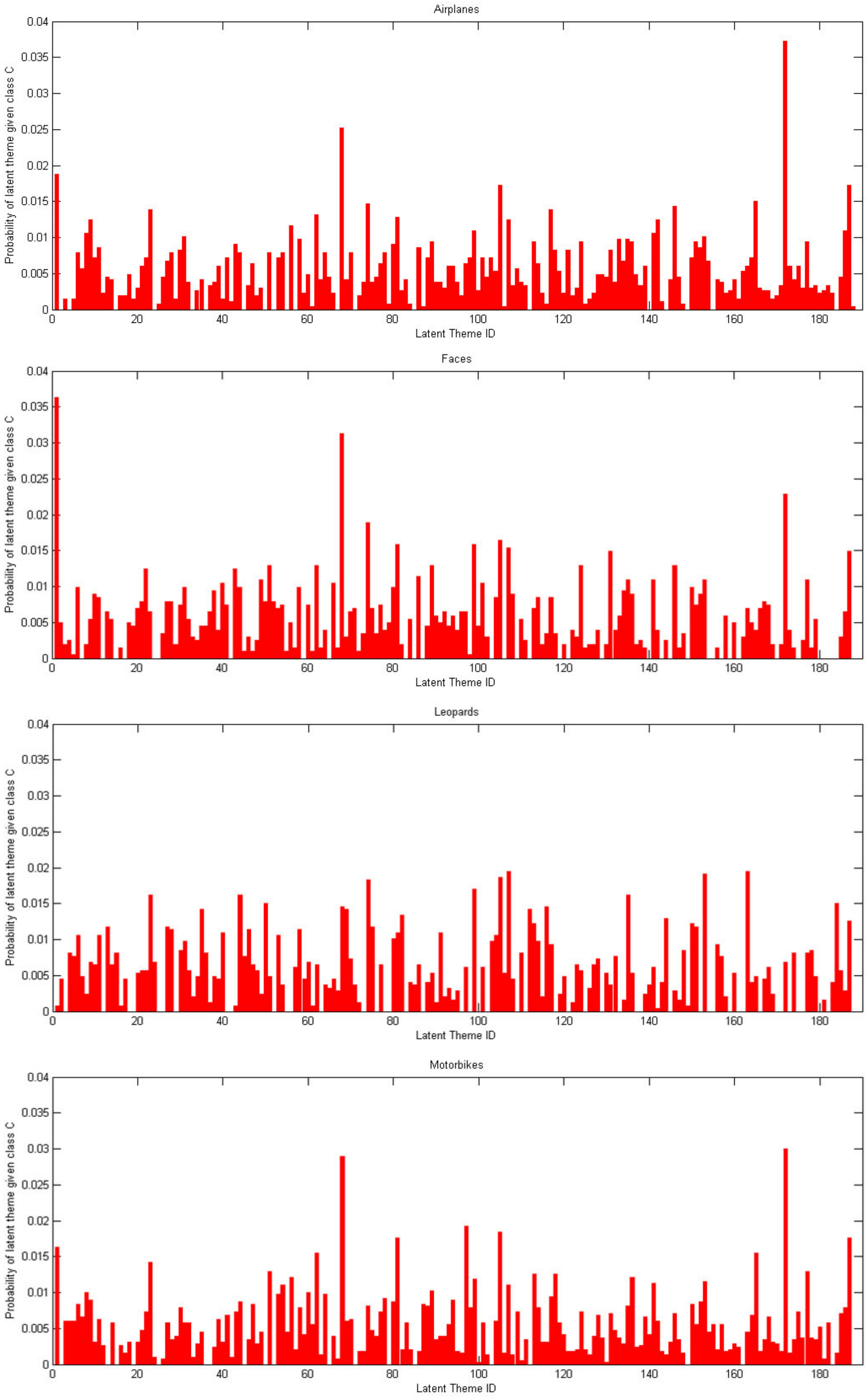


Figure 3.7: Theme distributions of the four classes in Caltech-4 from E2 test case with 188 latent themes.

with a considerably large membership. This high membership makes the value of $f(\mathbf{x}_{ji}|\theta_k)$ disproportionately large for a specific k value, compared with other latent themes. Hence, it will be more likely for a latent theme to get co-opted into this specific k, and eventually the cluster will become too big and much less informative.

It was found that stopping at 100 iterations (which gave 118 latent themes) gave considerably good results. It should be noted that the number of codewords was reduced by more than 90% (from 1200 to 118), which made calculations more efficient; however, at the same time, the distinction between the four categories in the distribution of latent themes was visible. Figure 3.6 shows the theme probability distribution by object classes $p(\theta_i|c)$ for E1. While we can see that some peaks and valleys of theme distributions tend to be similar across classes, their probabilities can differ considerably. One such example is evident from the “Leopards” class in the distribution of Theme #19 and #24, which exhibits a considerably higher probability value than the other three classes.

These differences in distributions between themes are the features that will be used in the categorization of test images, as explained in Section 3.4. It should also be noted that by the corollary of equation 3.19, we are basically using the winner-takes-all paradigm in our approach, assigning test images to class c that produce the best favored weight $V(c)$.

As for E2, with the inclusion of approximate mask the Modified DHDP scheme started with 300 codewords from the dictionary as the initial "dish"; the Modified DHDP merged the codewords into 188 latent themes to be shared among the four object categories. This number is slightly higher than the latent themes obtained from E1 because a smaller number of initial codewords were used. This made it easier for the cluster to merge, since it was more likely that the codewords were similar to each other.

While the latent themes were 50% greater in quantity than those obtained in E1, the theme distributions still show distinctive features throughout the four different classes. Figure 3.7 shows the theme distribution obtained from the DHDP iterations.

As with the previous process, these differences in theme distribution will be the features used in the categorization of test images, as explained in Section 3.6.

3.8.3 Cardinality coefficient reduced rich-get-richer effect

E1 was tested and compared with the framework provided by Wang et al. in [1], which confirmed the hypothesis that noise patches decrease the overall performance of DHDP, and, in turn, the accuracy of object category recognition.

From the experiment, it was found that without the contribution of cardinality coefficients, the rich-get-richer effect is very severe. In other word, it takes fewer steps of iterations for DHDP to merge codewords into a smaller number of latent themes, but in these latent themes, one latent theme tends to dominate the others; hence the data set loses information. It has been shown that on average, it takes 40 iterations for DHDP to start showing the rich-get-richer effect, compared with 150 iterations of the Modified DHDP. When cardinality coefficient was included, it was found that although the speed of the merging of codewords was slower, the resulting theme distributions were less prone to the "rich-get-richer" effect.

3.8.4 Approximate shape mask

Examples of approximate shape mask results are shown in Figure 3.5. While it is clear that the resulting approximate shape mask is far from an ideal segmentation, the resulting segmented images are close enough to our intended results, after filtering out the background clutter and giving weight to each individual key-point.

It is interesting to observe that the approximate segmentation tended to filter out pixels at the top of the images. This is actually not a surprising fact, as the top of the image usually shows only the background, which is not directly informative to the object of interest within the picture. However, it is encouraging to see that this algorithm managed to catch this trend and reflect it in the resulting shape mask.

The approximate shape mask, however, encountered much difficulty trying to create a useful approximation for the "Faces" class. This was mainly because some of the images in this class had bad contrast (some faces were very unclear and some pictures even focused more on the background instead of the face), or their backgrounds were cluttered with a large number of objects. The high variations in the shapes of the objects were also a significant factor affecting the accuracy of the shape mask, as reflected in the accuracy of recognition.

The generated shape mask might closely resemble the saliency map from the key-point detector. This is to be expected, as we use key-points generated from the salient point detector to generate the shape masks. However, the main difference here is that while the salient point detector looks for all points that stand out in an image, the proposed method generates a shape mask from a selection of only the 30-40 most salient points detected, in the hope of ruling out false positives (background and noise patches) from the overall recognition. While it is true that not all the tops of images depict the foregrounded object, it was found that this led to a lower number of background patches being included in the training set. In a way, the shape mask can be seen as a pruned subset of the saliency map from the key-point detector, albeit one that focuses more on the predicted location of foreground objects.

It is worth mentioning that the approximate shape mask would likely fail when the object inside an image is not dominant, or when there are several salient objects within an image. Whenever this happens, we would see a weak mask over several regions inside the image, as illustrated in the third column of Figure 3.5. This, however, is not a big limitation for the Caltech dataset, since the images in that dataset often contain dominant objects. On the other hand, this approximate shape mask could be used to assist recognition in later stages (in example, as a preliminary proposal for objectness of an image as presented in [84]). Object window candidates can be narrowed down from these preliminary proposals, to obtain a more accurate location of the object.

In terms of the "rich-get-richer" effect, we found that anything beyond 30 iterations of E2 has a high risk of the effect occuring. However, this does not mean that E2 is worse than E1 in terms of reducing this effect. It should be noted that the number of starting codewords in E2 is merely a quarter of the number used in E1. When the number of codewords was increased to the level of E1 (by lowering the KB detector saliency threshold so we have more patches to cluster), it was found that E2 could run up to 130-150 iterations without risking the "rich-get-richer" effect. This represents approximately 20% more resistance to the effect, compared with E1.

3.8.5 Accuracy

In terms of recognition accuracy, the experiment results show significant improvement from the framework used in [1]. The algorithm is tested using 100 randomly selected test images from each class, and detects these images using the theme distribution trained by the DHDP and Modified DHDP. The overall results showed increased accuracy, from 70.5% with the framework from [1] to 76.75% for this algorithm. Table 3.1 shows the recognition rate for each class. Figure 3.8 also shows these results as a bar chart for a more graphical representation of the results.

An interesting point arise from the result of the normal DHDP. While it seems that average accuracy increased, the accuracy of the "Airplanes" class dropped significantly from 91% using DHDP, to 68% using the proposed algorithm. This may seem odd initially, but we should be aware that the recognition scheme used was basically a winner-takes-all algorithm. In other words, all recognitions will assign a class label to a test image based on the evaluation of

Table 3.1: Detailed class-by-class performance comparision between results obtained by normal DHDP mentioned in [1], DHDP with cardinality coefficient (E1), and DHDP with cardinality coefficient and approximate shape mask (E2).

Object Classes	DHDP	E1	E2
Airplanes	91%	68%	96%
Faces	68%	83%	70%
Leopards	55%	86%	70%
Motorbikes	68%	70%	91%
Average	70.5%	76.75%	81.75%

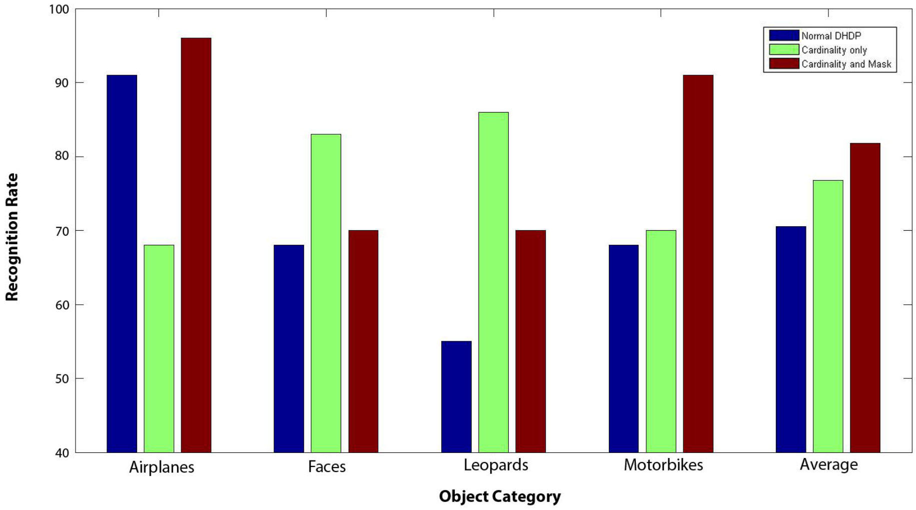


Figure 3.8: Performances comparison between Normal DHDP, E1, and E2.

$V(c)$. It was found that for the result obtained using normal DHDP, there was a tendency to assign errors from any of the classes to the "Airplanes" class. This explains the very high success ratio of the "Airplanes" class (91%) compared with those of the other classes (68%, 55%, and 68% for "Faces", "Leopards", and "Motorbikes" respectively). In other words, the detector inherently favored the "Airplanes" class in its recognition.

This biased recognition was corrected in E1, as the theme distribution is more distinguishable and evident from the result obtained: 68%, 83%, 86%, and 70% recognition rates for "Airplanes", "Faces", "Leopards", and "Motorbikes" respectively. The differences in accuracy between classes are less striking, which shows that the recognition was less biased than the recognition obtained from normal DHDP.

This result comes from the addition of the cardinality coefficient, which suppresses the effect of noise patches in the recognition step. Any patches that do not agree with the statistics will be downplayed by $D(\cdot,\cdot,\cdot)$, and object patches (which usually agree with the statistics) will be prioritized by the same coefficient. This ensures that the contribution of informative patches will be more dominant than those of noise patches. This becomes very important, especially in the bag-of-words method, where we pool salient patches in an indiscriminate fashion; naturally, more and more noise patches may arise from an image containing a cluttered background, poor contrast, or multiple objects.

In fact, the cardinality coefficient $D(\cdot,\cdot,\cdot)$ acts as a segmentation procedure for patches detected by the keyword recognition system. This finding encourages us to investigate the effect of segmentation on object recognition - more specifically, how we are able to fit a segmentation method into a non-parametric Bayesian framework. It should be clear by now that the DHDP training process can be computationally costly; adding accurate segmentation would only increase the complexity of it, not to mention that obtaining an accurate segmentation would be very hard to achieve.

Table 3.1 also shows a detailed comparison of the recognition results of E1 and E2. Overall, a significant increase can be observed for E2 (up to 81.75% correct matches), compared with the result obtained from E1. This result confirms the hypothesis that while the bag-of-words method offers acceptable results with a relatively low computational cost, it ignores the information that is possessed by the spatial location of patches, and the relationship between patches.

It is interesting, however, to observe that the method that utilizes an approximate mask had very high success rates with rigid objects like "Airplanes" and "Motorbikes" (96% and 91% respectively). This is a very significant increase from the result of E1, where the two classes had success rates of only 68% and 70% respectively. The success rate for non-rigid objects, however, dropped to 70% in both the "Faces" and "Leopard" classes, compared with 83% and 86%

of correct matches for E2, respectively.

As discussed earlier, the quality of the approximate shape mask was relatively lower in the "Faces" class than in the other classes. The non-rigid objects are harder to approximate since they have a high degree of freedom in their shapes. A human head in an image can have various slants and different facial shapes, while a leopard can show a different pose in every image. The approximate segmentation, in this case, would be a very rough estimate with a broader support area, making the standard deviations of $M(x, y)$ throughout the image more widely spread out. Hence, it is natural that the coefficient for these two classes plays a less significant role.

However, for the rigid objects, it is evident that the shape masks are effective enough to produce a high recognition rate. This is not surprising, as variations of shape within these objects are low. Even with different models of airplanes and motorbikes, the shapes within the two classes remain generally similar.

This result leads to an interesting conclusion: localization works well with rigid objects. In other words, spatial information has a significant contribution to objects that have low variations in shape, but contribute less to non-rigid objects. As information about object rigidity would be readily available before training and recognition, it may be possible to switch between the two recognition methods to achieve better performance. In addition, and interestingly, the two algorithms (with or without segmentation masks) have the same training process. Hence, with the same initial conditions, both algorithms possess the same latent theme distributions. However, some adjustments should be made to ensure that the favored weights, $V(c)$ in equation (3.17) and $V(c)$ in equation (3.18), are normalized, so that we can directly compare the two.

3.9 Concluding remarks

Throughout this chapter, we have investigated the effect of adding additional information to the Hierarchical Dirichlet Process with the motivation of reducing its shortcomings in terms of the "rich-get-richer" effect. The additional information has two sources. The first is the knowledge that certain codewords tend to appear in a consistent arrangement. Utilizing the cardinality coefficient, we model this relationship into the learning process. The second sorce of additional information is the spatial information in the form of the approximate shape mask, as background clutter is the root cause of the "rich-get-richer" effect. Using this approximate shape mask, the negative effect of HDP is slowed down considerably.

At the end of our experiment, however, we found that HDP-based approaches were not performing as well as other approaches simultaneously conducted. The lesson from the research described in this chapter is that spatial information can be a powerful addition to image representation. This served as the motivation for the research described in subsequent chapters.

Chapter 4

Overlapping Spatial Window

4.1 Introduction

As discussed in previous chapter, knowledge of the spatial information of patches can be beneficial to the discriminability of the image descriptor or the learning process of the classifier. While we are able to produce encouraging result from non-parametric Bayesian learning system, it was clear at that moment that sparse coding based recognition paired with Support Vector Machine (SVM) produces more promising result. Interestingly enough, these approaches utilizes Spatial Pyramid Matching (SPM) to incorporate spatial information into "bag-of-words" (BoW) method.

SPM was first proposed by Lazebnik et al. [13] to extend the traditional BoW approach, with the aim of incorporating spatial configurations into image representation. Under SPM, each image is described by a concatenation of multiple histograms based on the spatial pyramid, built on L layers of image partitions. The l^{th} layer (where $l \in \{0, 1, \ldots, L-1\}$) is obtained by dividing the image into $2^l \times 2^l$ disjoint sub-windows (with a typical setting of $L = 3$). Thus, a pyramid is defined as a collection of sub-windows, with each sub-window acting as a “bag-of-words”. From each sub-window, a histogram is extracted by assigning patches of codeword entries from a learned dictionary (encoding). These histograms are then concatenated to produce the SPM representation.

As encoding of patch descriptors into discrete dictionary entries shifted from hard-assignment (assigned to a single dictionary entry) to soft-assignment (assigned to several dictionary entries using membership coefficients), combining soft-assignment with SPM representation proves to be very powerful, especially when coupled with sparse coding. With sparse coding, the membership of each image patch can be assigned to more than one entry, while at the same time it forces the membership coefficient of each patch to have few non-zero elements. This approach is known as Sparse Coding SPM (ScSPM) [25] and it has since become the foundation of various state-of-art object recognition research.

However, only few researchers have challenged the usage of disjoint subwindows in SPM. Overlapping spatial windows, at half the original size, have been proposed by Ergul and Arica [85] for scene recognition, but it retains the same window size for each pyramid level. This leads to an increased number of sub-blocks, and consequently increases its memory cost (the size of the image representation is tripled, as 59 sub-blocks are used in computing these features, compared to 21 with traditional SPM at $L = 3$). Yan et al. [58] used dense spatial sampling to replace SPM with sub-blocks of variable sizes. The differently-sized spatial windows may overlap with each other, but its memory complexity considerably increases with the higher degree of variability.

This work is inspired by the fact that it is very rare for humans to examine an image in disjoint parts. More often than not, we examine local regions in our field of vision that are likely to overlap with each other. This work proposes to extend the traditional SPM further afield by introducing two types of overlapping windows: overlapping rectangular windows SPM (OWSPM) and overlapping circular windows SPM (CWSPM). By utilizing overlapping subwindows, we inherently improve the probability that a sub-window will enclose a higher proportion of an object, leading to increased discriminability of image representation, while still at the same memory cost. This work is the first to propose the usage of overlapping spatial windows while retaining the same storage and computational cost.

This proposed concept is tested on both ScSPM and LLC (Locality-constrained Linear Coding) frameworks. A variety of popular databases (Caltech 101, Caltech 256, and 15-Scene) is used in experiments, achieving up to 3.68% improvement in recognition rates. Further experiments lead us to an interesting discovery, where it is found that the $l = 2$ layer contributes to the majority of information used in recognition. While obviously the full pyramid still provides the best recognition result, the experiments show that using only the overlapping layers at $l = 2$ can give a better result than the traditional SPM with all three layers. By doing so, it is possible to save 24% of memory consumption (a resource that will be used extensively in the training process) while achieving a better result altogether.

This chapter's contributions are thus summarized in three points:

1. Introduction of overlapping rectangular windows and their optimal size of overlap to increase the rate of recognition of traditional SPM,
2. Introduction of overlapping circular windows, and
3. Bypassing the first two layers of traditional SPM.

4.2 The design of overlapping sub windows

In the proposed concept, the rectangular overlapping spatial window is designed such that for all layers with $l > 0$, the number of spatial windows needed to represent an image remain unchanged. Imagine the case for $l = 1$ in Figure 4.1. Only the sub-windows on the top half of the image are shown for clarity.

Let h_I, w_I, h_s, w_s be the height of the image, width of the image, height of the sub-window and width of the sub-window, respectively. Let A_s be the area of S_1. As all the windows have the same size, S_2 also have the same area A_s. Let A_ω be the area of overlap between S_1 and S_2. We define overlap parameter

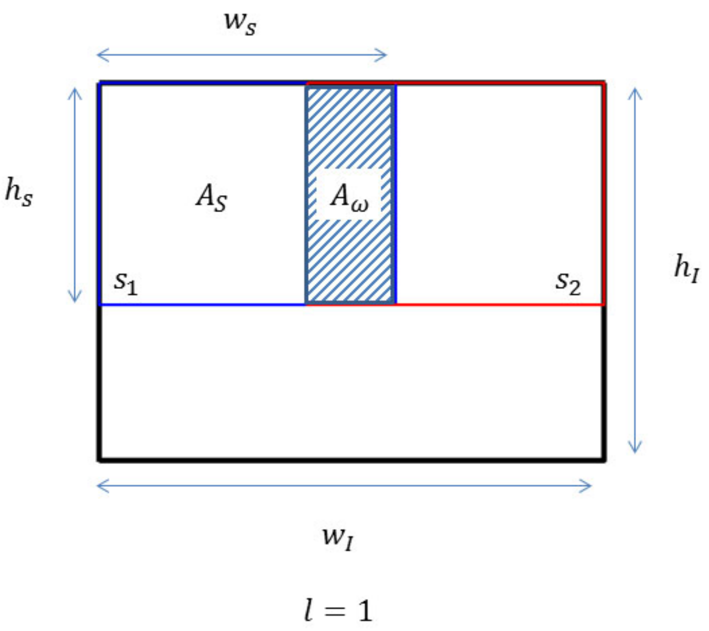


Figure 4.1: Illustration of proposed sub-window division at layer $l = 1$. Bottom two sub-windows are omitted in the figure for clarity.

ω as the ratio of the overlapping area between two adjacent sub-windows over the area of a single sub-window in the same layer (adjacency is defined in a top-bottom and side-by-side fashion), i.e. for all $l > 0$

$$\omega = \frac{A_\omega}{A_s} \tag{4.1}$$

By definition the width of the overlapping area in Figure 4.1 will be $w_\omega = \omega \times w_s$, and since $w_I = (2^l \times w_s) - ((2^l - 1) \times w_\omega)$ a simple mathematical manipulation yield:

$$\begin{aligned} w_s &= \frac{w_I}{2^l(1-\omega)+\omega} \\ h_s &= \frac{h_I}{2^l(1-\omega)+\omega} \end{aligned} \tag{4.2}$$

All patches inside the sub-window would then be pooled together to form the histogram, using max-pooling as shown in equation (2.4). As with SPM, the image representation is then constructed by concatenating all resulting BoW representations from all sub-windows. This approach is referred as the overlap-

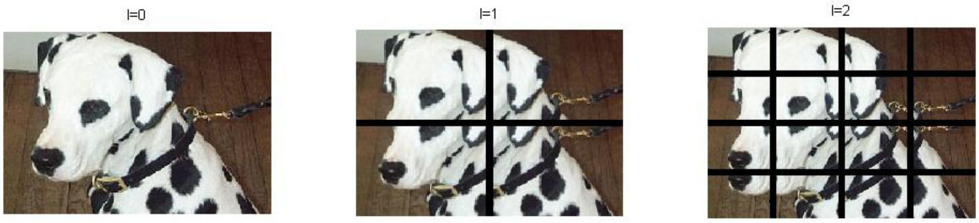


Figure 4.2: Example of OWSPM sub-window definition.

ping rectangular windows SPM (OWSPM) scheme throughout this thesis. Figure 4.2 illustrate the OWSPM sub-windows on a Dalmatian image from Caltech 101 dataset.

4.3 Finding the best-performing ω

As mentioned in Section 4.1, we use ScSPM and LLC as the baseline of our test on overlapping sub-window scheme.

Sparse Coding SPM (ScSPM) [25] is a “bag-of-words” approach to image representation utilizing the SPM model, with two main features. First, as the name implies, instead of using hard-assignment in encoding the patches, ScSPM encodes these patches using soft-assignment; the end result is forced to be sparse by following equation 2.2 in Chapter 2. The other main feature is that instead of using average pooling for each spatial window, ScSPM utilizes maximum pooling.

Locality-constrained Linear Coding (LLC) [23] is an extension to ScSPM. The model adds a new rule to the sparsity constraint in encoding patches, by forcing it to be assigned to the codeword center that is close to the patches.

4.3.1 Building codeword dictionary

The sparse coding in ScSPM is obtained from optimizing equation (2.2) in Chapter 2. 50000 random patches are extracted from various images (i.e. the "Background" class in Caltech 101 or Caltech 256) as training patches, which are then described using SIFT. These patches are then clustered using the K-

Algorithm 3 Finding $\mathbf{V}$

```
Input: Random $M$ patches from a dataset, $\mathbf{p}_m$.
Output: Dictionary $\mathbf{V}$.
for all Patches $\mathbf{p}_m$ do
    Extract SIFT descriptor $\mathbf{x}_m$
end for
Cluster using K-means algorithm to $K$ clusters.
for iter=1 to max_iter do
    for all Patch descriptor $\mathbf{p}_m$ do
        Solve equation (2.2) by fixing $\mathbf{V}$.
    end for
    Solve the equivalent of equation (2.2) obtained by fixing $\mathbf{U}$:
    $\min_{\mathbf{V}} ||\mathbf{X} - \mathbf{VU}||$.
end for
```

means algorithm as an initial guess for $\mathbf{V}$. Following that, the optimization of equation (2.2) is done iteratively, by fixing $\mathbf{V}$ and $\mathbf{U}$ in an alternating pattern. The details can be observed in Algorithm 3.

4.3.2 Obtaining ScSPM/OWSPM representation

The dictionary $\mathbf{V}$ obtained in Section 4.3.1 is then used to encode the patch descriptor *sparsely* with equation (2.2). Given a specific SPM configuration (be it traditional SPM or our proposed OWSPM), the sparsely coded features are pooled using *maximum-pooling* as described in equation (2.4), based on the window memberships. The resulting window descriptors are then concatenated to form the *mid-level representation* of an input image. The process of finding the mid-level image descriptor $\mathbf{Z}$ is shown in Algorithm 4.

4.3.3 Training classifier using multi-class linear SVM

This section describes the implementation of linear SVM used Chapter 4 to 6. Given training data $\{(\mathbf{Z}, y_i)\}_{i=1}^{n}, y_i \in \mathbb{Y} = 1, ..., C$, a linear SVM aims to learn C linear functions $\{\mathbf{w}_c^T \mathbf{z} | c \in \mathbb{Y}\}$ such that, for a test image descriptor $\mathbf{Z}$, its

Algorithm 4 Representing image I using ScSPM/OWSPM (also CWSPM).

```
Input: Input image $I$.
Output: Mid-level image representation $\mathbf{Z}$.
Using dense grid over image $I$, extract patches $\mathrm{x}_m$, described using SIFT.
for all patches descriptor $\mathrm{x}_m$ do
    Using $\mathbf{V}$ from Algorithm 3, encode $\mathrm{x}_m$ into its sparse-coded descriptor $\mathbf{u}_m$ using equation (2.2).
end for
for all Spatial window $w$ in the SPM representation do
    Find the boundaries of $w$ based on traditional SPM/OWSPM/CWSPM
    for all Dictionary entry $k$ do
        Do maximum pooling on $k^{\text{th}}$-dimension: $z_{wk} = \max_{\mathbf{x}_m \in w} u_{mk}$.
    end for
    $\mathbf{z}_w = [z_{w1}, z_{w2}, ..., z_{wK}]^T$.
end for
Construct the mid-level image representation $\mathbf{Z} = [\mathbf{z}_1^T, \mathbf{z}_2^T, ..., \mathbf{z}_W^T]^T$.
```

class membership is predicted by

$$y = \max_{c \in \mathbb{Y}} \mathbf{w}_c^T \mathbf{z} \tag{4.3}$$

A one-against-all strategy is adopted to train C binary linear SVMs, each solving the optimization problem

$$\min_{\mathbf{w}_c} \{ J(\mathbf{w}_c) = ||\mathbf{w}_c||^2 + C \sum_{i=1}^{n} l(\mathbf{w}_c; y_i^c, \mathbf{Z}_i) \} \tag{4.4}$$

in which $y_i^c = 1$ if $y_i = c$; otherwise -1. $l(\mathbf{w}_c; y_i^c, \mathbf{Z}_i)$ is a hinge loss function, which is defined as

$$l(\mathbf{w}_c; y_i^c, \mathbf{Z}_i) = [\max(0, \mathbf{w}^T \mathbf{z} \cdot y_i^c - 1)]^2 \tag{4.5}$$

which is designed to be differentiable so the training process can be done with gradient-based optimization. As with [25], we use LBFGS to train the classifier.

Table 4.1: Recognition rate of ScSPM with rectangular overlapping windows OWSPM under different ω values.

ω	15-Scene 100 train	Caltech-101 15 train	Caltech-101 30 train
0	80.28%	66.28%	72.46%
0.1	81.06%	66.74%	73.73%
0.2	81.19%	67.44%	**73.74**%
0.3	**81.54**%	**67.57**%	73.72%
0.4	81.09%	67.49%	73.60%
0.5	80.76%	66.78%	73.72%

4.3.4 Searching for ω

Having discussed how to extract features and train classifiers, the first step in our OWSPM experiment was to determine what value of ω maximizes recognition accuracy. Different settings of ω were tested on three different experiment settings: (1) 15-Scene dataset with 100 training images, (2) Caltech 101 dataset with 15 training images, and (3) Caltech 101 dataset with 30 training images. Every other parameter is set to be as similar as possible to those reported in [23, 25], to allow for direct comparison. The mean recognition rate over 10 iterations is reported in Table 4.1.

From these experiments, it is evident that the disjoint spatial windows method does not give the best accuracy. As the overlap increases, the accuracy also increases, until a certain maximum point within the $0.2 \leq \omega \leq 0.3$ region, after which it decreases. This decrease after a certain point in ω is fully expected, because the more ω increases, the less difference there will be between windows (which will become the entire image when $\omega = 1$). Based on these findings, ω is set to be 0.3 whenever the OWSPM scheme is used in this thesis.

4.4 Circular overlapping spatial windows

To further extend the proposed concept, the circular overlapping window (CWSPM) is introduced in this section. Traditionally, SPM divides each layer

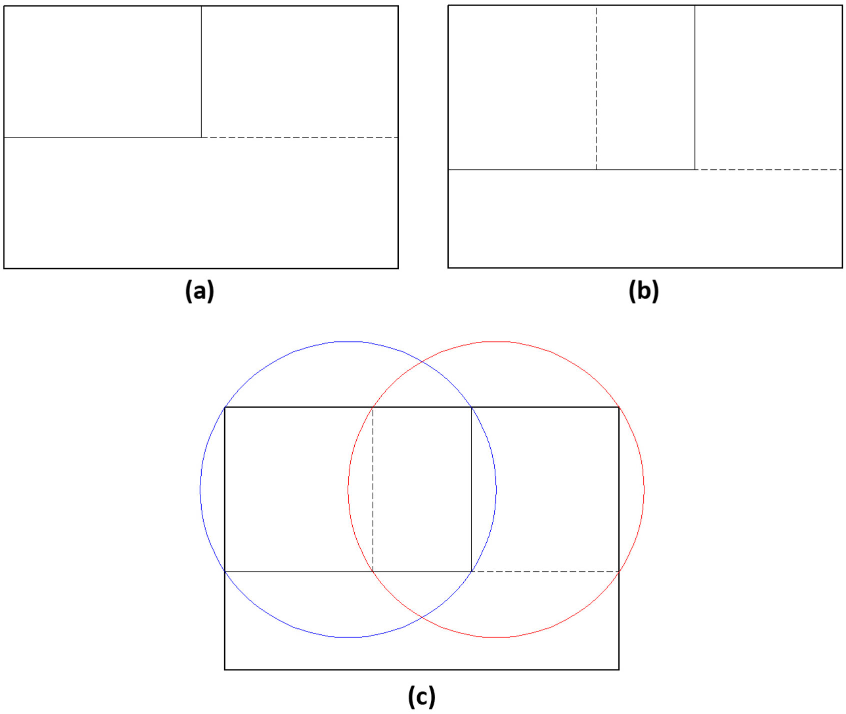


Figure 4.3: Sub-window division of (a) traditional SPM, (b) OWSPM with its overlapping rectangular windows, and (c) CWSPM with its overlapping circular windows. In this illustration, l and ω is set to 1 and 0.3 respectively and only the top two sub-windows are shown for clarity.

into rectangular sub-windows with sizes proportional to the image, and patches are then pooled based on the membership of sub-windows. This is also applicable to OWSPM sub-windows. As OWSPM aims to achieve better coverage from its sub-windows, it is intuitive that a rectangular shape will not give the optimal result.

Let us consider the rectangle's center of gravity as the focal point of a given sub-window. As such, the farthest point that can be pooled by the sub-window will be located at the four corners of the rectangle. If we wish to fully consider the context surrounding that point, a circular window would be the obvious choice. By doing so, we hope to be able to improve the descriptive power of image representations, since: (1) the contexts are fully described in all directions, and (2) the circular sub-windows inherently overlap with each other, and

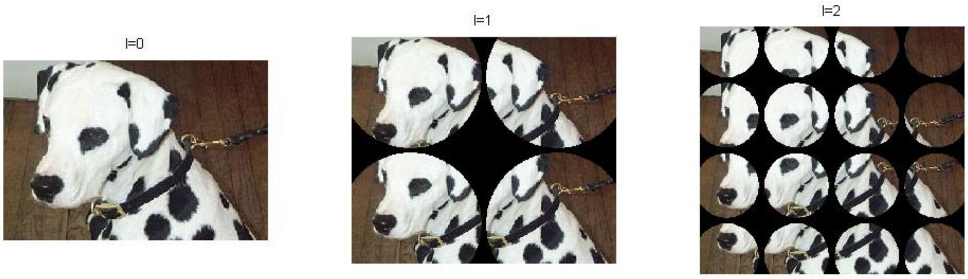


Figure 4.4: Example of CWSPM sub-window definition.

as such, CWSPM will receive the same benefits as OWSPM.

Construction of the circular windows is done by first defining the regular overlapping sub-windows of OWSPM. Then, we construct a circumcircle over each sub-window, creating circular windows with a radius of $0.5\sqrt{w_s^2 + h_s^2}$ centered at the rectangle's center of gravity. h_s and w_s are the height and width of the individual sub-windows, respectively.

4.5 Finding ω for CWSPM scheme

Similar to OWSPM, the same experiment is performed using the CWSPM scheme, and Table 4.2 reports the experimental results.

Under CWSPM, it was shown that the peak accuracy value occurs when ω is within the 0 to 0.1. It should be noted, however, that this does not mean that the spatial windows did not overlap with each other. ω controls the amount of overlap between two adjacent rectangular windows used as the basis of the circle. As such, the case of $\omega = 0$ depicts a non-overlapping situation for the base rectangles; however, the circular windows generated are already in overlap with each other.

Based on these findings, ω is set to be 0 throughout this paper whenever the CWSPM scheme is involved.

Table 4.2: Recognition rate of ScSPM with circular overlapping windows OWSPM under different ω values.

Algorithm	15-Scene 100 train	Caltech-101 15 train	Caltech-101 30 train
0	**81.62**%	67.68%	**74.14**%
0.1	81.52%	**67.84**%	73.91%
0.2	81.06%	66.70%	73.83%
0.3	81.46%	66.92%	72.63%
0.4	80.84%	66.46%	71.85%
0.5	80.06%	64.95%	71.09%

4.6 Testing of OWSPM and CWSPM scheme

Table 4.3: Recognition rate of OWSPM and CWSPM when applied to ScSPM and LLC on 15-Scene and Caltech 101.

Algorithm	15-Scene 100 train	Caltech-101 15 train	Caltech-101 30 train
ScSPM	80.28%	67.00%	73.20%
ScSPM + OW	81.54%	67.57%	73.72%
ScSPM + CW	**81.62**%	**67.68**%	**74.14**%
LLC	80.11%	64.03%	72.54%
LLC + OW	80.27%	65.89%	72.34%
LLC + CW	80.58%	66.33%	73.06%

Using the obtained ω values, the performance of ScSPM under six different schemes are compared: (1) ScSPM, (2) OW-ScSPM, (3) CW-ScSPM, (4) LLC, (5) OW-LLC, and (6) CW-LLC. Note that we use OW and CW to denote OWSPM and CWSPM being used to replace traditional SPM in ScSPM/LLC. Table 4.3 and Table 4.4 show the result for the 15-Scene, Caltech 101, and Caltech 256 datasets. The results obtained in this work may differ from [23] due to two reasons: (1) the original LLC method in [23] utilized HOG features with 3 different scales to describe patches (instead of SIFT used by ScSPM) and (2) unavoidable differences in some parametric settings. SIFT is chosen for both ScSPM and LLC in this work, for purposes of direct comparison.

ScSPM and LLC: From the results gathered in the experiments, the usage of OWSPM and CWSPM outperforms the traditional SPM for both ScSPM and LLC in term of recognition rate. It is possible to achieve up to 3.68%

Table 4.4: Recognition rate of OWSPM and CWSPM when applied to ScSPM and LLC on 15-Scene and Caltech 101.

ω	Caltech-256	Caltech-256	Caltech-256	Caltech-256
	15 train	30 train	45 train	45 train
ScSPM	27.73%	34.02%	37.46%	40.14%
ScSPM + OW	31.31%	36.57%	39.15%	41.27%
ScSPM + CW	**31.41**%	**36.59**%	**39.32**%	**41.50**%
LLC	26.17%	31.78%	34.52%	36.64%
LLC + OW	27.09%	32.35%	34.68%	37.21%
LLC + CW	27.91%	32.85%	35.72%	37.30%

improvement in recognition accuracy, which is quite a significant increase considering the simplicity of this concept. Additionally, the results are obtained at a similar computational cost. Another observation is that CWSPM constantly outperforms the OWSPM across all experiments, confirming that the disjoint rectangular sub-windows omit important information and context.

The proposed method is tested with more challenging datasets such as STL-10, MIT-Indoor, and UIUC-Event. Furthermore, with these datasets, different classification algorithms are used to test the feasibility of this overlapping paradigm on other methods that are different from ScSPM and LLC. Object Bank [16] is used for MIT-Indoor and UIUC-Event, while the deep network from [55] (Simulated Fixation) is used for STL-10.

Object Bank [16] is a high-level image representation, using a scale-invariant response map of a large number of pre-trained generic object detectors that are blind to the testing dataset or visual task. For each object detector, its map response will fire one of the spatial windows designed using the SPM paradigm, and the final image representation is found by concatenating the map responses from each detector, as depicted in Figure 4.5.

As such, for each object detector, we will have a single SPM model. The proposed OWSPM and CWSPM paradigms are used to modify the architecture of SPM in the map response of Object Bank, noting that overlapping regions will provide a map response that is less precise due to the non-disjoint nature of OWSPM and CWSPM, but being more accurate since there will be less confu-

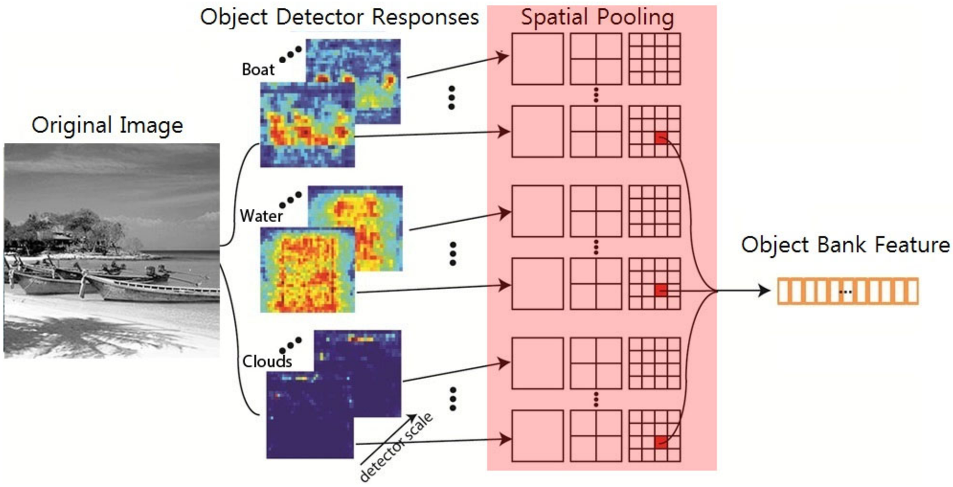


Figure 4.5: Overview of object bank and how it utilizes SPM's spatial pooling. The step highlighted in red is replaced with OWSPM and CWSPM scheme with $\omega = 0.3$ and 0 , respectively.

sion when the object lies in the middle of two windows.

Each object detector will undergo its own OWSPM and CWSPM pooling. As for overlapping parameter, we found that $\omega = 0.3$ for OWSPM and $\omega = 0$ for CWSPM still produce the best results. The Object Bank pipeline is highly similar to conventional BoW with SPM models, thus the benefits of overlapping spatial windows are also transferable to Object Bank. Details of comparison between Object Bank with overlapping windows and without can be seen in Table 4.5.

Deep Convolutional Neural Network (Deep CNN) with Simulated Fixation [55] is a deep learning paradigm to learn image representations. The framework was originally designed and learned in video sequences, by simulating fixation on salient objects, giving meaning to the sequence. However, the image representation itself is spatial in nature, instead of being spatial-temporal. As such, it is possible to simulate the same fixation on salient objects within a still image.

Unlike Object Bank, the Deep CNN framework is very different compared to ScSPM framework. However, throughout the layers of CNN, there exists a pooling step, which surprisingly utilizes the same ideas as SPM. In fact, a quick look to this pooling step will make us able to identify the step as SPM model,

Table 4.5: Average recognition success rate for other image database under different baseline.

Baseline	Dataset	Standard	OW	CW
Object Bank [11]	UIUC-Event	76.30%	78.56%	**78.83%**
	MIT-Indoor	37.60%	38.41%	**39.33%**
	15-Scene	80.90%	82.60%	**83.00%**
Simulated Fixations [55]	STL-10	61.00%	62.11%	**62.91%**
	Caltech-101	74.60%	75.90%	**76.45%**

although most works in CNN does not explicitly called it spatial pyramid. CNN utilizes SPM windows to pool the response from convolutional layers. The proposed OWSPM and CWSPM is used to replace the disjoint SPM arrangement in the pooling stage within this deep architecture.

Their performances are listed in detail in Table 4.5. These results further confirm that the usefulness of overlapping windows is not only confined to simple datasets. Both Object Bank and Simulated Fixations are used because they make use of the non-BoW approach, but still utilize the Spatial Pyramid. Furthermore, the two methods are chosen because we want to show that OWSPM and CWSPM can be applied to both methods that is similar to BoW (Object Bank) or completely different to BoW approach (CNNs).

4.7 Qualitative results

In terms of overall accuracy, we can see that OWSPM and CWSPM are an upgrade from the traditional SPM. In this section, however, we will evaluate the two proposed methods more deeply by looking into the results. Figure 4.6 shows the false negatives obtained from Caltech 101 with 30 training images from class "Leopard". We choose this class as an example since it exists in the Caltech 4, Caltech 101, and Caltech 256 datasets. Furthermore, the class "Leopard" has some similar classes in the dataset, such as "Cougar_body", "Cougar_face", and "Wild Cat". These classes are of a different category but have some degree of similarity, enough for them to get easily mistaken for each other.

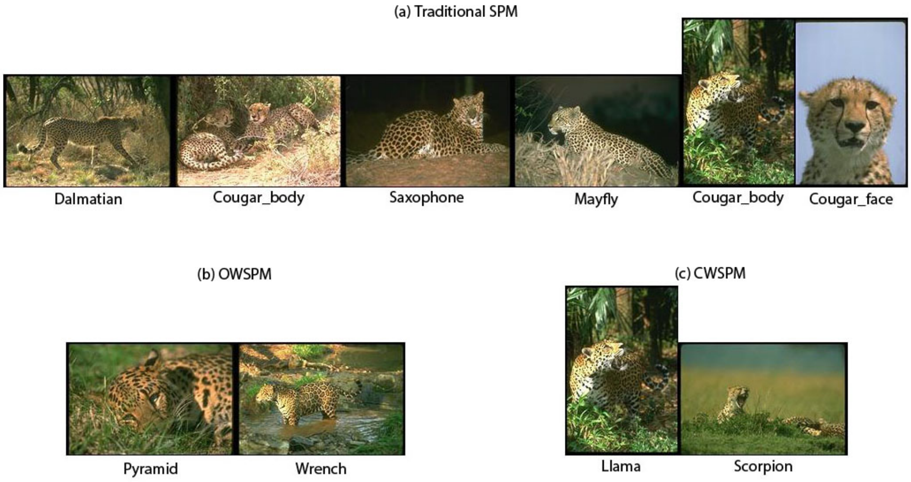


Figure 4.6: List of all false negatives from class "Leopard" in Caltech 101 under Traditional SPM, OWSPM, and CWSPM. The labels below the images indicate what classes they were falsely classified as.

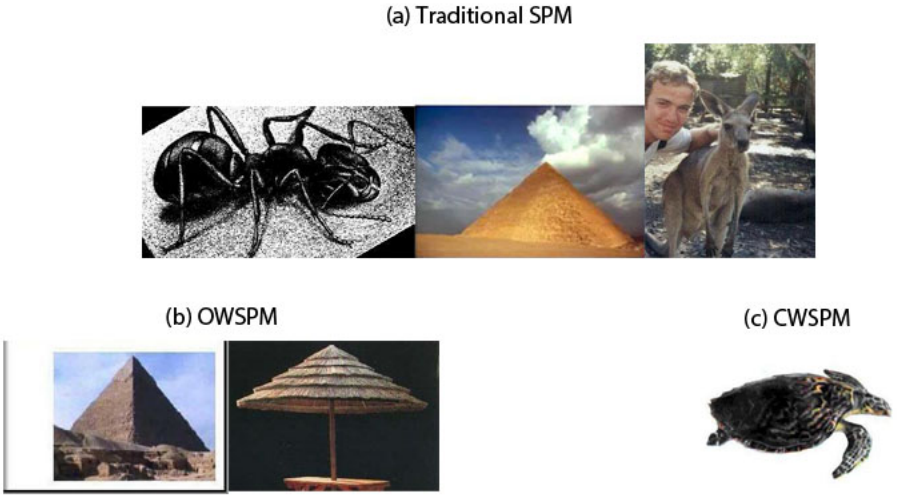


Figure 4.7: List of all false positives from class "Leopard" in Caltech 101 under Traditional SPM, OWSPM, and CWSPM.

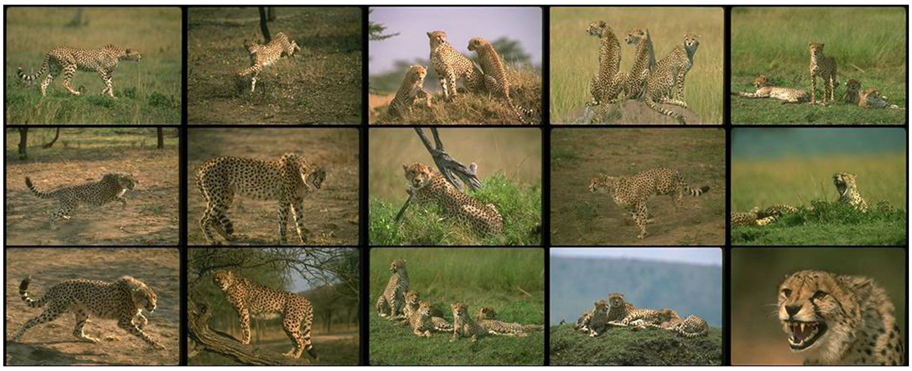
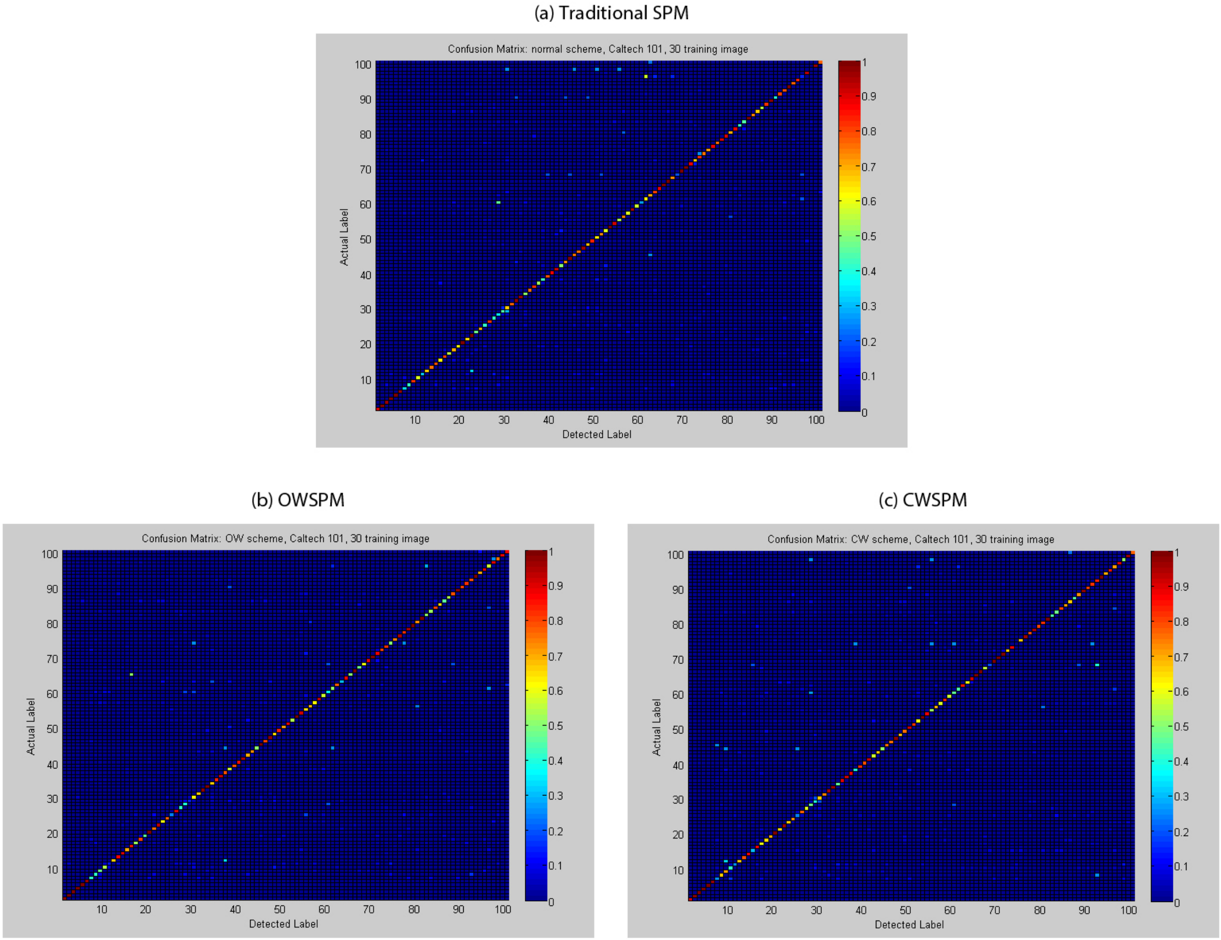


Figure 4.8: True positives from class "Leopard" in Caltech 101.



(a) Traditional SPM

(b) OWSPM

(c) CWSPM

Figure 4.9: Comparison of confusion matrices between Traditional SPM, OWSPM, and CWSPM. Detailed label names are not displayed due to space limitation; instead, the label number is displayed. Heat maps are shown next to each matrix.

As expected, for *Traditional SPM*, we have more false negatives (a total of 6 images from the class "Leopard"). Of these false negatives, some are classified as classes that are particularly close to the class "Leopard". "Cougar_body" and "Cougar_face" are obvious examples, while the class "Dalmatian" can be attributed to the form and texture of the leopard being considered. However, traditional SPM also returned misclassification into classes that do not have any semblance at all, like "Saxophone" and "Mayfly". In contrast, the overlapping scheme performed much better, with each method registering 2 false negatives. *OWSPM* performed less desirably, registering false negatives from two classes that are far from "Leopards", while *CWSPM* performed better, returning one close class in "Llama" and one seemingly odd misclassification in "Scorpion". Qualitatively, we see improvements coming from a broader context being integrated into the image representation.

Figure 4.7 shows the false positives from the same class. Here we see that traditional SPM was again the worst performer of the 3 methods, registering 3 images, followed by 2 from OWSPM and 1 from CWSPM. The classes falsely assigned to "Leopard", however, are a bit confusing as the assignment was seemingly random, both for traditional SPM and for overlapping schemes. Interestingly, we can observe that the class "Pyramid" appears again after being in the false negative list, which indicates that SVM learns that the image descriptors of "Pyramid" and "Leopard" are somewhat close.

Finally, Figure 4.8 shows examples of true positives. However, as we have already listed all the false negatives from the recognition process, if we split them into three categories the same list will be displayed three times, with only minor differences in terms of the false negatives. Therefore, we only show them as one list, noting that the three methods managed to classify the majority of the images in class "Leopard", aside from those listed in the false negative list.

Figure 4.9 shows the confusion matrices (in the form of heat maps as each matrix is a 101×101-sized matrix) derived from each method. What we can observe from these confusion matrices is that traditional SPM has a tendency to have more distributed values throughout the matrices. Overlapping schemes produce confusion matrices that are more sparse, but with stronger points in the heat map. This indicates that the false negatives are grouped together in fewer classes.

4.8 The $l = 2$ layer

Let us define pyramid configuration $\mathbf{P}$ to denote the configuration of the pyramid layers used in a specific experiment. The setting $\mathbf{P}= \{0, 1, 2\}$ refers to the traditional SPM arrangement under the condition $L= 3$. When subscripts o and c appear under a number, it means that the layer was set up using the OWSPM or CWSPM respectively (i.e., 2_c refers to the set of windows at $l = 2$ arranged under the CWSPM scheme).

An important discovery can be made by inspecting the performance of each layer for the traditional SPM, OWSPM and CWSPM. Table 4.6 shows the recognition rate of Caltech 101 with 30 training images under selected pyramid configurations (all results were obtained using ScSPM or its OWSPM/CWSPM variant). The results from the complete pyramids ($\mathbf{P}_{nw}$, $\mathbf{P}_{ow}$, $\mathbf{P}_{cw}$) compared to results coming from only the $l = 2$ layer are of a particular interest. It is interesting to see that the overlapping schemes shrunk the distance between full pyramid and $l = 2$ layer considerably, compared to traditional SPM arrangement ($\mathbf{P}_{ow} - \mathbf{P}_{2o} = 0.53\%$ for OWSPM and $\mathbf{P}_{cw} - \mathbf{P}_{2c} = 0.66\%$ for CWSPM, compared $\mathbf{P}_{nw} - \mathbf{P}_2 = 2.33\%$ for traditional SPM).

These results suggest that when memory allocation is limited, we can bypass the first two layers of the pyramid and use the $l = 2$ layer directly for image representation. That is, instead of using 21 sub-windows (when $L = 3$), we need only use 16 sub-windows. This cuts the memory consumption by 24% while achieving similar results to the method utilizing the complete pyramid.

Table 4.6: Average recognition rate of Caltech 101 database with 30 training images for various spatial pyramid configurations.

Pyramid Configurations	Recognition Rate
$\mathbf{P}_2 = \{2\}$	70.13%
$\mathbf{P}_{2o} = \{2_o\}$	73.19%
$\mathbf{P}_{2c} = \{2_c\}$	73.48%
$\mathbf{P}_{nw} = \{0, 1, 2\}$	72.46%
$\mathbf{P}_{ow} = \{0, 1_o, 2_o\}$	73.72%
$\mathbf{P}_{cw} = \{0, 1_c, 2_c\}$	74.14%

Table 4.7: Average recognition rate of Caltech 101 database, Caltech 256 database, and 15 scene database using various pyramid configurations.

Database	$\mathbf{P}_{nw}$	$\mathbf{P}_{2o}$	$\mathbf{P}_{ow}$
15-Scene (100 train)	80.28%	80.30%	81.54%
Caltech 101 (15 train)	66.28%	66.83%	67.57%
Caltech 101 (30 train)	72.46%	73.19%	73.72%
Caltech 256 (15 train)	27.73%	29.79%	31.31%
Caltech 256 (30 train)	34.02%	34.96%	36.57%
Caltech 256 (45 train)	37.46%	37.60%	39.15%
Caltech 256 (60 train)	40.14%	40.36%	41.27%

Database	$\mathbf{P}_{nw}$	$\mathbf{P}_{2c}$	$\mathbf{P}_{cw}$
15-Scene (100 train)	80.28%	80.48%	81.62%
Caltech 101 (15 train)	66.28%	67.28%	67.68%
Caltech 101 (30 train)	72.46%	73.48%	74.14%
Caltech 256 (15 train)	27.73%	30.83%	31.41%
Caltech 256 (30 train)	34.02%	35.19%	36.59%
Caltech 256 (45 train)	37.46%	38.86%	39.32%
Caltech 256 (60 train)	40.14%	40.38%	41.50%

This can be very useful when the database is very large, as in the case of Caltech 256, with 60 training images (in such a case, we would need at least 2.5 GB of memory just to store the training data).

Furthermore, these results are still consistent when tested on different datasets and training numbers, as shown in Table 4.7. While it is clear that the complete pyramid gives the best results, the experiment shows that image representation using the $l = 2$ layer does not fall too far behind, and can be used as a reasonable compromise when memory cost is critical. It should be noted that the results shown here are solely based on ScSPM and its OWSPM/CWSPM variants.

4.9 Concluding remarks

Two extensions of the traditional SPM have been proposed and tested, using the concept of overlapping spatial windows. The first proposal extends the rectangular sub-windows to overlap with each other without increasing memory complexity, while the second proposal strives to further increase the discriminability by using circular windows. Experiments show that the recognition rate for both rectangular (OWSPM) and circular (CWSPM) overlapping windows outperforms the traditional SPM across the two different frameworks of ScSPM and LLC.

Furthermore, it has been shown that the OW and CW variants allow us to bypass the lower layers of traditional SPM, cutting memory complexity by 24%. It is important that both OWSPM and CWSPM improve the recognition rate of ScSPM and LLC, as both have been used as the building blocks of current state-of-art object recognition systems, and thus, we can expect both of them to contribute to other frameworks as well.

Chapter 5

Interleaved Spatial Window

5.1 Introduction

As discussed in the last chapter, the l^{th} layer of the SPM pyramid divides the image equally into $2^l \times 2^l$ sub-windows. Experiments with the OWSPM and CWSPM schemes led to the discovery that the bulk of the information is located in the bottom layer of the pyramid. That is, when L= 3, most of the information is stored in the layer where $l = 2$. Knowing this, it is possible to bypass all other layers during training without losing much in the recognition rate, effectively reducing the size of the mid-level feature by 24%. Furthermore, it was found that the performance of this single layer was superior to the traditional SPM, even with all its layers included.

These findings from the topic of overlapping spatial windows lead us to some questions. Firstly, with the introduction of overlapping spatial windows, is there a way to make the SPM model more efficient in terms of memory cost, without sacrificing performance? Secondly, assuming that this is possible; can we utilize the saved cost in memory by introducing more complexity into the dictionary used by the overlapping window model, with the aim of improving the recognition rate? To answer these two questions, this chapter proposes to delve deeper OWSPM and CWSPM schemes.

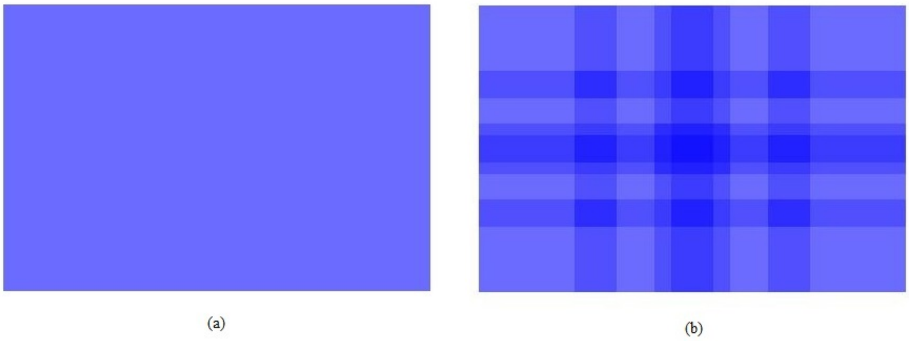


Figure 5.1: The repeated usage of a single patch in (a) traditional SPM and (b) OWSPM with ω= 0.3. Differences in color denote how many times a patch at a particular location is used to form the mid-level image representation.

With overlapping windows, each spatial window covers a larger area of the image for pooling. A patch can be used multiple times in the process of creating mid-level features, adding redundancy within the process. In fact, in the traditional SPM model, a patch is used exactly L times. Motivated by this fact and the increasing coverage brought by overlapping windows, this work proposes to extend the concept further, by designing an interleaved pooling scheme that will reduce the cost of SPM representation, while maintaining a similar level of performance. This concept will be referred to as the Interleaved window SPM (IWSPM) throughout this thesis.

5.2 Interleaved window

Assuming low-level features are extracted from the dense grid, let $\mathbf{X}$ and $\mathbf{V}$ be the collection of N low-level descriptors, and the codeword dictionary with K cluster centers, respectively, as described in the previous section. As mentioned in Section 2.2, each low-level feature $\mathbf{x}_i$ is coded as $\mathbf{u}_i$ with $i \in 1, 2, 3, \ldots, N$ using a pre-trained codebook $\mathbf{V}$. If the mid-level feature **z** is collected from each spatial window by max-pooling, a total of $\sum_{l=0}^{L} 4^l$ vectors will be collected and concatenated as an image representation, with a dimension of $K \sum_{l=0}^{L} 4^l$.

If we use the overlapping scheme instead of the traditional scheme, some

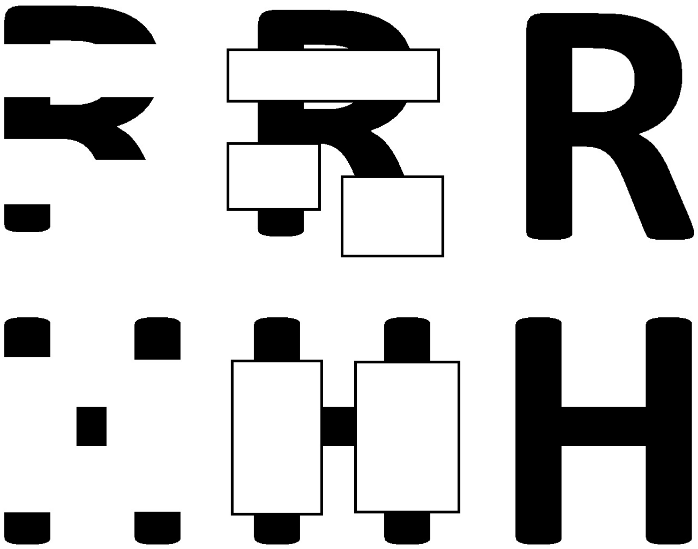


Figure 5.2: The case of missing information. Left: two letters "R" and "H" with severe case of missing information, middle: having prior knowledge that the image is occluded at particular locations lead to easier recognition, and right: the original "R" and "H" letters.

portions of the image will be covered by the spatial windows more often than the others. Figure 5.1 illustrates this behavior by putting different shades of blue in the frame of an image. The darker the shade, the more sub-windows cover a specific portion of an image. Observing the shading detail in Figure 5.1, it is possible to come up with several observations.

Firstly, a patch will be used multiple times in the formation of the image representation. In the case of traditional SPM, each patch will be used exactly L times. In the overlapping scheme, a patch can be used more than L times in the process. This shows us that there is a degree of redundancy in the pooling process of the overlapping window scheme. While at times these redundancies may be beneficial, one might wonder if it is possible to reduce them without lowering the overall recognition performance.

Secondly, to consider the problem of missing information, in recognizing the letters shown in left-most column of Figure 5.2, the letters "R" and "H" have large portions of the image removed in the first column, making it very difficult for human eyes to correctly identify both letters. In middle column,

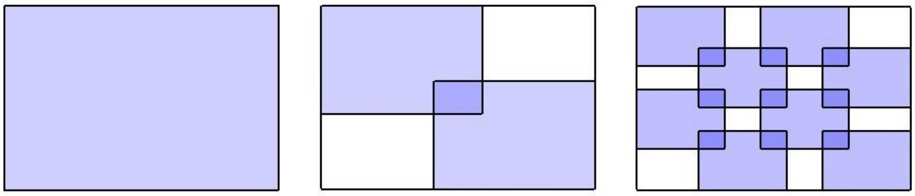


Figure 5.3: Spatial window arrangement of IWSPM with ω set as 0.3. Left to right: the arrangement at $l = 0$, $l = 1$, and $l = 2$, respectively. Colored boxes are included in the image representation.

the boxes with deleted information are visible. In this case, since we know the parts where the image is incomplete, we are able to easily identify the two letters. This demonstrates that prior knowledge of the location where missing information occurs can make the task of recognizing incomplete objects much easier.

Thirdly, with overlapping windows, the larger size of each spatial window means that the context within an image of interest will be covered more extensively. Removing a spatial window from OWSPM will be less costly to the recognition result, compared to removing a spatial window from the traditional SPM.

Using these observations, this thesis proposes to simplify the pooling scheme by using a checkerboard-like layer as shown in Figure 5.3. That is, if a particular spatial window is used, then the window adjacent to it will be removed from the layer. By doing so, each layer l will have half its content removed (except for l= 0).

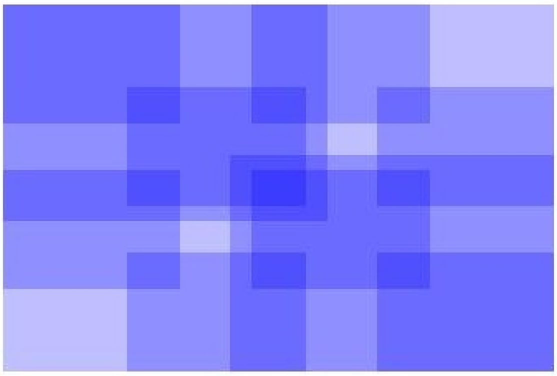


Figure 5.4: Coverage of IWSPM, the darker the color in a particular location, the more frequent a patch will be used for constructing mid-level image representation.

5.3 Experiments

During the experiments, the proposed concept was implemented and compared with ScSPM and OWSPM, which served as benchmarks. The three-layered SPM ($L = 3$) was used in this particular approach, and we tested the ScSPM, OWSPM, and IWSPM using three databases: Caltech 101, Caltech 256, and 15 Scene.

As presented in Figure 5.3, we included the first sub-window of each spatial layer in the representation and discarded any sub-window adjacent to it. We then proceeded to include the next sub-window that was not adjacent to any window already in the representation, and repeated the process (without loss of generality we can pick the top-left sub-window as the first, but there is no restriction to using the complement arrangement). This process was done for all layers $l = 0, 1, 2$, however, it should be noted that there is no difference between traditional SPM and the IWSPM scheme t layer $l = 0$.

As in Chapter 4, the settings in [25] were followed to allow for direct comparisons. The size of the codebook $\mathbf{V}$ was set as $K = 1024$ for all three databases. As the number of sub-windows involved in the representation was cut down by almost half, the final length of the image representation was also cut down by the same amount. Thus, increasing the size of $\mathbf{V}$ became viable (although it also resulted in the increase of both memory and computational cost). The steps for training dictionary, extracting descriptors, and training classifier were the same as those discussed in Sections 4.3.1 to 4.3.3, with the exception of the pooling step, in which the IWSPM arrangement was used.

ω was set as 0.3 for the overlapping rectangular window, in accordance with the result from Chapter 4. All three layers were included in the experiments. Similarly, each experiment was repeated 10 times, using randomly selected training and testing sets. The experiment for IWSPM itself did not differ much from those for OWSPM and CWSPM. The only difference here was that not all sub-windows were included in the image representation; a selection process was performed on them first. The experiment was designed as an exploratory experiment, to observe how the result would change with a simple modification of SPM.

IWSPM was executed using two test cases:

1. Testing of IWSPM: In this test scenario, the performance of interleaved spatial windows in recognizing databases was evaluated.
2. Completing the dictionary: In this test scenario, a broader codewords dictionary was evaluated by increasing the number of K, to capitalize on the memory saved by IWSPM.

Any unmentioned parameter settings follow similar settings to those in Chapter 4, to allow for direct comparison. Both sets of possible arrangement were tested (i.e. the set from Figure 5.3 and its complement), and we used the arrangement with better results for comparison. This was done because the selection of the top-left sub-window as the first is rather intuitive, and the complement arrangement is, logically also viable for experimentation.

The purpose of these experiments was mainly to further investigate the question “Is the current SPM arrangement optimal?” Based on the results from the previous chapter, it was found that SPM is likely to be sub-optimal. The fact that most information is located in the lowest layer of the pyramid, with the higher level only contributing to a minimum increase in recognition accuracy, is very surprising. By setting the SPM arrangement in an interleaved manner as designed here, we can further examine whether all sub-windows in one layer of the pyramid are really necessary. Considering the increasing coverage provided by overlapping window schemes, it seems likely that not all the sub-windows are necessary.

5.4 Results

5.4.1 Results on 15 Scene database

100 images from each class were randomly selected for the purpose of training the classifier, and it was tested using the remaining images. Table 5.1 shows the recognition rate for ScSPM, OWSPM and IWSPM. In this database, we can see a reduction in performance from OWSPM to IWSPM of 1.06%, which is expected since we are using fewer spatial windows. However, the performance of IWSPM was still higher (by 0.2%) as compared to ScSPM. This result shows us that some of the information in a complete set of spatial windows might be redundant for the task of scene classification. As information is spread throughout the image in scene classification, removing a number of spatial windows will not bring about a severe reduction in terms of performance.

5.4.2 Results on Caltech 101 database

While information in scene recognition is spread throughout the images, the same advantage is not experienced in the case of object recognition. In most cases, information about the object of interest will be localized in the vicinity of the object. However, it is also worth noting that most current state-of-the-art object recognition techniques do not detect the location of the image by using keyword detectors, but by sampling patches over a dense grid. This is due to the fact that the contextual information spread outside of the object's location

Algorithms	15-Scene 100 train
ScSPM	$80.28 \pm 0.93\%$
OWSPM	$81.54 \pm 0.46\%$
IWSPM	$80.48 \pm 0.32\%$

Table 5.1: Recognition rate of ScSPM, OWSPM and IWSPM on 15-Scene.

Algorithms	Caltech 101 15 train	Caltech 101 30 train
ScSPM	$67.00 \pm 0.45\%$	$73.20 \pm 0.54\%$
OWSPM	$67.57 \pm 0.41\%$	$73.72 \pm 0.83\%$
IWSPM	$66.28 \pm 0.47\%$	$73.22 \pm 0.98\%$

Table 5.2: Recognition rate of ScSPM, OWSPM and IWSPM on Caltech 101.

proves to be very beneficial for recognition purposes.

Caltech 101 is a database containing a total of 9146 images over 101 classes of objects (living and still). As the number of images within a single class can vary from 31 to 800 images, the maximum number of training images is restricted to 30. ScSPM, OW-SPM and IW-SPM are used to test the proposal using 15 and 30 training images. The results are shown in Table 5.2.

In this dataset, it was found that the performance of IWSPM drops to below that of ScSPM when 15 training images were used, recording a difference of 0.72% and 1.29% as compared to ScSPM and OWSPM respectively. This difference in performance became less severe when more training images were used, scoring slightly better results than ScSPM (73.22% compared to 73.20%) and closing the gap with OWSPM to a mere 0.5%. The reduction of the recognition rate is expected, as less data is involved in IWSPM. The reason ScSPM performs better than IWSPM under 15 training images is because in Caltech 101, the objects are localized with low intra-class variance in term of appearance and location, hence the omission of spatial windows will remove discriminative data. The gap, however, is shortened when more training data was used. In such cases, IWSPM was able to match ScSPM.

Algorithms	15 train	30 train	45 train	60 train
ScSPM	$27.73 \pm 0.51\%$	$34.02 \pm 0.35\%$	$37.46 \pm 0.55\%$	$40.14 \pm 0.91\%$
OWSPM	$31.31 \pm 0.22\%$	$36.57 \pm 0.24\%$	$39.15 \pm 0.33\%$	$41.27 \pm 0.61\%$
IWSPM	$30.02 \pm 0.27\%$	$35.10 \pm 0.23\%$	$38.48 \pm 0.40\%$	$40.77 \pm 0.70\%$

Table 5.3: Recognition rate of ScSPM, OWSPM and IWSPM on Caltech 256.

5.4.3 Results on Caltech 256 database

Caltech 256 offers an extension to Caltech 101. As the name implies, it contains 256 object classes with over 30607 images; hence it is considered a challenging dataset, not only due to its sheer size, but also because of the high intra-class variability in terms of shape and location. In contrast to Caltech 101, each class contains at least 80 images, thus it is possible to train the classifier with a higher number of training images as compared to Caltech 101. The settings of 15, 30, 45 and 60 training images are used to test IWSPM, and the results are shown in Table 5.3.

Here we can observe that IWSPM consistently performs in the region between ScSPM and OWSPM. Again, IWSPM is expected to give a worse performance than OWSPM, as it is after all, the simplified version of OWSPM. However, the average drop in performance is small (1.2%). In contrast to ScSPM, IWSPM performs surprisingly well, recording some significant improvements over various numbers of training images. The level of intra-class variability plays an important part in making IWSPM perform well, as missing information became less costly as variability increased.

5.4.4 Comparing IWSPM to OWSPM at $l = 2$

The results consistently show that IWSPM outperforms ScSPM, with an exception of Caltech 101 with 15 training images. In addition to this, OWSPM does not fall behind OWSPM much. Generally, it was found that IWSPM offers a recognition rate somewhere in between ScSPM and OWSPM. Moreover, all this was achieved at half the memory cost of ScSPM and OWSPM each. As

Database	$\mathbf{P}_{nw}$	$\mathbf{P}_{2o}$	$\mathbf{P}_{iw}$
15-Scene (100 train)	80.28%	80.30%	80.48%
Caltech 101 (15 train)	67.00%	66.83%	66.28%
Caltech 101 (30 train)	72.46%	73.19%	73.22%
Caltech 256 (15 train)	27.73%	29.79%	30.02%
Caltech 256 (30 train)	34.02%	34.96%	35.10%
Caltech 256 (45 train)	37.46%	37.60%	38.48%
Caltech 256 (60 train)	40.14%	40.36%	40.77%

Table 5.4: Comparison of recognition performance between ScSPM, OWSPM with only $l = 2$ layer, and IWSPM.

described in the last chapter, using only the $l = 2$ layer in OWSPM led to an improvement over ScSPM, even though it utilized 24% lesser memory. Since IWSPM and OWSPM at layer $l = 2$ performs generally at the same level, it will be interesting to compare the performance between them both. A detailed comparison can be found in Table 5.4.

Clearly, we can see that IWSPM offers better performance than $l2$-OWSPM, even with less memory cost compared to the $l2$-OWSPM (except, again, at Caltech 101 with 15 training images). Our results shows that IWSPM can be considered as an effective way of obtaining mid-level image representation for scene and object recognition, offering efficiency in terms of memory consumption and performance.

Clearly, IWSPM offers better performance than OWSPM at $l = 2$, even at a lesser memory cost compared to this $l2$-OWSPM (except, again, in the case of Caltech 101 with 15 training images). These results show that IWSPM can be considered as an effective way of obtaining mid-level image representations for scene and object recognition, offering efficiency in terms of memory consumption and performance.

5.4.5 Discussions on the effectiveness of traditional SPM arrangement

To summarize the findings from the experiment, it was found that:

1. IWSPM performs at a similar level to traditional SPM (it is higher than except for Caltech 101 with 15 training images),
2. Complete OWSPM achieves the best results, but with 100% more cost compared with IWSPM, and a return of around 1% of performance,
3. Intelligent selection of sub-windows is essential to achieve efficient (in terms of storage cost and performance) image descriptor, and
4. IWSPM scheme, which sits at 50% memory cost of the traditional SPM, performs very close to the $\mathbf{P}_{2o}$ scheme, which sits at 75% memory cost of the traditional SPM.

From these four findings, it is possible to conclude that the traditional SPM is sub-optimal when compared to the IWSPM scheme (in terms of memory cost) and to the OWSPM scheme (in terms of performance). In addition, it was found that the selection of windows in SPM was pertinent, as a good selection of windows can lead to better performance at a lower memory cost, saving both time and space within the system.

What is most surprising, however, is that from findings (2) and (4) it is possible to see that IWSPM, $\mathbf{P}_{2o}$, and the full pyramid did not differ much with respect to the increase in memory cost. The results of this experiment indicate that there is a point of saturation, lying in the region between half and full memory, in SPM. Furthermore, if increasing the number of spatial windows led to increasing recognition accuracy, it is possible that a local or global maximum may exist between these two points in the spatial window selection. Determining these points would therefore lead us to the actual optimization for the SPM model.

Finding these points, however, will prove a difficult task, as they are specific to each dataset. Finding local maxima will be considerably easier as compared to proving that a local maxima is actually the global maximum. This thesis shall present two such selection processes in the next chapter to see if the conjecture made in this chapter is indeed true.

5.5 Concluding remarks

This chapter proposed a novel mid-level image representation that is cost-effective, with respect to memory consumption and recognition performance. The new representation takes its inspiration from overlapping sub-windows, noting that omitting a sub-window will be less detrimental here as compared to when it is done on traditional SPM. The Interleaved Window scheme extends the normal SPM arrangement by omitting adjacent sub-windows, resulting in a checkerboard-like arrangement, and using overlapping sub-windows to reduce the negative effect of the windows omitted.

Experiments run on 15 Scene, Caltech 101 and Caltech 256 datasets showed that IWSPM performs well, with recognition rates in between ScSPM and OWSPM, but at half the memory cost. In turn, the memory saved can be used to accommodate more complex models.

Chapter 6

Optimal Window Arrangement

6.1 Introduction

SPM has proved to be very simple and effective, which led to it being used in most systems that utilize the BoW model. Under this spatial modelling, researchers have been able to produce state-of-the-art results. Additionally, the main idea of SPM implementation has gone beyond the BoW approach. One example is the spatial pooling process used in unsupervised feature learning, such as Deep Learning. It is evident that the computer vision community has accepted SPM as a basic approach towards the construction of image representations.

The aim of this chapter is to challenge the traditional SPM model by showing the sub-optimality of its current configuration. A simple example of sub-optimality can be found when we consider the representation cost brought on by the SPM model. With a codeword dictionary of size K, the traditional BoW model would represent an image as a K-dimensional vector. Conversely, when we apply the SPM model, an image would be represented using $K \sum_{l=0}^{L} 4^l$ dimensions. Most models utilize $L= 3$, which leads to a representation 21 times larger than BoW. As the number of training images increases with the introduction of more complex datasets, this limitation will become problematic.

Krapac et al. have shown in [28] that using the Fisher Vector (FV) as an appearance descriptor not only allows a smaller codeword dictionary, but also when combined with spatial pyramids, it is able to achieve state-of-the-art performances with only two layers of the pyramid. This is particularly interesting, as it indirectly asserts that the current SPM architecture is sub-optimal.

In previous chapters, the traditional disjoint arrangement of spatial windows was found to be ineffective, hence, image representations would benefit from a broader context offered by overlapping spatial windows. It was found that the introduction of this new concept of spatial windows led to a significant increase in recognition accuracy. This chapter will affirm the conclusion from the previous chapter that the traditional SPM is sub-optimal, by showing that in a 3-layer overlapping spatial window of SPM ($L = 3$), the contributions of the 0^{th} and 1^{st} layers are very small compared to the 2^{nd} layer. In fact, representations of an image (using the overlapping spatial window with only the 2^{nd} layer) consistently outperform the traditional SPM. These findings show that the arrangement of spatial windows in SPM is highly redundant, and can be further optimized.

Taking the aforementioned aspects into consideration, this chapter will present in detail how SPM can be improved. The OWSPM and CWSPM scheme will be utilized in the SPM to show the benefit of overlapping spatial windows. To extend this further, this chapter proposes a novel method to learn the optimal spatial arrangement that is superior to the traditional SPM in terms of both memory consumption and performance. This scheme shall be referred as Optimal Arrangement Spatial Pyramid Matching (OA-SPM). This chapter will also introduce a cheap and fast way to approximate the optimal arrangement of the spatial pyramid in this chapter.

As such, the contributions in chapter can be broken down into several points:

- Investigating the extent of sub-optimality in the traditional SPM model.
- Introducing overlapping spatial windows as a possible optimization module.

- Finding the optimal spatial window arrangement through the combination of overlapping spatial windows and learning their arrangement using the OA-SPM.

6.2 The sub-optimality of traditional SPM

OWSPM and CWSPM both exhibit consistent improvement compared to the traditional SPM when applied to both ScSPM and LLC (up to 3.7% in the average recognition rate). These improvements confirm the hypothesis that the current disjoint spatial window arrangement is sub-optimal. The fact that these improvements are reproducible when applied to different recognition frameworks tell us that the benefits of overlapping windows are not limited to a particular framework.

In addition to that, the 2^{nd} layers of both the OWSPM and CWSPM schemes outperform traditional SPM (that has all three layers). As such, it is possible to save nearly 24% of memory cost in the training step, by using overlapping spatial windows and discarding the 0^{th} and 1^{st} layers of OWSPM or CWSPM, and yet still outperforming traditional SPM.

Interestingly, even with incomplete information, our vision system is able to evaluate and infer a meaningful conclusion from what we see. Furthermore, when the identity of some missing information (what information is missing, where is the missing data located, etc.) is known, the deduction becomes more efficient. Under the IWSPM model, it is possible to reduce the memory cost of image representation by approximately 50%, as the total number of windows involved decrease from $\sum_{l=0}^{L-1} 4^l$ to $\sum_{l=0}^{L-1} 2^l$. While IWSPM performs less accurately than in the OWSPM and CWSPM scheme, it proves to be more efficient than the traditional SPM model. In fact, Caltech 101, under the 15 training images condition, is the only exception where the IWSPM scheme performs worse than ScSPM (by 0.22%).

It is interesting to note that the decrease in performance between OWSPM and IWSPM schemes is very small (around 1% in difference). This means that 50% of the data in the complete SPM set contributes to only 1% of additional discriminability. The most interesting result can be seen when we compare the result of IWSPM with the $l = 2$ layer of OWSPM. Note that the $l = 2$ layer comprises approximately 76% of the image representation, which is more than the IWSPM scheme. However, as listed in Table 5.4, it was found that the IWSPM scheme consistently outperforms the $l2$-OWSPM scheme (with the exception of Caltech 101 with 15 training images). These results show that even though less information was contained in the spatial pyramid, it does not necessarily yield a worse performance.

In other words, if we could formulate a way for selecting spatial windows in SPM, we might be able to see improvements in terms of both accuracy and memory consumption. From these findings, it is possible to conclude with utmost confidence that the traditional arrangement of spatial windows in SPM is highly sub-optimal.

Throughout the discussion of the results from the OWSPM, CWSPM and IWSPM schemes, it is possible to list several points of interest:

1. The disjoint arrangement of spatial windows is sub-optimal,
2. The usage of overlapping windows allows for broader context at no additional cost,
3. The current spatial pyramid arrangement of traditional SPM is inefficient, and
4. Recognition systems will benefit from a scheme that learns the best spatial window arrangement for construction of image representation.

6.3 Finding the optimal arrangement

6.3.1 OA-SPM

Let us denote W as the total number of candidate spatial windows (in the case where $L=$ 3, then $W=$ 21). Excluding the case where we did not use any of the windows, there are a total of $(2^W - 1)$ possible arrangements varying from 1 spatial window to all W candidates. To evaluate all of them will be very expensive and intractable. As such, we require a way to learn the optimal arrangement with a cheap yet tractable approach. This thesis adopts a greedy approach by maintaining a set of selected windows and iteratively adding a new unselected window to the set, after which the performance obtained is evaluated.

Let $R(\mathbf{P})$ be the recognition performance of the image representation constructed using $\mathbf{P}$, where $\mathbf{P}$ is defined as in section 4.8. In addition, let $\mathbf{W}$ be the set of all candidate windows. By definition, if a window w is a member of $\mathbf{P}$, then $w \in\mathbf{W}$. Denote the operation of adding a window w to $\mathbf{P}$ as $\mathbb{U}(\mathbf{P}, w)= \mathbf{P} \cup \{w\}$. The best arrangement is then obtained by looking for the best candidate window w_{max} to be included into $\mathbf{P}$ using

$$w_{\text{max}} = \max_{w} R(\mathbb{U}(\mathbf{P}, w)) \tag{6.1}$$

and update $\mathbf{P}= \mathbb{U}(\mathbf{P}, w_{\text{max}})$. This process is then iterated from $\mathbf{P} = \emptyset$ until all candidates have been selected. Using this model, the complexity of the process is a mere $O(W^2)$ as compared to $O(2^W)$ of the exhaustive search. This process is called Optimal Arrangement SPM (OA-SPM).

There are some similarities shared between this proposed model and the models adopted in [60], whereby those models propose to evaluate the representations from a collection of spatial windows (or receptive fields) and then learn the optimal representations. The model proposed in this thesis differs in three main factors. Firstly, OA-SPM utilizes overlapping spatial windows due to

Algorithm 5 OA-SPM

```
Input: Complete mid-level dataset representation $D$.
Output: Optimal arrangement $\mathbf{P}_{\text{opt}}$.
$\mathbf{P}(0) = \emptyset$, $W = \{w_1, w_2, ..., w_{21}\}$, $\mathbf{P}(t)$ denotes the configuration at step t.
for $t = 1$ to $21$ do
    for $w = 1$ to $|W|$, $w \notin \mathbf{P}(t-1)$ do
        $\mathbf{P}_w(t) = \mathbb{U}(\mathbf{P}(t-1), W(w))$.
        Train classifier based on $\mathbf{P}_w(t)$.
        Test the classifier and obtain performance score $R_w(t) = R(\mathbf{P}_w(t))$.
    end for
    $\mathbf{P}(t) = \mathbf{P}(t-1) \cup W(w_{\text{max}}) : w_{\text{max}} = \max_{w \notin \mathbf{P}(t-1)} R_w(t)$.
end for
$\mathbf{P}_{\text{opt}} = \max_t R(\mathbf{P}(t))$.
```

their efficiency, as detailed in Chapter 4. Secondly, the base collections of spatial window candidates greatly differ from each other: 21 spatial windows are selected from the OWSPM/CWSPM in OA-SPM, as opposed to the overcomplete set of windows based on the 4×4 superpixel method proposed in [60], where a total of 100 candidates are evaluated. Finally, the aforementioned paper adopted a learning strategy to optimize the arrangement of each dictionary index k within every spatial window instead of the spatial window itself, as in our proposed learning scheme.

6.3.2 Implementations and results

Let us denote $\mathbf{B}$ as the collection of all mid-level representation inside a specific database of size S. That is,

$$\mathbf{B} = [\mathbf{Z}_1, \mathbf{Z}_2, ..., \mathbf{Z}_S] \tag{6.2}$$

As we want to examine the effect of individual windows, we note that we can write $\mathbf{Z}$ as $[\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_{21}]$ (assuming we are using standard $L=$ 3 spatial pyramid, without loss of generality). As such we can partition $\mathbf{B}$ as

$$\begin{aligned} \mathbf{B} &= [\mathbf{B}_1^T, \mathbf{B}_2^T, ..., \mathbf{B}_{21}^T]^T \\ \mathbf{B}_i &= [\mathbf{Z}_{1i}, \mathbf{Z}_{2i}, ..., \mathbf{Z}_{Si}] \end{aligned} \tag{6.3}$$

	SPM	
Dataset	$\|\mathbf{P}\| = 21$	Peak
15 Scene (100 train)	80.22%	$81.67\%(15)$
Caltech 101 (15 train)	67.00%	$67.02\%(14)$
Caltech 101 (30 train)	74.02%	$74.68\%(18)$

	CWSPM				
Dataset	$\|\mathbf{P}\| = 21$	$\|\mathbf{P}\| = 11$	$\|\mathbf{P}\| = 12$	$\|\mathbf{P}\| = 13$	$\|\mathbf{P}\| = 14$
15 Scene (100 train)	81.54%	81.94%	81.96%	$\mathbf{82.28}\%$	81.54%
Caltech 101 (15 train)	67.57%	66.81%	67.68%	$\mathbf{67.83}\%$	67.12%
Caltech 101 (30 train)	73.72%	74.65%	$\mathbf{74.62}\%$	75.05%	74.21%

Table 6.1: Average recognition rate based on OA-SPM process. Any number inside a bracket signifies the number of spatial windows needed to produce the corresponding result.

Using this notations, we can say that $R(\mathbf{P} = \{w\}) = R(\mathbf{B}_w)$, and

$$R(\{w_1, w_2, ..., w_n\}) = R([\mathbf{B}_{w_1}^T, \mathbf{B}_{w_2}^T, ..., \mathbf{B}_{w_n}^T]^T) \tag{6.4}$$

Using these notations, Algorithm 5 shows the details of OA-SPM implementations as discussed in section 6.3.1.

Extraction of features and training of classifier all follows the approach discussed in sections 4.3.1 to 4.3.4, especially Algorithm 3 and 4. Furthermore, CWSPM is used as it was found to be the most consistent among the two overlapping methods.

Experiments were run using 15 Scene and Caltech 101 databases, as they represent two different problems: scene and object recognitions. The training and testing processes is repeated for 5 times, with the performance average measured. Figure 6.1 shows the progression of performances as more candidates are included into the set for both databases, while a detailed comparison can be observed in Table 6.1. Based on these results, it is possible to see some interesting properties of the SPM arrangement.

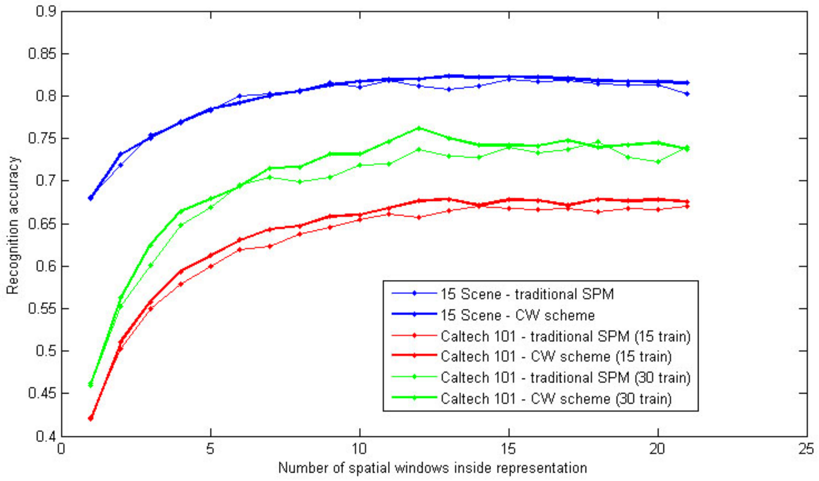


Figure 6.1: The progression of $R(\mathbf{P})$ as more spatial windows is added into $\mathbf{P}$ for CWSPM and traditional SPM. Complete set of spatial windows (the tail of the graphs) performs less effective compared to those in the middle of the graphs. All experiment tested with $\omega = 0$ for CWSPM scheme using ScSPM as baseline.

6.3.3 Discussions

First, including all spatial windows in the image representation does not necessarily yield the best performance. It was found that for both CWSPM and traditional SPM, pyramid configuration with complete set of sub-windows is not the best performing arrangement. It appears that the non-crucial information contained in the spatial windows actually had a detrimental effect to the overall performance of the system. As such, a feature selection process is essential on top of SPM representation.

Furthermore, it was found that the peak performance was achieved from using 11 to 14 candidates out of the full collection. In other words, it consists of nearly 60% of the full collection of windows in the traditional SPM. Compared with the $\mathbf{P}_{2c}$ arrangement, it is possible to cut the memory cost even further, but at the same time achieve better performance than that of $\mathbf{P}_{\text{cw}}$. This result is very important, as it proves that it is possible to increase the accuracy of SPM representation, while at the same time reducing its memory complexity.

Finally, OA-SPM affirms the importance of overlapping windows in the efficiency of an individual spatial window when comparing the findings from CWSPM and traditional SPM. With the overlapping scheme, it is possible to find higher performance from each individual window. Each overlapping spatial window demonstrates a noticeable increase in discriminability as compared to the traditional (disjoint) window.

Additionally, another interesting observation is that the global spatial window (the single window at $l = 0$) is always selected first in all cases. Thus, the results at that particular window, for both traditional SPM and CWSPM schemes, always coincide with each other. To some extent, this finding is fully understandable as the global window contains the most information from the image. This learning scheme suggests that, in any case, the usage of the global window is highly recommended.

The average recognition performance and their optimal spatial window arrangements are presented in Table 6.2 and Figure 6.2 respectively. It can be argued that a unique optimal arrangement should exist for a specific dataset, and it is highly recommended that the optimization process be repeated for each new dataset. OA-SPM yields a significant increase in performance as compared to the traditional SPM, at a typically 40% lesser memory cost. Even though these results are lower than those offered by state-of-the-art technologies, this proposed model can be adopted to any SPM-based image representation to achieve even better performance at lesser memory cost.

As mentioned in Section 5.4.5, finding the global maximum over such a large space without an exhaustive search is not possible. The term optimal that is used throughout this chapter for the proposed method does not necessarily refer to the global maximum out of all possible 2^W windows. If $\mathbf{P}(t)$ denotes the set at iteration t, OA-SPM only considers the window that maximizes the increment $R(\mathbf{P}(t + 1)) - R(\mathbf{P}(t))$. In other words, it is possible that there are other maxima in the space of all possible arrangements that return higher recognition accuracy than those found by OA-SPM.

Dataset	Training	Algorithm	Best OA-SPM		SPM		Improvement	Space saved
	Image		Recognition Rate R_1	$\lvert\mathbf{P}\rvert$	Recognition Rate R_2	$\lvert\mathbf{P}\rvert$	$(R_1 - R_2)$	
15 Scene	100	ScSPM	$\mathbf{82.28 \pm 0.35}\%$	13	$80.28 \pm 0.93\%$	21	2.00	38%
Caltech 101	15	ScSPM	$\mathbf{67.83 \pm 0.66}\%$	13	$67.00 \pm 0.41\%$	21	0.83	38%
Caltech 101	30	ScSPM	$\mathbf{76.24 \pm 0.40}\%$	12	$73.20 \pm 0.54\%$	21	3.04	42%
Caltech 256	15	ScSPM	$\mathbf{32.11 \pm 0.52}\%$	12	$27.73 \pm 0.51\%$	21	4.38	42%
Caltech 256	30	ScSPM	$\mathbf{37.22 \pm 0.38}\%$	12	$34.02 \pm 0.35\%$	21	3.20	42%
Caltech 256	45	ScSPM	$\mathbf{40.20 \pm 0.35}\%$	12	$37.46 \pm 0.55\%$	21	2.74	42%
Caltech 256	60	ScSPM	$\mathbf{42.40 \pm 0.67}\%$	12	$40.14 \pm 0.91\%$	21	2.26	42%
Caltech 101	30	Simulated Fixations	$\mathbf{75.71 \pm 0.44}\%$	13	74.60%	21	1.11	38%
STL-10	500	Simulated Fixations	$\mathbf{63.26 \pm 0.33}\%$	13	61.00%	21	2.26	38%
UIUC-Event	70	Object Bank	$\mathbf{79.13 \pm 0.41}\%$	10	76.30%	21	2.83	52%
MIT-Indoor	80	Object Bank	$\mathbf{39.95 \pm 0.64}\%$	11	37.60%	21	2.35	47%
15 Scene	100	Object Bank	$\mathbf{83.50 \pm 0.39}\%$	11	80.90%	21	2.60	47%

Table 6.2: Recognition rate using optimized arrangement SPM based on the findings of OA-SPM.

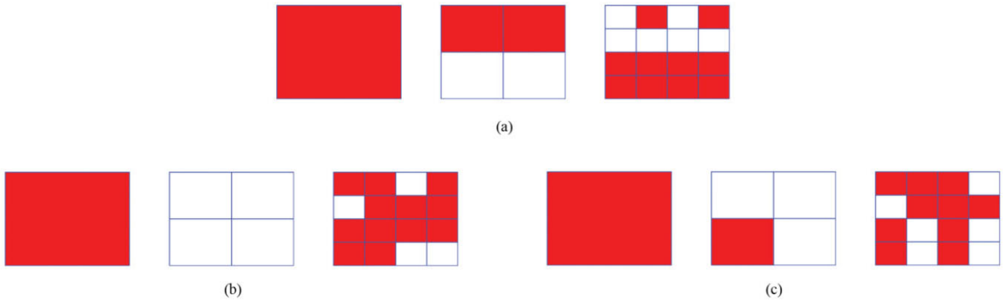
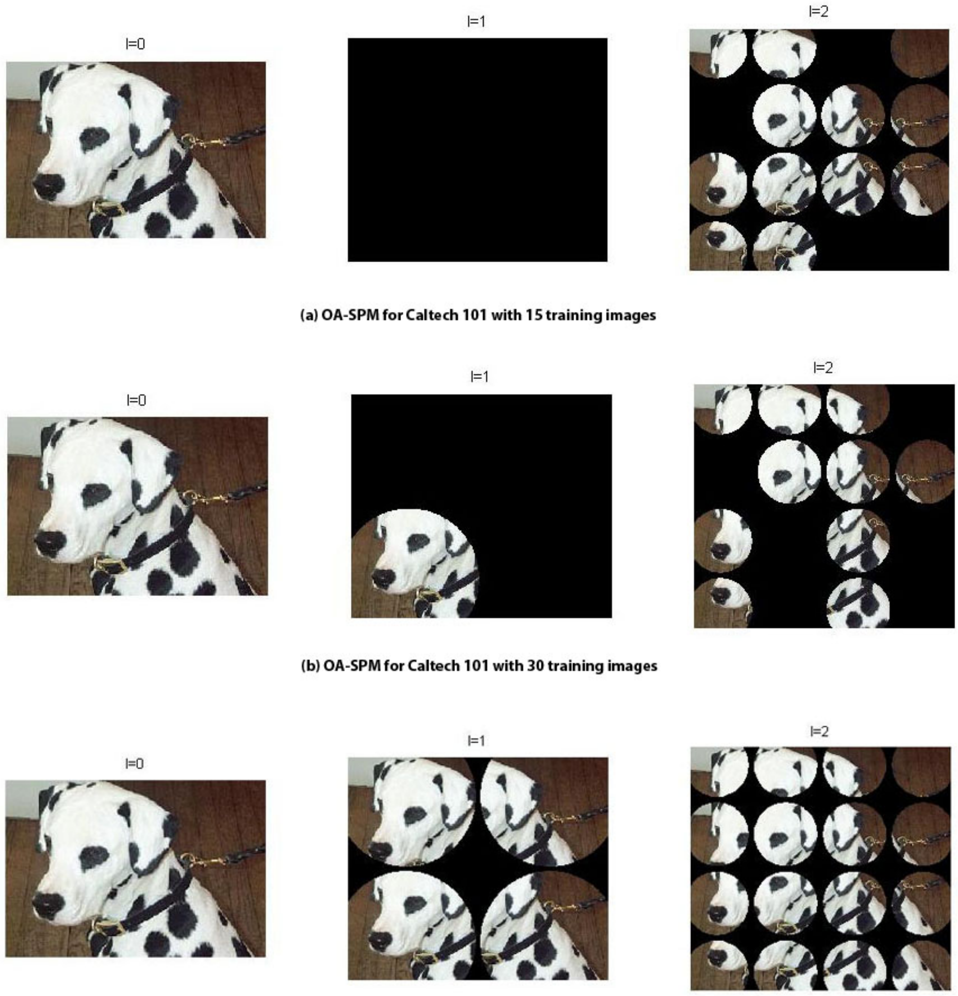


Figure 6.2: Optimized arrangement obtained from OA-SPM for (a) 15 Scene (b) Caltech 101 with 15 training image and (c) Caltech 101 with 30 training image. Overlapping regions are omitted in the illustration for clarity purposes. Shaded regions are selected by OA-SPM to be included into the image representation.

However, this does not cancel the hypothesis that the traditional SPM is sub-optimal. In fact, OA-SPM proves to some extent that there exists a peak (or at least a saturation point) between the half-pyramid and full-pyramid arrangement. The existence of such points gives rise to the necessity of having a selection algorithm.

Additionally, the window arrangement is dataset-specific. OA-SPM needs to be retrained whenever a new dataset is taken into consideration, with the complexity of $O(W^2)$. This means that the training step of the classifier will take longer than those without OA-SPM. However, the testing side will become much faster, as it involves much shorter image representation, which means less computational cost. This is an acceptable trade-off, since its implementation in the real world will focus on user experience; thus, having a faster testing step is more important than a faster training step.

If we look at Figure 6.2 again, we can find out the relation between what OA-SPM learn and the dataset that it is being subjected to. Take the result from 15 Scene dataset for example (6.2a). Interestingly enough, on layer $l = 1$, the two windows at the bottom half of the layer are not included while both top half are. The bottom half, however are included in its entirety at layer $l = 2$. This tells us that when OA-SPM learns for 15 Scene dataset, it is sufficient for it to only consider the top half of the image in broader context, but the bottom



(a) OA-SPM for Caltech 101 with 15 training images

(b) OA-SPM for Caltech 101 with 30 training images

(c) complete CWSPM arragement

Figure 6.3: Examples of OA-SPM arrangement on Caltech 101 dataset with (a) 15 training images, and (b) 30 training images. Row (c) shows the complete arrangement of CWSPM. Black regions where sub-windows are missing when compared to (c) means that the sub-window are discarded.

half of the image need extra details for recognition to return best result. This suits how most images in this set, since an image of a scene normally have more information at the bottom half, and the top half either does not contribute much to classification (i.e. outdoor pictures will most likely contain sky at the top half, indoor picture will contain ceiling, both are not as informative) or considerably less object and is enough to be covered by the larger $l = 1$ sub-windows.

As we move to Caltech 101 (6.2b and 6.2c), the narrative becomes very different. The $l = 1$ layer is selected less (only once at Caltech 101 with 30 training image, that is 1 in 8), and OA-SPM favors sub-windows from $l = 2$

layer more. Furthermore, there is a tendency to pick the middle sub-window (those at row 2/3 and column 2/3). This is consistent with Caltech 101 images since most objects in this dataset are centered. The tendency to pick up more detailed description from $l = 2$ rather than from $l = 1$ is also not surprising, as objects came in varying size and texture. Our results show good adaptability to various dataset, and it shows that OA-SPM is able to not only learn the optimal arrangement, but also the locations where most of the informations are located. As such, it can also be used to analyze a dataset in a broader context.

Furthermore, if we compare $l = 1$ layer and $l = 2$ from each of the dataset, surprisingly there does not seem to be much overlap between the regions covered by the two layers. Caltech 101 with 15 training images does not have any overlapping regions between $l = 1$ and $l = 2$ (because $l = 1$ is completely discarded), while 15 Scene and Caltech 101 with 30 training images both have overlap with the size of two $l = 2$ sub-windows (12.5% of image size). As such, we can deduce that there is a tendency to not cover an area of an image when it has been covered by another sub-windows before. This directly contradicts the traditional SPM, and evidently, we are able to obtain better performances and better memory cost compared to it. The finding supports our claim (and conclusion) that the traditional SPM arrangement is not optimal.

Another interesting findings from Figure 6.2 is that the global sub-window at $l = 0$ is always selected in all experiments. This tells us that the 0^{th} layer in SPM holds the most important information regardless of dataset used.

6.4 Approximating the optimal arrangement

The model given in Section 6.3 gives us a reliable way to obtain the optimal spatial window arrangement for the SPM model that is both cheaper and more accurate than the traditional SPM arrangement. However, OA-SPM involves the evaluation of $(W - k + 1)$ arrangements at step k, which can prove to be very costly as the number of windows for consideration (W) increases. With the current model, it takes $O(W^2)$ windows to compute. While this is much more convenient than a complete evaluation of all possible arrangements as given by $O(2^W)$, dropping the computational complexity to a linear cost would make prediction faster and tractable to more advanced datasets.

To this end, an assumption on a linear relationship between recognition performance $R(\mathbf{P})$ and individual window performance $R(w_i)$ is made ($\mathbf{P} = \{w_1, w_2, \ldots, w_n\}$). Assume that the base window at $l = 0$ is always included inside $\mathbf{P}$ without loss of generality. Then, the index $i = 1$ is assigned to this window (i.e. w_1 denotes this global window). As any arrangement $\mathbf{P}$ will contain the base window from this point onwards, the index 1 will be omitted for brevity. As such, let us redefine the notation of recognition accuracy made by the set $\{w_1, w_i\}$ as R_i, i.e. $R_i = R(\{w_1, w_i\})$. Similarly, $R_{ij} = R(\{w_1, w_i, w_j\})$. This redefinition is done to improve the readability of this section.

To model the relationship between spatial pyramid arrangement and recognition accuracy, the following assumption are made:

$$\begin{aligned} R_i &= R_1 + \Delta_i \\ R_{ij} &= R_1 + \Delta_i + \Delta_j + \epsilon_{ij} \end{aligned} \tag{6.5}$$

In these equations, Δ_i is the adjustment factor needed to correct the recognition accuracy when including w_i to w_1. Similarly and ϵ_{ij} denotes the correction factor for the linear model of the next order. Essentially, the assumption made will approximate recognition performance by considering it as being linear to an extent, and then learn a correction factor to straighten the model. The assumption in Eq. (6.5) is then extended to a larger number of windows. For example, in a four-window condition with $\mathbf{P} = \{w_1, w_i, w_j, w_k\}$, the recognition performance is expressed as

$$R_{ijk} = R_1 + \Delta_i + \Delta_j + \Delta_k + \epsilon_{ijk} \tag{6.6}$$

Dataset	Training Image	Best OA-SPM Recognition Rate	$\lvert\mathbf{P}\rvert$	Best LA-SPM Recognition Rate	$\lvert\mathbf{P}\rvert$
15 Scene	100	$\mathbf{82.28 \pm 0.35\%}$	13	$82.08 \pm 0.43\%$	12
Caltech 101	15	$\mathbf{67.83 \pm 0.66\%}$	13	$67.61 \pm 0.51\%$	14
Caltech 101	30	$\mathbf{76.24 \pm 0.40\%}$	12	$75.10 \pm 0.63\%$	14
Caltech 256	15	$\mathbf{32.11 \pm 0.52\%}$	12	$31.98 \pm 0.62\%$	13
Caltech 256	30	$\mathbf{37.22 \pm 0.38\%}$	12	$37.04 \pm 0.44\%$	12
Caltech 256	45	$\mathbf{40.20 \pm 0.35\%}$	12	$40.16 \pm 0.49\%$	12
Caltech 256	60	$\mathbf{42.40 \pm 0.67\%}$	12	$42.14 \pm 0.80\%$	13

Table 6.3: Comparing the recognition rate of OA-SPM and LA-SPM.

The value of Δ_i can be easily obtained using empirical method by considering all R_i values in the set. However, to compute the correction factor ϵ, another assumption is needed:

$$\epsilon_{a_1 a_2 \ldots a_n} = \frac{1}{n}(\epsilon^{-a_1}_{a_1 a_2 \ldots a_n} + \epsilon^{-a_2}_{a_1 a_2 \ldots a_n} + \ldots + \epsilon^{-a_n}_{a_1 a_2 \ldots a_n}) \tag{6.7}$$

In this equation, the subscript a_i denotes a window w_{a_i} within the set; while the superscript $-a_i$ denotes a window w_{a_i} that is excluded from the selected set (i.e. $\epsilon_{ijk} = (\epsilon_{ij} + \epsilon_{ik} + \epsilon_{jk})/3$). By using this assumption and some mathematical manipulations, the approximate performance can be computed as:

$$\begin{aligned} R_{a_1 a_2 \ldots a_n} &= R_1 + \sum_{i=1}^{n} \Delta_i + \epsilon_{a_1 a_2 \ldots a_n} \\ &= \frac{(n-2)!}{n!} \sum_{i,j}^{i \neq j} R_{a_i a_j} + (1 - \frac{2}{n}) \sum_{i=1}^{n} \Delta_i \\ &= \frac{1}{n(n-1)} \sum_{i,j}^{i \neq j} R_{a_i a_j} + \frac{n-2}{n} \sum_{i=1}^{n} \Delta_i \end{aligned} \tag{6.8}$$

That is, given a set $\mathbf{P}$, $R(\mathbf{P})$ can be approximated by averaging the recognition performance of all possible 2-window combinations from $\mathbf{P}$ with a correction factor obtained from the increment coefficient Δ_i. This simple approximation method provides us with a fast and inexpensive way to select the optimal arrangement for SPM.

The framework is tested using 15 Scene, Caltech 101 and Caltech 256

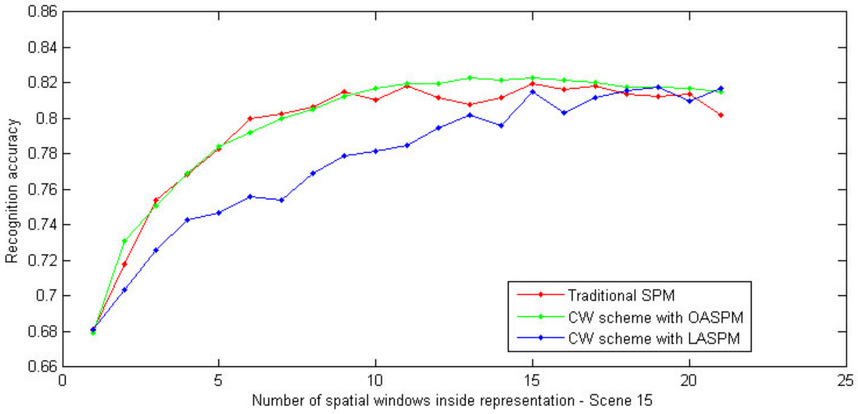


Figure 6.4: Recognition performance vs number of window on 15 Scene. Result tested using Traditional SPM, CWSPM with OA-SPM, and CWSPM with LA-SPM.

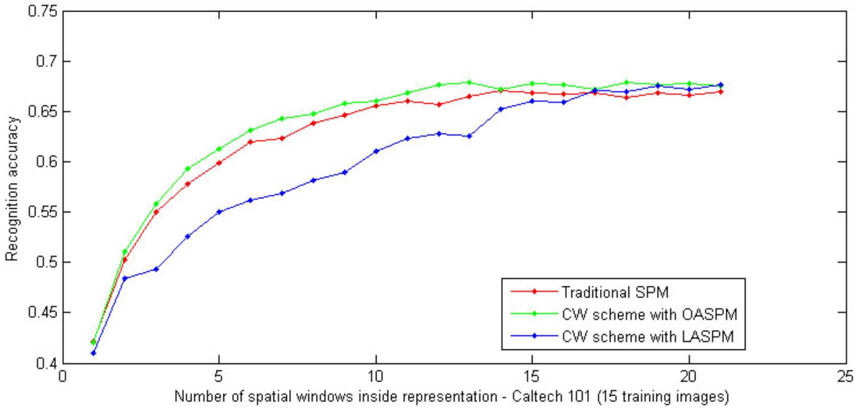


Figure 6.5: Recognition performance vs number of window on Caltech 101 with 15 training images. Result tested using Traditional SPM, CWSPM with OA-SPM, and CWSPM with LA-SPM.

datasets. Let us denote $|\mathbf{P}|$ as the cardinality of the set $\mathbf{P}$, and select the best arrangement using the linear approximation for $|\mathbf{P}| = 1$ (containing the $l = 0$ window) to $|\mathbf{P}| = W$, after which we compare the results with traditional SPM and OA-SPM (refer to Figure 6.4 to 6.6 for the complete results of the experiment). It was found that linear approximation returned similar peaks when compared with OA-SPM ($11 \leq |\mathbf{P}| \leq 14$), with comparable performance. However, the performance from $|\mathbf{P}| = 1$ until the peak builds up slower. This is understandable as the arrangement was obtained from the approximation of the performance at a particular cardinality of set.

On the other hand, it is interesting to note that even though performance un-

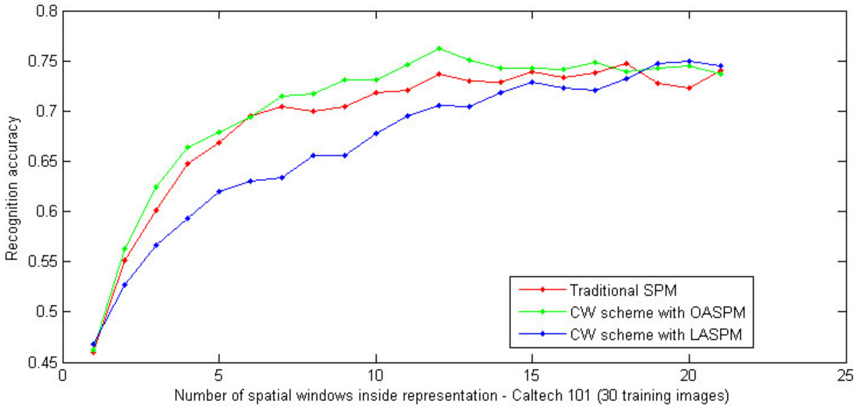


Figure 6.6: Recognition performance vs number of window on Caltech 101 with 30 training images. Result tested using Traditional SPM, CWSPM with OA-SPM, and CWSPM with LA-SPM.

der the linear approximation increases at a slower rate than performance under OA-SPM, their peaks generally coincide within a certain number of cardinality of set, with similar performance results. Both OA-SPM and the linearly approximated arrangement (referred to as LA-SPM from here on) perform better than the traditional SPM, in both cases of optimized and complete sets, for all tested datasets. Based on this result, it can be concluded that the spatial window arrangement will reach a saturation point as more windows are added into the set for representation, which also represents the system's peak performance. As more windows are added to the set, recognition performance does not improve; in fact, it deteriorates. As such, contrary to current practice, utilizing the complete set of spatial windows in SPM is not optimal.

6.5 Comparison to current state-of-the-art approach

It has been shown in table 6.2 that the proposed approach is not just applicable to BoW, but also to other families of image representation, through the application of OA-SPM and LA-SPM on ObjectBank and deep learning with simulated fixations. It is possible to achieve up to 2.83% and 2.26% improvements for ObjectBank and simulated fixations, respectively. It shows that the proposed optimization approach to Spatial Pyramid Matching is not confined to the BoW approach, but to spatial pyramid usage in general. More importantly, it is possible to improve the performance of deep convolutional neural network approaches, using the proposed method.

Currently, Convolutional Neural Networks have consistently shown great and rapid improvements in the field of image recognition. During the course of completing this thesis, several works which significantly outperform the proposed approach here have been published. One of the best results recently published in [86] reports results (with Caltech 101 and PASCAL VOC 2007) as high as $93.42 \pm 0.5\%$ accuracy (the author would like to point out that that work was published during the process of revising this thesis, and was therefore unable to investigate the effects of OA-SPM or LA-SPM when applied to the method). That result is significantly higher than what is reported in this thesis, even on the CNN-based approach.

However, it is interesting that [86] utilizes SPM to circumvent the need of having an image input of fixed size, which would reduce recognition accuracy. As discussed in Section 2.9, the second part of the CNN-based approach utilizes highly connected layers, which requires an input of fixed size to be fed into the network. Prior to [86], the patch fed to the network was obtained using the sliding window technique, which resulted on the omission (clipping) of important information. He et al. circumvented this by using warped versions of the window containing the whole object, and fed the SPM response to the network. As such, it is possible to achieve more accurate recognition results with faster computational cost.

Although it does seem that this work performs poorly as compared to current state-of-the-art technologies, it was not the aim of this work to compete with them from the start. Rather, the aim of this thesis is to enforce the importance of spatial information, and repudiate the assumption that SPM is already efficient enough. In fact, the SPM used in [86] follows the traditional architecture, and it should benefit from the proposed method, pushing its results to an even higher accuracy.

6.6 Concluding remarks

This chapter presented a detailed investigation on the effectiveness of the traditional SPM architecture. Based on the findings from overlapping and interleaved windows, it is possible to conclude that the traditional SPM model is sub-optimal in at least two regards: its disjoint arrangement of spatial windows and the construction of image representations. It was found that by introducing overlapping spatial windows, the discriminative power of each window was strengthened, and half of the information in the complete SPM pyramid contributed to a mere 1% in recognition performance.

As such, this chapter introduced two schemes to learn the optimal spatial window arrangement for SPM, OA-SPM and LA-SPM. Both frameworks show us that the feature selection process is essential, on top of the usual SPM pipeline, to filter out redundant spatial images in image representation. By doing so, it is possible to consistently achieve significant performance improvements (up to 4.38%) and at the same time reduce the memory cost by nearly 40%, compared to that within the traditional SPM arrangement.

It is important to note that some may consider the graph in Figure 6.4-6.6 to be inconclusive to claim that the recognition performance peaks at a certain arrangement when $11 \leq |\mathbf{P}| \leq 14$; it can also be interpreted as a saturation point where the results plateau with slight variations. However, this does not contradict the hypothesis that SPM is not optimal, as it is possible to achieve a very similar result with much lower spatial windows adopted when picking the point where recognition starts to saturate.

Chapter 7

Conclusion and Recommendations for Future Work

7.1 Conclusion

Spatial information is crucial to the task of image recognition. The fact that it is useful and powerful should be apparent to vision researchers. This thesis presents an example of the influential impact of spatial information on the quality of image representation, learning processes, and system performance, through deterministic approaches such as Support Vector Machine (SVM) and non-parametric approaches such as the Hierarchical Dirichlet Process (HDP). This work can be categorized in two major parts: improving DHDP and introducing new paradigms for Spatial Pyramid Matching.

The work on improving HDP through modified-DHDP with approximate shape masks validate our belief that spatial information is an integral part to a successful object recognition. While DHDP suffers greatly from the "rich-get-richer" effect, the addition of spatial information in the form of cardinality coefficient and approximate shape mask led to the mitigation of the effect. This is why the proposed modified-DHDP in this thesis are able to reach better performances (by 11%) compared to the original DHDP which does not utilize spatial information in any kind. The proposed method tackles the problem of "rich-get-richer" effect on two fronts. Cardinality coefficient makes sure that only relevant codewords are grouped together by rejecting noise patches, while at the same time approximate shape masks filter out noise patches that do not belong to the predicted object shape.

As evident in DHDP, "bag-of-words" (BoW) method suffers from the inherent assumption that spatial information can be discarded. The modified-DHDP demonstrate that there is a need to include spatial information back into BoW image representation. The second part of this thesis focused on this goal.

The Spatial Pyramid Matching (SPM) model is a staple in BoW-based representation. SPM provides rough spatial information from the image into BoW image representation. SPM is used by many works since they are both cheap and simple to implement, but offer considerable improvements to the final results of object recognition. However, most works in computer vision tends to take SPM without questioning its efficiency. This work aims to investigate SPM and attempts to improve the model.

This work improves the traditional SPM model by proposing two novel spatial window arrangements in overlapping rectangular windows (OWSPM) and overlapping circular windows (CWSPM). These arrangements question the traditional SPM spatial window arrangement, which is disjoint in nature. Replacing them with overlapping windows have been shown to improve recognition accuracy upwards of 3.68%. This improvement occurs without imposing additional strain on memory allocation or increasing training cost. It was also shown that the lowest layer in the pyramid at $l = 2$ contains the bulk of the information, and 25% of the information contained in the preceding layers contributes to a mere 1% to recognition accuracy. These findings propel us to push our work further, as it proves that traditional SPM model is sub-optimal.

The interleaved window (IWSPM) scheme confirms the hypothesis that the architecture of SPM is sub-optimal. With a systematic way of selecting the arrangement of spatial windows, it is possible to push the boundaries of SPM further in the task of object and scene recognition. This work introduces proposals for finding this optimal arrangement using two approaches: OA-SPM and LA-SPM. For both optimization procedures it was found that approximately 40% of the spatial windows do not contribute additional information to the rest, as well as having a detrimental effect on recognition performance. It was found that excluding them would not only lead to a reduction in cost, but also improvements in performance. Improvements as high as 4.38% with OA-SPM were found during the experiments conducted for this thesis. This confirms our claim that SPM is sub-optimal, and at the same time validate our proposed optimization method.

7.2 Recommendations for future work

Several extensions of OA-SPM come to mind:

7.2.1 Finding global maximum within all possible $\mathbf{P}$

As mentioned in Chapter 6, OA-SPM searches along the line that maximizes the increment of accuracy between the t^{th} and $(t + 1)^{\text{th}}$ iteration. While this can be an acceptable optimization, it is not sufficient to guarantee the global maximum throughout all possible $\mathbf{P}$, and is only a local maximum. Therefore, the OA-SPM arrangement does not refer to the real best arrangement possible, which may even be shorter than those found from OA-SPM. In any case, further research on this topic could be interesting.

7.2.2 Class-specific arrangement and adaptive classifier

It is worth noting that OA-SPM utilizes the same optimized arrangement for all classes. This might not be optimal as the arrangement that comes as a product of OA-SPM is evaluated to meet the need of all classes in contention. The idea is to extend OA-SPM to cater to the need of a specific class, and in doing so produce an optimal arrangement for each class.

The caveat of such an approach is that the classifier also needs to be adaptive. When considering one class, a specific arrangement must be used, and when considering another class, another arrangement should be used. Furthermore, the results of each class-specific classifier will not be directly comparable, and will need some kind of normalization. Constructing the optimal arrangement for each class might be a simple task, but formulating the classifier will prove to be a challenge.

7.2.3 Application to other family of methods

While we have presented our models largely using ScSPM as a baseline, the BoW family of methods is losing ground to the Fisher Vector (FV) and Deep Learning (CNN). Although, in this thesis we also demonstrate that OA-SPM can be readily extended to CNN, OA-SPM was not created specifically for it. A proper investigation into adapting OA-SPM to Deep Learning or other more current topics may prove worthwhile.

Publications

Conferences

1. Kristo and Chua, C.S., "Utilizing region cardinality and dependency for object categorization in non-parametric Bayesian framework", ICICS, 8^{th} IEEE international conference on. 2011.
2. Kristo and Chua, C.S., "Utilizing overlapping windows in spatial pyramid matching", MVA, 13^{th} IAPR International conference on. 2013.
3. Kristo and Chua, C.S., "Image representation for object recognition: utilizing overlapping windows in spatial pyramid matching", ICIP, IEEE international conference on. 2013.
4. Kristo and Chua, C.S., “Optimized Window Arrangement for Spatial Pyramid Matching Scheme”, ICPR, 22^{nd} IAPR international conference on. 2014.

Journals

1. Kristo and Chua, C.S., “Cost Effective Window Arrangement for Spatial Pyramid Matching”, Journal of Visual Communication and Image Representation, May 2015, (29), pp. 79-88.

Bibliography

[1] Gang Wang, Ye Zhang, and Li Fei-Fei, "Using dependent regions for object categorization in a generative framework," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, 2006, vol. 2, pp. 1597–1604.

[2] D.G. Lowe, "Object recognition from local scale-invariant features," in *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, 1999, vol. 2, pp. 1150–1157 vol.2.

[3] Marcin Marszałek and Cordelia Schmid, "Accurate object recognition with shape masks," *International Journal of Computer Vision*, vol. 97, no. 2, pp. 191–209, 2012.

[4] Ashish Kapoor, Kristen Grauman, Raquel Urtasun, and Trevor Darrell, "Gaussian processes for object categorization," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 169–188, 2010.

[5] Tommer. Leyvand, Casey. Meekhof, Yi-Chen Wei, Jian Sun, and Baining Guo, "Kinect identity: Technology and experience," *Computer*, vol. 44, no. 4, pp. 94–96, April 2011.

[6] Moshe Bar, "Visual objects in context.," *Nat Rev Neurosci*, vol. 5, no. 8, pp. 617–29, Aug. 2004.

[7] Irving Biederman, "On the semantics of a glance at a scene," pp. 213–263, 1981.

[8] Aude Oliva and Antonio Torralba, "The role of context in object recognition," *Trends in Cognitive Sciences*, vol. 11, no. 12, pp. 520 – 527, 2007.

[9] Lior Wolf and Stanley Bileschi, "A critical view of context," *International Journal of Computer Vision*, vol. 69, no. 2, pp. 251–261, 2006.

[10] Jasper R.R. Uijlings, Arnold W.M. Smeulders, and Remko J.H. Scha, "The visual extent of an object," *International Journal of Computer Vision*, vol. 96, no. 1, pp. 46–63, 2012.

[11] Li-Jia Li, Gang Wang, and Li Fei-Fei, "Optimol: automatic online picture collection via incremental model learning," in *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, June 2007, pp. 1–8.

[12] Gregory Griffin, Alex Holub, and Pietro Perona, "Caltech-256 Object Category Dataset," Tech. Rep. CNS-TR-2007-001, California Institute of Technology, 2007.

[13] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, 2006, vol. 2, pp. 2169–2178.

[14] Mark Everingham, Luc Van Gool, Christopher K.I. Williams, John Winn, and Andrew Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.

[15] Alex Krizhevsky, "Learning Multiple Layers of Features from Tiny Images," M.S. thesis, 2009.

[16] Li-Jia Li, Hao Su, Li Fei-Fei, and Eric P. Xing, "Object bank: A high-level image representation for scene classification &amp; semantic feature sparsification," in *Advances in Neural Information Processing Systems 23*, J.D. Lafferty, C.K.I. Williams, J. Shawe-Taylor, R.S. Zemel, and A. Culotta, Eds., pp. 1378–1386. Curran Associates, Inc., 2010.

[17] Thomas Berg and Peter N. Belhumeur, "How do you tell a blackbird from a crow?," in *Proc. Int. Conf. Computer Vision (ICCV)*, December 2013.

[18] Peter Welinder, Steve Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and Pietro Perona, "Caltech-UCSD Birds 200," Tech. Rep. CNS-TR-2010-001, California Institute of Technology, 2010.

[19] Bangpeng Yao, "A codebook-free and annotation-free approach for fine-grained image categorization," in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Washington, DC, USA, 2012, CVPR '12, pp. 3466–3473, IEEE Computer Society.

[20] Zellig Harris, "Distributional structure," *Word*, vol. 10, no. 23, pp. 146–162, 1954.

[21] David M. Blei, Andrew Y. Ng, Michael I. Jordan, and John Lafferty, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 2003, 2003.

[22] Jinguo Zhang, Marcin Marszałek, Svetlana Lazebnik, and Cordelia Schmid, "Local features and kernels for classification of texture and object categories: A comprehensive study," *International Journal of Computer Vision*, vol. 73, no. 2, pp. 213–238, 2007.

[23] Jinjun Wang, Jianchao Yang, Kai Yu, Fengjun Lv, T. Huang, and Yihong Gong, "Locality-constrained linear coding for image classification," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, June 2010, pp. 3360–3367.

[24] Shenghua Gao, Ivor W. Tsang, Liang-Tien Chia, and Peilin Zhao, "Local features are not lonely 2013; laplacian sparse coding for image classification," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, June 2010, pp. 3555–3561.

[25] Jianchao Yang, Kai Yu, Yihong Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, June 2009, pp. 1794–1801.

[26] Shu Kong and Donghui Wang, "A dictionary learning approach for classification: Separating the particularity and the commonality," in *Computer Vision – ECCV 2012*, Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid, Eds., vol. 7572 of *Lecture Notes in Computer Science*, pp. 186–199. Springer Berlin Heidelberg, 2012.

[27] Florent Perronnin, Jorge Sánchez, and Thomas Mensink, "Improving the Fisher kernel for large-scale image classification," in *European Conference on Computer Vision (ECCV)*, 2010.

[28] Josip Krapac, Jakob Verbeek, and Frederic Jurie, "Modeling spatial layout with fisher vectors for image categorization," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, Nov 2011, pp. 1487–1494.

[29] Herve Jegou, Matthijs Douze, Cordelia Schmid, and P. Perez, "Aggregating local descriptors into a compact image representation," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, June 2010, pp. 3304–3311.

[30] Herve Jegou and Andrew Zisserman, "Triangulation embedding and democratic aggregation for image search," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, June 2014, pp. 3310–3317.

[31] Honglak Lee, Alexis Battle, Rajat Raina, and Andrew Y. Ng, "Efficient sparse coding algorithms," in *Advances in Neural Information Processing Systems 19*, B. Schölkopf, J.C. Platt, and T. Hoffman, Eds., pp. 801–808. MIT Press, 2007.

[32] Lubomir Bourdev, *Poselets and Their Applications in High-Level Computer Vision*, Ph.D. thesis, EECS Department, University of California, Berkeley, May 2012.

[33] Lubomir Bourdev and Jitendra Malik, "Poselets: Body part detectors trained using 3d human pose annotations," in *International Conference on Computer Vision (ICCV)*, 2009.

[34] Pedro Felzenszwalb, David McAllester, and Deva Ramanan, "A discriminatively trained, multiscale, deformable part model," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, June 2008, pp. 1–8.

[35] Lingqiao Liu, Lei Wang, and Xinwang Liu, “In defense of soft-assignment coding,” in *Computer Vision (ICCV), 2011 IEEE International Conference on*, Nov 2011, pp. 2486–2493.

[36] Piotr Koniusz, Fei Yan, and Krystian Mikolajczyk, “Comparison of mid-level feature coding approaches and pooling strategies in visual concept detection,” *Computer Vision and Image Understanding*, vol. 117, no. 5, pp. 479 – 492, 2013.

[37] Guangcan Liu, Zhouchen Lin, Xiaoou Tang, and Yong Yu, “A hybrid graph model for unsupervised object segmentation,” in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, Oct 2007, pp. 1–8.

[38] Andreas Opelt, Axel Pinz, and Andrew Zisserman, “Learning an alphabet of shape and appearance for multi-class object detection,” *International Journal of Computer Vision*, vol. 80, no. 1, pp. 16–44, 2008.

[39] David D. Lewis, “Naive (bayes) at forty: The independence assumption in information retrieval,” in *Machine Learning: ECML-98*, Claire Nédellec and Céline Rouveirol, Eds., vol. 1398 of *Lecture Notes in Computer Science*, pp. 4–15. Springer Berlin Heidelberg, 1998.

[40] Koray Kavukcuoglu, Marc’Aurello Ranzato, Rob Fergus, and Yann LeCun, “Learning invariant features through topographic filter maps,” in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, June 2009, pp. 1605–1612.

[41] Jianxin Wu and James M. Rehg, “Beyond the euclidean distance: Creating effective visual codebooks using the histogram intersection kernel,” in *Computer Vision, 2009 IEEE 12th International Conference on*, Sept 2009, pp. 630–637.

[42] Xiaochun Cao, Xingxing Wei, Yahong Han, Yi Yang, Nicu Sebe, and Alexander Hauptmann, “Unified dictionary learning and region tagging with hierarchical sparse representation,” *Computer Vision and Image Understanding*, vol. 117, no. 8, pp. 934 – 946, 2013.

[43] Hui-Lan Luo, Hui Wei, and Fan-Xing Hu, “Improvements in image categorization using codebook ensembles,” *Image and Vision Computing*, vol. 29, no. 11, pp. 759 – 773, 2011.

[44] Xinggang Wang, Baoyuan Wang, Xiang Bai, Wenyu Liu, and Zhuowen Tu, “Max-margin multiple-instance dictionary learning,” June 2013, ICML.

[45] Mohammad Norouzi and David J. Fleet, “Cartesian k-means.,” in *CVPR*. 2013, pp. 3017–3024, IEEE.

[46] Jingjing Yang, Yuanning Li, Yonghong Tian, Lingyu Duan, and Wen Gao, “Group-sensitive multiple kernel learning for object categorization,” in *Computer Vision, 2009 IEEE 12th International Conference on*, Sept 2009, pp. 436–443.

[47] Andrea Vedaldi, Varun Gulshan, Manik Varma, and Andrew Zisserman, "Multiple kernels for object detection," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2009.

[48] Thomas Mensink, Jakob Verbeek, Florent Perronnin, and Gabriela Csurka, "Metric Learning for Large Scale Image Classification: Generalizing to New Classes at Near-Zero Cost," in *ECCV 2012 - 12th European Conference on Computer Vision*, Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid, Eds., Florence, Italy, Oct. 2012, vol. 7573, pp. 488–501, Springer.

[49] Y-Lan. Boureau, Francis Bach, Yann LeCun, and Jean Ponce, "Learning mid-level features for recognition," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, June 2010, pp. 2559–2566.

[50] Zechao Li, Jing Liu, and Hanqing Lu, "Structure preserving non-negative matrix factorization for dimensionality reduction," *Computer Vision and Image Understanding*, vol. 117, no. 9, pp. 1175 – 1189, 2013.

[51] Yu Zhang, Jianxin Wu, and Jianfei Cai, "Compact representation for image classification: To choose or to compress?," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, June 2014, pp. 907–914.

[52] Jianchao Yang, Kai Yu, and Thomas Huang, "Efficient highly over-complete sparse coding using a mixture model," in *Computer Vision – ECCV 2010*, Kostas Daniilidis, Petros Maragos, and Nikos Paragios, Eds., vol. 6315 of *Lecture Notes in Computer Science*, pp. 113–126. Springer Berlin Heidelberg, 2010.

[53] Kihyuk Sohn, Dae Yon Jung, Honglak Lee, and Alfred O. Hero, "Efficient learning of sparse, distributed, convolutional feature representations for object recognition," in *Proceedings of the 2011 International Conference on Computer Vision*, Washington, DC, USA, 2011, ICCV '11, pp. 2643–2650, IEEE Computer Society.

[54] Liefeng Bo, Xiaofeng Ren, and D. Fox, "Multipath sparse coding using hierarchical matching pursuit," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, June 2013, pp. 660–667.

[55] Will Zou, Shenghuo Zhu, Kai Yu, and Andrew Y. Ng, "Deep learning of invariant features via simulated fixations in video," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, Eds., pp. 3203–3211. Curran Associates, Inc., 2012.

[56] Jiquan Ngiam, Zhenghao Chen, Daniel Chia, Pang W. Koh, Quoc V. Le, and Andrew Y. Ng, "Tiled convolutional neural networks," in *Advances in Neural Information Processing Systems 23*, J. Lafferty, C. Williams, J. Shawe-taylor, R.s. Zemel, and A. Culotta, Eds., pp. 1279–1287. 2010.

[57] Quoc Le, Marc'Aurelio Ranzato, Rajat Monga, Matthieu Devin, Kai Chen, Greg Corrado, Jeff Dean, and Andrew Ng, "Building high-level features

using large scale unsupervised learning," in *International Conference in Machine Learning*, 2012.

[58] Shengye Yan, Xinxing Xu, Dong Xu, Stephen Lin, and Xuelong Li, "Beyond spatial pyramids: A new feature extraction framework with dense spatial sampling for image classification," in *Computer Vision – ECCV 2012*, Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid, Eds., vol. 7575 of *Lecture Notes in Computer Science*, pp. 473–487. Springer Berlin Heidelberg, 2012.

[59] Jianxiong 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 (CVPR), 2010 IEEE Conference on*, June 2010, pp. 3485–3492.

[60] Yangqing Jia, Chang Huang, and T. Darrell, "Beyond spatial pyramids: Receptive field learning for pooled image features," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, June 2012, pp. 3370–3377.

[61] Timor Kadir and Michael Brady, "Saliency, scale and image description," *International Journal of Computer Vision*, vol. 45, no. 2, pp. 83–105, 2001.

[62] Krystian Mikolajczyk and Cordelia Schmid, "Scale and affine invariant interest point detectors," *International Journal of Computer Vision*, vol. 60, no. 1, pp. 63–86, 2004.

[63] Tony Lindeberg, "Feature detection with automatic scale selection," *International Journal of Computer Vision*, vol. 30, no. 2, pp. 79–116, 1998.

[64] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool, "Speeded-up robust features (surf)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346 – 359, 2008, Similarity Matching in Computer Vision and Multimedia.

[65] Navneet Dalal and Bill Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, June 2005, vol. 1, pp. 886–893 vol. 1.

[66] Judea Pearl, *Probabilistic reasoning in intelligent systems : networks of plausible inference*, The Morgan Kaufmann series in representation and reasoning. Morgan Kaufmann, San Mateo (Calif.), 1988.

[67] Yee Whye Teh, Michael I. Jordan, Matthew J. Beal, and David M. Blei, "Hierarchical Dirichlet processes," *Journal of the American Statistical Association*, vol. 101, no. 476, pp. 1566–1581, 2006.

[68] Michael D. Escobar, "Estimating Normal Means with a Dirichlet Process Prior," *Journal of the American Statistical Association*, vol. 89, no. 425, pp. 268–277, 1994.

[69] Radford M. Neal, “Markov chain sampling methods for dirichlet process mixture models,” *Journal of Computational and Graphical Statistics*, vol. 9, no. 2, pp. 249–265, 2000.

[70] Shivani Agarwal, Aatif Awan, and Dan Roth, “Learning to detect objects in images via a sparse, part-based representation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 11, pp. 1475–1490, Nov. 2004.

[71] Bastian Leibe, Aleš Leonardis, and Bernt Schiele, “Robust object detection with interleaved categorization and segmentation,” *International Journal of Computer Vision*, vol. 77, no. 1-3, pp. 259–289, 2008.

[72] Martin Fussenegger, Andreas Opelt, and Axel Pinz, “Object localization/segmentation using generic shape priors,” in *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, 2006, vol. 4, pp. 41–44.

[73] Li Deng and Dong Yu, “Deep learning: Methods and applications,” *Found. Trends Signal Process.*, vol. 7, no. 3&#8211;4, pp. 197–387, June 2014.

[74] David Liu, Gang Hua, Paul Viola, and Tsuhan Chen, “Integrated feature selection and higher-order spatial feature extraction for object categorization,” in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, June 2008, pp. 1–8.

[75] Ka Y. Hui, “Direct modeling of complex invariances for visual object features,” in *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, Sanjoy Dasgupta and David Mcallester, Eds. May 2013, vol. 28, pp. 352–360, JMLR Workshop and Conference Proceedings.

[76] Roland Memisevic and Georgios Exarchakis, “Learning invariant features by harnessing the aperture problem.,” in *ICML (3)*. 2013, vol. 28 of *JMLR Proceedings*, pp. 100–108, JMLR.org.

[77] F. Shahbaz Khan, J. van de Weijer, and M. Vanrell, “Top-down color attention for object recognition,” in *Computer Vision, 2009 IEEE 12th International Conference on*, Sept 2009, pp. 979–986.

[78] MichaelC. Burl, Markus Weber, and Pietro Perona, “A probabilistic approach to object recognition using local photometry and global geometry,” in *Computer Vision — ECCV’98*, Hans Burkhardt and Bernd Neumann, Eds., vol. 1407 of *Lecture Notes in Computer Science*, pp. 628–641. Springer Berlin Heidelberg, 1998.

[79] Rob Fergus, Pietro Perona, and Andrew Zisserman, “Object class recognition by unsupervised scale-invariant learning,” in *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, June 2003, vol. 2, pp. II–264–II–271 vol.2.

[80] Josip Sivic and Andrew Zisserman, "Video google: a text retrieval approach to object matching in videos," in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, Oct 2003, pp. 1470–1477 vol.2.

[81] Gabriella Csurka, Christopher R. Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray, "Visual categorization with bags of keypoints," in *In Workshop on Statistical Learning in Computer Vision, ECCV*, 2004, pp. 1–22.

[82] Frederic Jurie and Bill Triggs, "Creating efficient codebooks for visual recognition," in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, Oct 2005, vol. 1, pp. 604–610 Vol. 1.

[83] Eric Nowak, Frédéric Jurie, and Bill Triggs, "Sampling strategies for bag-of-features image classification," in *Computer Vision – ECCV 2006*, Aleš Leonardis, Horst Bischof, and Axel Pinz, Eds., vol. 3954 of *Lecture Notes in Computer Science*, pp. 490–503. Springer Berlin Heidelberg, 2006.

[84] Ming-Ming Cheng, Ziming Zhang, Wen-Yan Lin, and Philip Torr, "Bing: Binarized normed gradients for objectness estimation at 300fps," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, June 2014, pp. 3286–3293.

[85] E. Ergul and N. Arica, "Scene classification using spatial pyramid of latent topics," in *Pattern Recognition (ICPR), 2010 20th International Conference on*, Aug 2010, pp. 3603–3606.

[86] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *CoRR*, vol. abs/1406.4729, 2014.