EGE UNIVERSITY GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCE

(PHD THESIS)

SEMANTIC INTEREST POINT DETECTION

Sinem ASLAN

Supervisor: Prof. Dr. E. Turhan TUNALI

Co-Supervisor: Prof. Dr. Bülent SANKUR

International Computer Department

Presentation Date : 28.09.2016

Bornova-İZMİR
2016

Sinem ASLAN tarafından Doktora tezi olarak sunulan “SEMANTIC INTEREST POINT DETECTION” başlıklı bu çalışma EÜ Lisansüstü Eğitim ve Öğretim Yönetmeliği ile EÜ Fen Bilimleri Enstitüsü Eğitim ve Öğretim Yönergesi’nin ilgili hükümleri uyarınca tarafımızdan değerlendirilerek savunmaya değer bulunmuş ve 28.09.2016 tarihinde yapılan tez savunma sınavında aday oybirliği/oyçokluğu ile başarılı bulunmuştur.

Jüri Üyeleri :		**İmza**
Jüri Başkanı	**:** Prof Dr. E. Turhan TUNALI	.....
Raportör Üye	**:** Prof. Dr. Bahar KARAOĞLAN	.....
Üye	**:** Doç. Dr. Muhammed CİNSDİKİCİ	.....
Üye	**:** Yrd. Doç. Dr. Haldun SARNEL	.....
Üye	**:** Yrd. Doç. Dr. Mustafa ÖZUYSAL	.....

EGE ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ

ETİK KURALLARA UYGUNLUK BEYANI

EÜ Lisansüstü Eğitim ve Öğretim Yönetmeliğinin ilgili hükümleri uyarınca Doktora Tezi olarak sunduğum "SEMANTIC INTEREST POINT DETECTION" başlıklı bu tezin kendi çalışmam olduğunu, sunduğum tüm sonuç, doküman, bilgi ve belgeleri bizzat ve bu tez çalışması kapsamında elde ettiğimi, bu tez çalışmasıyla elde edilmeyen bütün bilgi ve yorumlara atıf yaptığımı ve bunları kaynaklar listesinde usulüne uygun olarak verdiğimi, tez çalışması ve yazımı sırasında patent ve telif haklarını ihlal edici bir davranışımın olmadığını, bu tezin herhangi bir bölümünü bu üniversite veya diğer bir üniversitede başka bir tez çalışması içinde sunmadığımı, bu tezin planlanmasından yazımına kadar bütün safhalarda bilimsel etik kurallarına uygun olarak davrandığımı ve aksinin ortaya çıkması durumunda her türlü yasal sonucu kabul edeceğimi beyan ederim.

28.09.2016

Sinem ASLAN

ÖZET

ANLAMSAL İLGİ NOKTASI SEZİMİ

ASLAN, Sinem

Doktora Tezi, Uluslararası Bilgisayar Anabilim Dalı
Tez Danışmanı: Prof Dr. E. Turhan TUNALI
İkinci Danışmanı: Prof. Dr. Bülent SANKUR
Eylül 2016, 100 sayfa

Bu tez çalışmasında, çeşitli imge anlama uygulamalarında etkin şekilde kullanılabilecek, yenilikçi bir model-güdümlü görsel sözlük oluşturma yöntemi geliştirilmiştir.

Symbolic Patch Dictionary (SymPaD) olarak isimlendirilen bu görsel sözlük, Sözcükler Torbası (Bag of Visual Words) uygulama adımlarını izlemektedir, imge noktaları sıkça ve düzenli aralıklarla ziyaret edilir, bu imge noktaları yörelerine sözlük atomlarına benzerliklerine göre puanlar atanır, imge yörelerinin puanları bir histogramın selelerinde biriktirilir ve böylece imge imzası elde edilir. Önerilen yöntem, literatür çalışmalarından, görsel sözlüğün üretim aşamasında farklılaşmaktadır. SymPaD sözlüğündeki şekil bölütleri nitel imge karakteristiklerini kodlamak üzere matematiksel formülasyon ile üretilir. Geliştirilen yöntem, çeşitli imge anlama uygulamaları için kullanılan denektaşı verikümelerinde etkin performans başarımı sağlamaktadır.

Anahtar sözcükler: İmge betimleyici, imge yapıtaşları, model-güdümlü, görsel sözlük, sözcükler torbası, simgesel betimleme, imge anlama, nesne tanıma, kategori tanıma, imge gerigetirimi

ABSTRACT

SEMANTIC INTEREST POINT DETECTION

ASLAN, Sinem

PhD in International Computer Department
Supervisor: Prof. Dr. E. Turhan TUNALI
Co-Supervisor: Prof. Dr. Bülent SANKUR
September 2016, 100 pages

In this thesis, the problem of constructing a generic model-driven visual dictionary that is adequate for a variety of image understanding applications is investigated. The proposed dictionary based scheme that we call Symbolic Patch Dictionary (SymPaD), follows the steps of Bag of Visual Words (BoVW) paradigm in which, pixels are visited on a dense grid, local image characteristics are extracted in terms of shape similarity scores to the dictionary atoms, the scores are pooled, and finally an image signature is obtained. We differ from BoVW schemes in the literature in the generation of our shape dictionary. These shape patterns are generated by mathematical formulae encoding qualititative image characteristics. Compared with the existing model-driven schemes our method is able to represent images of a variety benchmark datasets of image understanding applications with better discrimination.

Keywords: image descriptor, model-driven, visual dictionary, Bag of Words, symbolic description, image understanding, object recognition, category recognition, image retrieval

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my supervisor Prof. E. Turhan Tunalı for his continuous support during my doctoral studies. I am very grateful to my co-supervisor Prof. Bülent Sankur for his invaluable guidance at every stage of this thesis study. It was a great chance for me to visit BUSIM laboratory in Boğaziçi University during my doctoral studies and study with him. I have learned a lot from him, not merely about research in computer vision and signal processing, but also about literature, mythology, art, classical music and many aspects of life. I sincerely thank to Dr. Ceyhun Burak Akgül who has introduced the initial thesis problem to me and has given valuable advices and suggestions during the whole study. I would like to thank Dr. Erdem Yörük for his valuable comments at our meetings conducted in Boğaziçi University.

I am thankful to the members of my thesis committee, Prof. Bahar Karaoğlan, Assoc. Prof. Muhammed Cinsdikici, Asst. Prof. Haldun Sarnel and Asst. Prof. Mustafa Özuysal. They have provided valuable suggestions for improvement of this dissertation.

I have made wonderful friends who have always stood by me at hard times of my doctoral studies. I can never forget the support of my precious friends Müge Sayıt, Elif Haytaoğlu, Ramin Fouladi, Ahmet Bilgili, Mehmet Yamaç, and George Tzagkarakis during these last three years and I feel so thankful to them. Our days in BUSIM laboratory with Burcu Tepekule, Leda Sarı, Can Altay, Erinç Dikici, İpek Şen, Artun Oyman, and Sezer Ulukaya were unforgettable. We were together at the laboratory in day and night times and mostly at weekends. We have studied a lot, we have talked and laughed a lot, we have shared a lot. I will always miss those days. I would like to thank to my colleagues at EU International Computer Institute, Serkan Ergun, Kaya Oğuz, Can Umut İleri, Gül Boztok Algın, Cihat Çetinkaya, Sercan Demirci and Cemre Candemir, for their support.

I would like to express my deepest gratitude to my parents Sevim and İsmail Aslan. They have always encouraged me to pursue my dreams. Finally, I would like to thank to my dear sister Çiğdem Aslan for her warm support.

TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

TABLE OF CONTENTS (continued)

LIST OF FIGURES

LIST OF FIGURES (continued)

LIST OF FIGURES (continued)

LIST OF TABLES

LIST OF TABLES (continued)

LIST OF ABBREVIATIONS

Abbreviation	Explanation
2D	2-Dimensional
3D	3-Dimensional
ALOI	Amsterdam Library Of Images
BIFs	Basic Image Features
BoVW	Bag-of-Visual Words
BRIEF	Binary Robust Independent Elementary Features
BRISK	Binary Robust Invariant Scalable Keypoints
CNNs	Convolutional Neural Networks
COIL	Columbia Object Image Library
DCT	Discrete Cosine Transform
DtG	Derivative of Gaussian
DWT	Discrete Wavelet Transform
EFB	Equal Frequency Binning
EWB	Equal Width Binning
FREAK	Fast Retina Keypoint
GLOH	Gradient Location And Orientation Histogram
GMM	Gaussian Mixture Model
HOG	Histogram of Oriented Gradients
ICA	Independent Component Analysis
K-SVD	K-Singular Value Decomposition
KL	KMeans - Laplacian

LIST OF ABBREVIATIONS (continued)

Abbreviation	Explanation
LAFs	Local Affine Frames
LCC	Local Coordinate Coding
LDA	Linear Discriminant Analysis
LLC	Local-Constraint Linear Coding
M-CORD	Multi-Colored Region Descriptor
MI	Mutual Information
MMC	Maximum Margin Clustering
MR8	Maximum Response 8
NMF	Nonnegative Matrix Factorization
oBIFs	Oriented BIFs
ODL	Online Dictionary Learning
OMP	Orthogonal Matching Pursuit
ORB	Oriented FAST And Rotated BRIEF
OVA	One-Vs-All
OVO	One-Vs-One
PAM	Partitioning Around Medoids
PCA	Principal Component Analysis
PMT	Piotr'S Computer Vision Matlab Toolbox
RBF	Radial Basis Function
RoI	Region of Interest
SalBayes	Salient Bayes

LIST OF ABBREVIATIONS (continued)

Abbreviation	Explanation
SBE	Sequential Backward Elimination
SFFS	Sequential Floating Forward Selection
SFS	Sequential Forward Selection
SIFT	Scale-Invariant Feature Transform
SNR	Signal Noise Ratio
SPM	Spatial Pyramid Matching
SRC	Sparse Representation Coding
SURF	Speeded Up Robust Features
SVM	Support Vector Machine
SymPaD	Symbolic Patch Dictionary
ZuBuD	Zurich Buildings Dataset

1. INTRODUCTION

Marr defined the goal of human vision as "to know *what* is *where* by looking" (Marr, 1982). Similarly in computer vision, we build machines that have the same ability, the goal is defined in terms of what humans care about, i.e., detection, verification, identification, categorization, etc. Image understanding applications dealing with such problems covers a large spectrum from face, gesture and object recognition to image retrieval, industrial inspection to medical diagnosis, remote sensed imaging to surveillance and security. *Bag-of-Visual Words* (*BoVW*) paradigm yields state-of-the-art performance for such tasks of image understanding. BoVW methods use visual words, such as clustered *Scale-Invariant Feature Transform* (*SIFT*) vectors as mid-level representations of image patches. Therefore, it is important in the BoVW framework to obtain good representative dictionaries. A plethora of visual dictionaries have been generated in the literature according to three paradigms:

- *Dictionaries built from mixtures of the column set of known transform matrices*, such as *Discrete Cosine Transform* (*DCT*) (Rubinstein et al., 2010), *Discrete Wavelet Transform* (*DWT*) (Figueras i Ventura et al., 2006), Gabor filter (Figueras i Ventura et al., 2006), curvelet (Candes et al., 2006), edgelet (Donoho, 1998a), ridgelet (Donoho, 1998b), contourlet (Do and Vetterli, 2005), bandelet (Le Pennec and Mallat, 2005), steerable filters (Freeman and Adelson, 1991), etc. The main advantage of these dictionaries is the viability of their fast implementation. However these dictionaries have limitations, i.e., they can only be successful as their underlying model. For example DCT is good at representing images with homogeneous components, DWT is good at representing point singularities and, edgelets, curvelets, ridgelets, contourlets, and bandelets, are good at representing the line singularities in images.

- *Dictionaries learned from data*. This approach derives dictionaries based on matrix factorization principles under sparsity constraints such as *K-Singular Value Decomposition* (*K-SVD*) (Aharon et al., 2006) and *Online Dictionary Learning* (*ODL*) (Mairal et al., 2010). Another approach is based on sparse or dense sampling images, obtaining local features such as *Histogram of Oriented Gradients* (*HOG*) (Dalal and Triggs, 2005) or SIFT (Lowe, 2004) and building a dictionary via some clustering technique (Csurka et al., 2004; Jurie and Triggs, 2005). These can be grouped under the name of unsupervised dictionary learning techniques. Recent studies have introduced supervised dictionary learning (Wright et al., 2009; Yang et al., 2010; Fulkerson et al.,

2008; Winn et al., 2005; Mairal et al., 2009; Zhang and Li, 2010; Pham and Venkatesh, 2008) for better classification performance where a class-specific discrimination term is added to the learning algorithm. The main advantage of dictionary-learning techniques is that dictionaries can be fine tuned to the underlying dataset as compared to the transform-based approaches. Furthermore, results in the literature indicate that much better performance can be achieved compared to the transform-based ones. The main disadvantage is that unsupervised techniques result in an unstructured dictionary which is costlier to apply compared to the transform-based ones. Supervised techniques are more discriminative than unsupervised ones and better in classification task, yet they still have some drawbacks, specifically: i) (Wright et al., 2009; Yang et al., 2010) may result in very large dictionary sizes, the size growing linearly with number of classes; ii) Supervised pruning of the initial unsupervised learned dictionary in (Fulkerson et al., 2008; Winn et al., 2005) does not make deteriorate the performance but causes computational load; iii) The optimization problem is non-convex and complex as in (Mairal et al., 2009; Zhang and Li, 2010; Pham and Venkatesh, 2008).

- *Dictionaries that are crafted on models of local image appearances*. These are typically models of gray-level image topological features, such as ramps, corners, wedges, bars, crosses, saddles, mesas, valleys, potholes, valleys, depressions, gorges, ridges, flat zones, etc. This technique has been a much less explored path for creating visual dictionaries. Marr's studies in 1980s (Marr, 1976, 1982) can be accepted as the beginning of describing natural images in terms of a geometrical structures set. Inspired from the findings in physiology (Barlow, 1953; Hubel and Wiesel, 1962; Hartline, 1938), he claimed that in order to achieve visual perception for machine vision systems, some primitive shape structures such as edge, bar and blob should be detected on the images firstly. Recently, Griffin et al. (Griffin and Lillholm, 2007; Lillholm and Griffin, 2008; Crosier and Griffin, 2010) have introduced a dictionary construction method, where images are described in terms of a pre-determined dictionary of merely 7 basic qualitative structures, that are *flat*, *dark* and *light bar*, *dark* and *light blob* and *saddle*, called as *Basic Image Features* (*BIFs*). The shape models are defined by a parametric mapping from a jet space to a partitioned orbifold. The details about this technique is presented in Appendix 1. These authors have subsequently enriched their coarse dictionary by replicating BIFs in different orientations, though its performance in object categorization tasks was far from being competitive (Lillholm and Griffin, 2008).

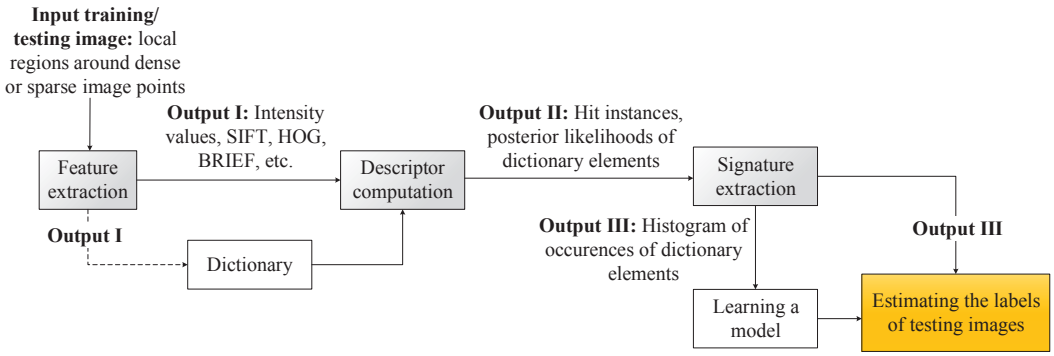


Figure 1.1. Processing steps in the pipeline of a dictionary-based computer vision task (dashed line takes part when dictionary was learned from data)

We believe that model-based dictionary methods has further room for exploration and improvement. The potential for improvement lies in a more detailed quantization of the parameter space of the shape models as well as exploring new representative shape types. The work conducted in this thesis is focused on this exact problem.

Fig. 1.1 shows the three main operations in the pipeline of a dictionary-based computer vision task. The operations are (i) Feature extraction, (ii) Descriptor computation, and (iii) Signature extraction.

Feature extraction. In the first stage, characteristics of the local patches around keypoints can be used. The simplest image feature can be the vector of pixel values or their histogram within a patch. However, raw pixel values are sensitive to position, illumination, noise, geometrical transforms, etc. Thus, image features have been developed in the literature (Mikolajczyk and Schmid, 2005; Li and Allinson, 2008), that, if not totally invariant, mitigate spatial and/or photometric transformations. One can use HOG (Dalal and Triggs, 2005), SIFT (Lowe, 2004), *Gradient Location and Orientation Histogram* (*GLOH*) (Mikolajczyk and Schmid, 2005), *Speeded Up Robust Features* (*SURF*) (Bay et al., 2006) features, on sparse points of interest or densely on a regular grid (Lazebnik et al., 2006). Other possibilities consist of the family of filter kernels, e.g., steerable filters (Freeman and Adelson, 1991), and Gabor filters (Daugman, 1980, 1985). Derivative-based features investigated by (Koenderink and van Doorn, 1987) are some other examples. These have been used successfully in many applications such as image coding (Zhu, 2002), foreground/background segmentation (Martin et al., 2004; Heiler and Schnörr, 2005), moving object detection (Dou and Li, 2014), pose estimation (Dantone et al., 2014)

or image registration (Cen et al., 2004). Recently, binary features, i.e., *Binary Robust Independent Elementary Features* (*BRIEF*) (Calonder et al., 2012), *Oriented FAST and Rotated BRIEF* (*ORB*) (Rublee et al., 2011), *Binary Robust Invariant Scalable Keypoints* (*BRISK*) (Leutenegger et al., 2011), *Fast Retina Keypoint* (*FREAK*) (Alahi et al., 2012), have attracted some attention, due to their computational simplicity, memory-efficiency and their inherent robustness against image variability.

Descriptor computation. Once the local features are extracted, they are encoded in a descriptor (a.k.a. *code vector*), using a collection of such elements (a.k.a. *code words*) of a predetermined visual dictionary (a.k.a. *codebook*). Principal encoding methods that have been used in the literature (Huang et al., 2014) can be grouped under categories of (i) voting-based methods such as *hard-voting* (Csurka et al., 2004) and *soft-voting* (Van Gemert et al., 2008), (ii) reconstruction-based methods such as *sparse coding* (Yang et al., 2009), *Local Coordinate Coding* (*LCC*) (Yu et al., 2009), and *Local-constraint Linear Coding* (*LLC*) (Wang et al., 2010), and (iii) *Fisher coding* methods (Perronnin and Dance, 2007; Perronnin et al., 2010). Fisher coding and reconstruction-based methods outperform voting-based methods (Huang et al., 2014). Among all, Fisher coding is reported as the best one, i.e., since *Gaussian Mixture Model* (*GMM*) provides richer information, it is more robust to unusual, i.e., noisy features. However, Fisher coding gives rise to very high dimensional descriptor vectors. Reconstruction-based methods yield a more exact representation of features than voting-based methods, but computational complexity is higher and they are the least robust ones among all as reported in (Huang et al., 2014).

Signature extraction An image should be represented by a unique vector, called its signature, to be compared with the other images. One way to accomplish this is to combine the descriptors occurrences (hits) into a "bag of features" vector. Essentially this is a *spatial pooling* operation. Spatial pooling provides not only compactness of representation, but also, invariance to transformations such as changes in position and lighting conditions, and robustness to noise and clutter (Boureau et al., 2010b). Sum (or average) and max pooling are the two common ways used for this purpose (Boureau et al., 2010b; Murray and Perronnin, 2014). Discriminability is reduced by sum pooling, since it is influenced strongly by most frequent features that may not be informative enough as in the stop words paradigm in text retrieval (Murray and Perronnin, 2014). Max pooling balances that biased attitude, however, it is not necessarily the best method to be used for every coding scheme. Thus for example it is not good with Fisher coding, but works well with soft-voting and sparse

coding (Murray and Perronnin, 2014). Furthermore, pooling spatially close descriptors as in the cases of *Spatial Pyramid Matching* (*SPM*) (Lazebnik et al., 2006) and *macro-features* (Boureau et al., 2011) has been shown to bring substantial improvements.

1.1 Contributions of the Thesis

In this thesis, the problem of constructing a generic model-driven visual dictionary that performs well for a large variety of image understanding applications is investigated. The proposed dictionary, that we call **Sym**bolic **Pa**tch **D**ictionary (**SymPaD**), follows the steps of BoVW paradigm in that, pixels are visited on a dense grid, local image characteristics are extracted in terms of shape similarity scores to the dictionary atoms, the scores are pooled, and finally an image signature is obtained. We differ from BoVW schemes in the literature in the generation of our shape dictionary. These shape patterns are generated by mathematical formulae encoding qualitative image characteristics (Marr, 1976, 1982; Saund, 1990; Horaud et al., 1990; Lowe, 1984; Griffin and Lillholm, 2007; Lillholm and Griffin, 2008; Griffin et al., 2009). We can summarize our contributions as follows.

- With the proposed SymPaD dictionary scheme, we try to minimize the dependence on the data. We demonstrate that with such a generic visual dictionary, we perform on a par, and in some instances outperform conventional data-driven approaches employing BoVW scheme.

  Well-established model-based dictionaries have some advantages over the dictionaries that are learned from the data. Extreme dependence on the data might be problematic or inefficient in the following situations:

  i. The training data to learn the dictionary from may not be abundant;

  ii. Dictionary learning is guided by observations from data, thus randomly sampled image patches that are input to the algorithm would naturally favour the most frequent appearance of primitives. Thus the algorithm, e.g. K-Means clustering, would overfit to the most frequent patterns in the images, and become probably less discriminative (Jurie and Triggs, 2005);

- The proposed SymPaD model-driven dictionary scheme proves to be superior to the current state-of-the-art model-driven methods. The most current model-driven approach proposed by (Griffin and Lillholm, 2007; Lillholm and Griffin, 2008; Griffin et al., 2009) makes model definitions by re-parametrizing

the jet space, i.e., filter response space, of *Derivative of Gaussian* (*DtG*) filters that are in 1^{st}, 2^{nd}, and $1^{st}\&2^{nd}$ order and this leads to a set of features in seven types (details are presented in Appendix 1). For a wider range of feature types, new parametric mappings for higher orders of DtG filters should be considered, however to the best of our knowledge they did not elaborate any further on how to define these features from filter responses. Differently from existing model-driven methods, our scheme can incorporate a very rich set of shape primitives in the visual dictionary thanks to its parametric generative function. More importantly, the parametric representation allows sampling of the parameter space in order to produce shape varieties.

We experimentally demonstrate that the designed shape dictionary is by far the best among the existing model-driven schemes in object and category recognition and image retrieval applications.

1.2 Thesis Outline

This thesis is organized as follows:

In Chapter 2, we review studies in the literature on natural image statistics and primitive structures of images that can potentially inspire model driven patch shape dictionaries. In Section 2.2, we address problems in state-of-the-art methods in visual dictionary construction adopting model-driven approach.

In Chapter 3, we describe our model-based shape dictionary. In Section 3.1 we introduce the chosen shape models forming the core dictionary. In Section 3.2, we present the parameterizing of the dictionary, i.e., the procedures of generating variations based on rotation, scaling and shifts and we give an overview of the shape dictionary in Section 3.3.

In Chapter 4, we provide a mutual information based pruning procedure for a more compact form of the dictionary and we make analysis of the pruned dictionary.

In Chapter 5, we introduce the main components that we use in the pipeline of procedures from feature extraction to image classification/recognition. In Section 5.1, we introduce the binary feature, namely BRIEF, that we compute on shape dictionary patterns and input image patches. In Sections 5.2 and 5.3, we describe the voting technique that we use to compute the patch descriptors on the test image points and the pooling technique that we use to obtain a unique signature for each

image, respectively. In Section 5.4, we briefly explain the final stage statistical classifier that we use to recognize class/category labels.

In Chapter 6, we provide extensive experimental results. We briefly introduce the benchmark datasets that we use in the evaluation of the techniques. In Section 6.1, we explain the preprocessing steps and experimental settings accepted in the literature for the datasets accordingly. In Section 6.2, we make a comparative analysis with the widely-used dictionary learning method of K-SVD. In Section 6.3, we assess the performance of our method and current state-of-the-art model-driven methods and demonstrate the superiority of SymPaD. In Section 6.4, we additionally report for comparative purposes the performance of recent data-driven methods from the literature and make a comparative analysis.

In Chapter 7, we conclude and present a discussion about future research directions.

2. LITERATURE SURVEY

In the first part of this chapter, we briefly review qualitative features of images that can potentially inspire model driven patch shape dictionaries. We present state-of-the-art in visual dictionary construction adopting model-driven approach in the second part.

2.1 Modelling Local Structures of Natural Images

Analysis of natural images, i.e., investigating the meaningful primitive representatives and their statistics, is crucial to design proper shape models to generate a useful visual dictionary. Many researchers have been attracted by this subject with the objective of finding evidences at an early stage of visual perception, so higher-level processing, e.g., recognition, compression, etc., could be made more computationally feasible.

First studies on types of perceived visual primitives were carried out in physiology. (Hartline, 1938) found out that there are certain cells in mammalian retina that respond to light, dark stimulus and transitions from light to dark named them as on-, off-, and on-off ganglion cells. In the early 1950s, (Barlow, 1953) and (Kuffler, 1953) discovered the phenomenon of lateral inhibition on mammalian retina that induces contrast increase and enhances the visual system's edge detection capability. (Barlow, 1953) also determined that the on-off ganglion cells of frog retina is highly sensitive to small moving blobs and named these cells as *bug detectors*. In the early 1960s, (Hubel and Wiesel, 1962) designated orientation-selective cells at visual cortex of cats and determined that these cells are more sensitive to rectangular bars of light rather than circular spots of light which are then named as *bar detectors*.

Marr was one of the earliest researchers who studied visual perception from a computational perspective, i.e., for machine vision systems (Marr, 1976, 1982). He established a representational framework for visual perception, and claimed that at the first stage of this framework, information is made explicit via symbolic representations of tokens that are intensity discontinuities in various formations such as edges, bars, and blobs. For the choice of these tokens, i.e., *primitive structures*, he was inspired from the studies in the physiology of vision, i.e., (Barlow, 1953; Hubel and Wiesel, 1962; Hartline, 1938), that showed existence of certain cells in mammalian retina and visual cortex that respond to only some particular type of visual stimuli, such as edge, blob, and oriented bar. He also suggested using various spatial groupings of initial tokens for a more meaningful description, named as *place token*,

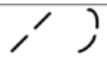
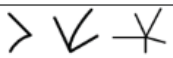
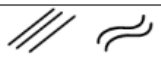
2D Relation	Example	2D Relation	Example
Collinearity and curvilinearity		Crossing of continuous curves	
Terminations at a common point		Parallelism	
Terminations at a continuous curve/line		Lines converging to a common point	

Figure 2.1. Examples to relations between image primitives on the 2D image that are invariant across viewpoint changes, thus unlikely to be emerged by accident. Image credits to (Lowe, 1984).

i.e., each group of tokens is defined as an individual token (Marr, 1976). Marr argued that such representations that include *primitive tokens* and *place tokens*, called as the *primal sketch*, should be sufficient to represent the original image (Marr, 1982, 1976). In the same decade, Julesz investigated the problem for pre-attentive discrimination of texture images, and asserted that texture images are characterized by repetition of a few atomic elements, such as bars, edges and terminations (or end points), that he named as *textons* (Julesz, 1981). (Tenenbaum and Witkin, 1983) and (Lowe, 1984) developed the idea later, by suggesting the non-accidentalness properties of successful grouping of primitive structures for better image representation. (Lowe, 1984) demonstrated that certain relations such as collinearity, curvilinearity, co-termination, crossings, parallelism and symmetry between initial primitives on the *2-Dimensional* (*2D*) image are invariant across viewpoint changes. He concluded that these relations are unlikely to be emerged by accident, so these grouping structures are thought to be more informative about the object. Some examples of these relations are illustrated in Figure 2.1.

Another research stream was through investigating the statistics of intensity discontinuities in natural images. (Field, 1994) demonstrated that the histograms of Gabor filter responses on natural images have high kurtosis. (Geman and Koloydenko, 1999) analyzed a large set of 3×3 patches from two natural image datasets by a modified order statistics and found that the non-background patches are in the appearance of edge shapes with a high probability. (Lee et al., 2001) analyzed the probability distribution of a large set of 3×3 high-contrast patches that were sampled from natural images and found that patch space is extremely sparse with patches located around the manifold of edges of different orientations and positions. The common outcome of these studies (Field, 1994; Geman and Koloydenko, 1999; Lee

et al., 2001), i.e., the observed high kurtosis in image statistics, demonstrates that it should be possible to represent natural image primitives with a limited number of visual elements.

These initial studies have inspired many researchers till today and their suggestions have been used in a variety of vision applications successfully. Extracting primitive structures of edge features, and describing their attributes and spatial relations, (Vilnrotter et al., 1981) obtained good performance results for texture recognition. (Saund, 1990) proposed to add scale dimension to Marr's Primal Sketch for shape recognition applications. (Horaud et al., 1990) suggested an intermediate-level description accomplished by an exhaustive search to detect geometric structures, and groupings of them, such as linear and curved contours, junctions, and local symmetries like parallels, ribbons, and parallelograms. (Horaud et al., 1990) that such a representation is useful for object recognition and stereo image matching in (Horaud and Skordas, 1989). By using primal sketch priors such as edges, ridges, corners, T-junctions and terminations, (Sun et al., 2003) obtained encouraging results in enhancing the quality of the hallucinated high-resolution generic images. (Guo et al., 2007) used a visual dictionary including primitive shapes of blobs, terminations, edges, ridges, multi ridges, corners, junctions, crosses to model structural components of natural images and demonstrated that such a representation is useful for lossy image coding. In a more recent study, (Crosier and Griffin, 2010) proposed to represent images in terms of a set of image primitives, that are flat, ramp, dark/light line, dark/light circle, and saddle, in a bag of words scheme, and demonstrated that the state-of-the-art performance is obtained in texture classification (Crosier and Griffin, 2010) when their occurrences in scale-space is considered. These successful implementations adopting model-driven schemes demonstrate that model-driven schemes deserves further investigation.

The last decade has seen a plethora of methods to obtain local image representations based on the data-driven methods. These methods that learn local structures of images automatically range from *Principal Component Analysis* (*PCA*) to clustering (or vector quantization) (Nasrabadi and King, 1988), from *Nonnegative Matrix Factorization* (*NMF*) (Lee and Seung, 1999) to *Independent Component Analysis* (*ICA*) (Hyvärinen et al., 2004). Recently, dictionary learning techniques based on constrained optimization have been popular (Aharon et al., 2006; Mairal et al., 2010). These learned dictionaries proved surprisingly effective in representing images for classification, detection and recognition tasks. In fact, most of these methods can be conceived under a single mathematical formalism, that of matrix factorization (Mairal et al., 2014). Let $X = [x_1, \ldots, x_n] \in \mathbb{R}^{p \times n}$ be the training set of

p-dimensional images or image patches, $D = [d_1, \ldots, d_r]$ the matrix of dictionary elements represented by $d_i \in \mathbb{R}^p$ and $A = [\alpha_1, \ldots, \alpha_n]$ is the matrix of decomposition coefficients represented by $\alpha_i \in \mathbb{R}^r$. The matrix is factorized into the product of a dictionary matrix and the mixture coefficients as $X \approx DA$ such that one obtains a good approximation of $x_i \approx D\alpha_i$, $D\alpha_i = \Sigma_{j=1}^{r} \alpha_i[j] d_j$ (Mairal et al., 2014). The various methods proposed differ in the regularizer terms, that is constraints imposed, such as sparsity of the mixing coefficients A, and low-rank, orthogonality, or the existence of structure in D.

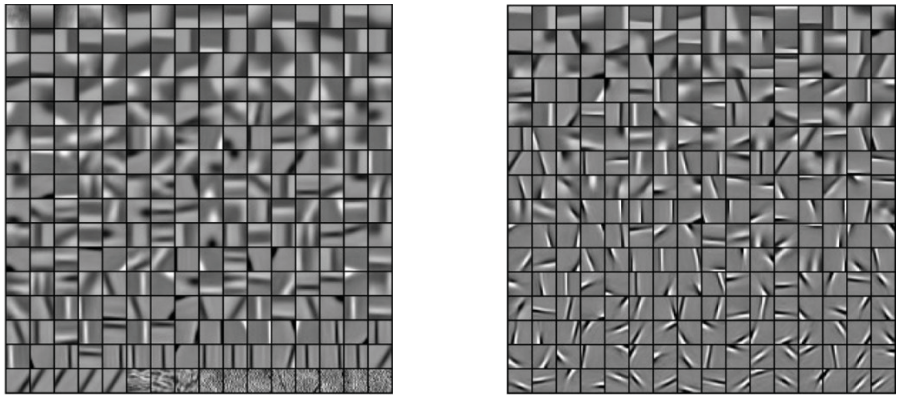
We present visualizations from (Mairal et al., 2014) in Fig. 2.2, that are dictionaries learned from a set of mean normalized natural image patches by two unsupervised algorithms, i.e., K-Means and ODL (Mairal et al., 2010). Both of the presented dictionaries that consists of $r = 256$ atoms are learned from $n = 4 \times 10^5$ natural image patches with size $p = 16 \times 16$. Figure 2.2.(a) shows the dictionary obtained by K-Means clustering. The goal of K-Means clustering is finding r clusters of input patches by minimizing the objective function in Eq. 2.1 iteratively by alternating minimization between A and D, where each $\alpha_i \in \mathbb{R}^r$ in $A = [\alpha_1, \ldots, \alpha_n]$ is a binary vector that sum to one. We observe in Fig. 2.1.(a) that, the learned visual dictionary mostly consists of low-frequency elements, i.e., ramps and bars in various orientations and positions.

$$\min_{D,A} \frac{1}{2n} \|X - DA\|_F^2 \text{ s.t. } \forall i, \ \sum_{j=1}^{r} \alpha_i[j] = 1 \tag{2.1}$$

The dictionary in Fig. 2.2.(b) is obtained by the formulation in Eq. 2.2 where $\Psi(A) = \sum_{i=1}^{n} \psi(\alpha_i)$, ψ is the l_1-norm and the regularization parameter λ is set to 0.1. The dictionary learning algorithm of ODL (Mairal et al., 2010) is used to obtain the visual dictionary. We observe in Figure 2.2.(b) that the learned dictionary consists of both low frequency elements and high-frequency, Gabor-like patterns with different orientations and positions.

$$\min_{D,A} \frac{1}{2} \|X - DA\|_F^2 + \lambda \Psi(A) \tag{2.2}$$

These visualized primitive shape structures that are learned from natural images coincide with the choice of primitive shapes used by model driven approaches in the literature (Marr, 1976, 1982; Vilnrotter et al., 1981; Saund, 1990; Horaud et al., 1990; Horaud and Skordas, 1989; Sun et al., 2003; Guo et al., 2007; Crosier and



(a) Visual dictionary learned by the K-Means algorithm.

(b) Visual dictionary learned by the Online Dictionary Learning (ODL) algorithm.

Figure 2.2. Visualization of two dictionaries with $r = 256$ atoms computed on $n = 400000$ mean-normalized image patches of size $p = 16 \times 16$. Image credits to (Mairal et al., 2014).

Griffin, 2010).

2.1.1 State-of-the-art in model-driven visual dictionary

The most current model-driven dictionary construction scheme following the steps of conventional BoVWparadigm, i.e., representing the images in terms of a set of primitive shape models, is developed by Griffin et al. (Griffin and Lillholm, 2007; Lillholm and Griffin, 2008; Crosier and Griffin, 2010). The designed shape dictionary that is named as *Basic Image Features* (BIFs) is applied on a wide variety of computer vision problems such as object categorization (Lillholm and Griffin, 2008), texture classification (Crosier and Griffin, 2010), quartz sand grains classification for forensic analysis (Newell et al., 2012), character recognition on natural images (Newell and Griffin, 2011), writer identification (Newell and Griffin, 2014), and biomedical image analysis (Jaccard et al., 2014, 2015).

We provide further details about the method in Appendix 1. Briefly, shape models are defined by partitioning the re-parametrized response space of six DtG filters, which corresponds to an orbifold, into seven regions (see Algorithm 1 in Appendix 1). Each partition on this orbifold corresponds to one of seven qualitative image structures, namely, *flat*, *ramp*, *dark* and *light line*, *dark* and *light circle* and *saddle*.

The classification performance obtained by representing texture images as histograms over the mentioned seven qualitative structures was not good enough (i.e. 65% on CuReT dataset) compared to the state-of-the art, since such a dictionary was

too coarse to represent images discriminatively (Crosier and Griffin, 2010). Thus, the authors proposed to use scale templates of BIFs named as *BIF-columns* that is a stack of BIFs computed on the same spatial location among four different scales (see Figure A.3 in Appendix 1). In this way, by excluding the flat structure, they increased the number of histogram bins, i.e., the image signature, to 1296 by considering all possible combinations of remaining six structures ($6^4 = 1296$) and probing the images in four scales. With BIF-columns, they obtain state of the art performance (98.5%) for KTH-TIPs, a challenging texture dataset (Crosier and Griffin, 2010).

For the object categorization problem, they used *oriented BIFs* (*oBIFs*) (Lillholm and Griffin, 2008). The 7 feature types are increased to 23 by quantizing orientations of ramp shape into 8 levels and line and saddle shapes into 4 levels. Since a histogram with 23 bins is still not enough to discriminatively describe an image, they used oBIF-columns that are computed in scale-space and selected 1000 most informative ones (by Mutual Information) to describe images in (Lillholm and Griffin, 2008). The performance results were not competitive with state-of-the-art methods on Pascal VOC 2007 dataset, which is probably because of the fact that the dictionary was not enriched sufficiently by adequate variety of main shape types. The same authors also stressed the same point in their paper (Griffin and Lillholm, 2007) and mentioned that using filters up to 4^{th} or 5^{th} order would provide a more enriched description of local structures, however, to the best of our knowledge, they did not elaborate any further on how to define these features from filter responses.

We believe that the fundamental mathematics needs to be changed for a simpler solution. Thus, in this thesis we move away from parametrization of higher order filter response spaces, and towards a more arithmetical implementation in order to generate a wider variety of visual primitive shape categories.

3. SHAPE DICTIONARY

In this chapter, we present our model-based shape dictionary. We first introduce the chosen shape models forming the core dictionary, then parameterize this dictionary and generate variations based on rotation, scaling and shifts.

3.1 Shape Models

Based on our knowledge of both visual primitives and their groupings employed by model-driven approaches and the local shapes in the data-driven, i.e., learned dictionaries, we decided to define a few essential base shape models for the shapes of local image appearances to start constructing a shape dictionary. These base shapes, also called as the core shapes, consist of *flat*, *ramp*, *valley* (dark line), *ridge* (light line), *basin* (dark circular blob), *summit* (light circular blob), *elongated basin* and *summit*, *termination*, *saddle*, *corner*, several kinds of *junctions*, *cross*, *curves* (like L-junction) and *Gabor-like* shapes. Definitions and notations used throughout in this chapter are given in Table 3.1.

We first introduce the core shape models as the first step of the dictionary atoms. These models or patterns are intended to represent the most basic local image appearances. Each pattern will later spawn a number of other atoms under geometrical transformations of rotations, stretching, translations and contractions. Furthermore, these shape models are combined nonlinearly to obtain a richer dictionary. Accordingly we have three groups of models:

i. *Group I models*, (Table 3.2), consist of linear, quadratic and cubic polynomials, as well as exponential and transcendental functions, all within the argument of a sigmoid. The sigmoid transformation provides us with one extra degree of freedom in its rate parameter α, "$\frac{1}{1+e^{-\alpha x}}$".

ii. *Group II models* (Table 3.3) are compounded models obtained by various nonlinear combinations of Group I models, that were inspired from perceptual groupings in the literature (Marr, 1976; Tenenbaum and Witkin, 1983; Lowe, 1984; Saund, 1990; Horaud et al., 1990; Guo et al., 2007).

iii. *Group III models* (Table 3.4) generate a subset of *Maximum Response 8* (*MR8*) (Varma and Zisserman, 2005) and Gabor filter (Daugman, 1985) sets to complement our shape library. We incorporated these transform-based patterns

Table 3.1. Definitions and notations used throughout Chapter 3.

Symbol	Definition
θ	Rotation angle
α	Transition rate, i.e., steepness of gray level transitions on the hillsides of shapes
ψ	Shape compounding angle
ρ	Eccentricity of the elongated basin/dome
p	Patch size
$f_i(\cdot)$	Shape primitive function
$F_i(\cdot)$	Primitive after being subjected to a sigmoidal operation, $f_i(\cdot) \overrightarrow{sigmoid} F_i(\cdot)$
λ	Eccentricity of the Gaussian mask used to generate Gabor patterns
σ	Standard deviation to set the scale of MR8 and Gabor patterns
ξ	Wavelength of the sinusoidal modulation of Gabor patterns
(x', y')	Translated coordinates by the vector $u = (u_x, u_y)$, i.e., $x' = x + u_x$, $y' = y + u_y$
u	Translation vector, $u = (u_x, u_y)$
v	Translation vector, $v = (v_x, v_y)$ that is used to create termination shapes of MR8 and Gabor patterns
$P1, P2$	Parent models that take part in the compounding operation, e.g., $P1 = 1$ and $P2 = 3$ corresponds to shape models of $F_1(\cdot)$ and $F_3(\cdot)$ in Table 3.2
$\alpha_{Pi}^1, \ldots, \alpha_{Pi}^{q_\alpha}$	Quantized transition rate values for the parent shape model Pi
$q_\theta, q_\alpha, q_\psi$	Number of quanta used in quantization of the parameters θ, α, and ϕ
$\{\theta_1, \ldots, \theta_{q_\theta}\}$	Quantized rotation angle values of a shape model
$\{\alpha_1, \ldots, \alpha_{q_\alpha}\}$	Quantized transition rate values of a shape model
$\{\psi_1, \ldots, \psi_{q_\psi}\}$	Quantized compound angle values of a shape model
D	Number of atoms in a dictionary
R	Number of atoms in a pruned shape dictionary
$D_\gamma^{(\text{Group } i)}$	Number of atoms in the dictionary generated by evaluating models in Group i by the values of a subset of (or included in) whole parameters′ set whatever applicable, i.e., $\gamma \subseteq \{\theta, \psi, \alpha, \sigma, \lambda, \xi, u, v\}$

since they were well-established shapes in the literature.

Notice that this grouping is done simply for convenience of the discourse, but otherwise our final shape dictionary consists of the union of these three sets.

3.1.1 Group I Models

A gray level image can be thought of as a landscape I(x,y) with (x,y) as the spatial dimensions and luminance as the third dimension, similar to a physical landscape (Morgan, 2011). Then, for example, uniform luminance regions correspond to flat surfaces or planes; negative and positive bar-like shapes to valleys or ridges; edges to ramps, scarps or hillsides, etc. In this context, ramps, represent rapid intensity transitions and correspond to edges parametrized by orientation and intensity transition rate; similarly valleys and ridges, correspond respectively, to dark and light lines vis-à-vis their background, characterized by their orientation and intensity transition rate on hillsides. Continuing this line of analogy, one can conceive basins and mesa (or tumulus), as representing circular, dark and light, respectively, blobs, and parametrized by the intensity transition rate from background to foreground. Finally elongated versions of basin and mesa, as representing elliptical dark and light blobs; the latter being parametrized by their orientation, intensity transition rate and eccentricity. Other more complex structures potentially corresponding to more complex local image structure are given in Table 3.2.

New atoms can be built from the collection of basic shape atoms by varying one or more of the *rotation*, *eccentricity*, *translation* and *transition rate* parameters, whenever applicable. By varying these parameters, oriented or shifted versions of the models, or models differing in steepness of transition and/or of eccentricity of the patterns can be created. One must, however, sample the parameter space, that is quantize the parameter values, judiciously so that shape models become sufficiently distinctive and informative. Classification performance results indicate that recognition rates have different sensitivities to the sampling interval of different parameters. For example, the fineness of rotation intervals affects the performance much more as compared to the sampling step size of the transition rate α.

The rotated versions of these models are obtained via the coordinate transformation $\begin{pmatrix} x_\theta \\ y_\theta \end{pmatrix} = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$, where θ is the *rotation angle*. Notice that θ is not applicable to rotationally symmetric shapes such as flat (f_1), basin (f_4) and mesa (f_5). Shifting constitutes a nonlinear operation, i.e., does not satisfy the additivity and homogeneity properties, and we generate new shape varieties by shifting cen-

Table 3.2. Generator functions for Group I shape patterns. (x_θ, y_θ) denotes the rotated coordinates of (x, y) with angle θ, (x', y') denotes the translated coordinates by the amount $u = (u_x, u_y)$, u_x and u_y, i.e., $x' = x + u_x$, $y' = y + u_y$ and $(u_x, u_y) = (0, 0)$ for the non-shifted shapes, $f(x, y, u; \theta)$ denotes the rotated and shifted versions, in that order, of $f(x, y)$, ρ controls the eccentricity of the elongated basin/dome, we use $\rho = 2.4$.

Descriptive Name	**Shape function**	**Appearance**
Flat	$f_0(x, y) = c$	
Ramp	$f_1(x, y, u; \theta) = x'_\theta + y'_\theta$	
Valley	$f_2(x, y, u; \theta) = (x'_\theta + y'_\theta)^2$	
Ridge	$f_3(x, y, u; \theta) = -(x'_\theta + y'_\theta)^2$	
Basin	$f_4(x, y, u; \theta) = x'^2_\theta + y'^2_\theta$	
Mesa	$f_5(x, y, u; \theta) = -(x'^2_\theta + y'^2_\theta)$	
Elongated Basin	$f_6(x, y, u; \theta) = (x'^2_\theta + y'^2_\theta)/\rho$	
Elongated Mesa	$f_7(x, y, u; \theta) = -(x'^2_\theta + y'^2_\theta)/\rho$	
-	$f_8(x, y, u; \theta) = x'_\theta \times y'^2_\theta$	
-	$f_9(x, y, u; \theta) = x'_\theta \times y'^2_\theta + x'^2_\theta \times y'_\theta$	
-	$f_{10}(x, y, u; \theta) = (x'_\theta + y'_\theta)^3$	
-	$f_{11}(x, y, u; \theta) = x'^3_\theta + y'^3_\theta$	
-	$f_{12}(x, y, u; \theta) = \exp(x'_\theta) \times y'_\theta$	
-	$f_{13}(x, y, u; \theta) = x'_\theta \times \cos\frac{y'_\theta}{2}$	
-	$f_{14}(x, y, u; \theta) = \cos\frac{x'_\theta + y'_\theta}{2}$	

tral shape patterns, i.e., $(u_x, u_y) = (0, 0)$, by some translation vector $u = (u_x, u_y)$, $u_x > 0$ or $u_y > 0$, so the parts of the shape moving out of the patch window is cropped out, while the uncovered regions, are filled with the continuation of the shape that was out of view before the translation. The steepness of gray level transitions on the hillsides of ramps, valleys and basins is controlled by α, the *transition rate* parameter, of the sigmoidal function as in Eq. 3.1 where $F : \mathbb{R} \to [0, 1]$ in Table 3.2. Finally, the parameter $\rho > 1$ controls the *eccentricity* of the elongated basin/mesa varieties.

$$F(x, y, u, \theta, \alpha) = \frac{1}{1 + e^{-\alpha f(x,y,u;\theta)}} \tag{3.1}$$

Notice that in Table 3.2, we have denoted the shape primitive function as $f_i(\cdot)$, while the actual shape model used is $F_i(\cdot)$, that is the primitive after being subjected

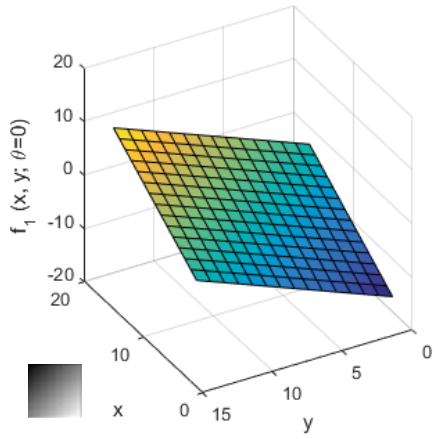
to a sigmoidal operation $f_i(\cdot) \xrightarrow{sigmoid} F_i(\cdot)$ with steepness parameter α. Each $F_i(\cdot)$ takes values in the range $[0, 1]$ and with support of $[-\frac{p-1}{2}, \frac{p-1}{2}] \times [-\frac{p-1}{2}, \frac{p-1}{2}]$, p is odd. For illustration, we show in Fig. 3.1, the 3D shape surfaces and corresponding gray-level pattern (lower left corner) for the ramp primitive function $f_1(x, y, u; \theta = 0^o)$, and two instances of the actual ramp model, $F_1(x, y, u = 0; \theta = 0^o, \alpha = 0.4)$ and $F_1(x, y, u = 0; \theta = 0^o, \alpha = 1.2)$.

One could also consider using as primitive shape models orthogonal polynomials such as Chebyshev, Hermite, Krawtchouk, etc. (Koekoek and Swarttouw, 1996) or higher order polynomial functions. We argue that within patch sizes, e.g., 15×15, sufficient shape diversity is produced by the low-order polynomials, in addition to exponential and transcendental functions as in Table 3.2. Higher-order terms of the orthogonal polynomials tend to have rapid oscillations and polynomial powers beyond three give rise to rapid amplitude excursions, and both behaviours are not commonly encountered in images and/or cannot be accommodated within patch sizes used.

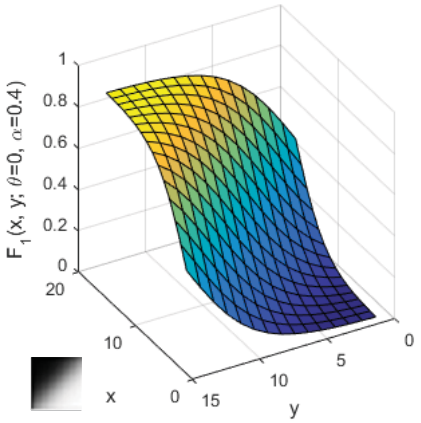
3.1.2 Group II Models

In order to enrich the shape dictionary, we investigated various linear combinations of Group I shapes. For example, we first collected a training set of images and their patches, and represented each patch in terms of Group I models using *Sparse Representation Coding* (*SRC*) (Mairal et al., 2014). Then, we clustered their representation coefficients, and finally obtained new shape models learnt from data in terms of sparse linear combinations of Group I shapes. We observed that the addition to the dictionary of these partly data-learnt shapes did not improve the classification performance significantly.

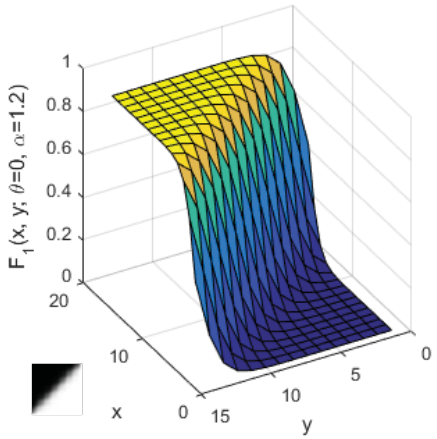
We then conjectured that pairwise nonlinear combinations of the basic shapes would create new discriminative patterns. One way to combine shapes would be to pixel to pixel kernelize them, for example by using linear, polynomial or *Radial Basis Function* (*RBF*) kernel, etc. However, such operations might not always result with meaningful shapes for image representation. In fact, discussions in the literature on the role of structure in vision indicate that groups of primitive structures that yield meaningful image representations are unlikely to emerge by accident; instead they are formed according to certain grouping relations that are invariant across viewpoint changes (Tenenbaum and Witkin, 1983; Lowe, 1984). Such well-known shapes are corners, a variety of junctions, crosses, parallel valleys/ridges, saddles (Marr, 1976; Tenenbaum and Witkin, 1983; Lowe, 1984; Saund, 1990; Horaud et



(a) The surface is generated by the ramp primitive function $f_1(\cdot)$ computed for $\theta = 0^o$ and $u = (0, 0)$.



(b) Sigmoidal transformed version of the surface in (a), i.e., under the $F_1(\cdot)$ function computed at the transition rate $\alpha = 0.4$.



(c) Sigmoidal transformed version of the surface in (a), i.e., under the $F_1(\cdot)$ function computed at the transition rate $\alpha = 1.2$.

Figure 3.1. 3D surface illustrations of the ramp model and its gray-level appearance at the lower left corner.

Table 3.3. Group II models. Note that the parent shapes, e.g., F_1 and F_2, are first scaled, or rotated, or shifted, and then compounded.

Descriptive Name	**Shape function**	**Parents**	**Appear.**
Saddle	$F_{15}(x,y,u;\theta,\psi,\alpha) = \min[\mathbf{max}(F_1(x,y,u;\theta,\alpha), F_1(x,y,u;\theta+\psi,\alpha)), 1-\mathbf{min}(F_1(x,y,u;\theta,\alpha), F_1(x,y,u;\theta+\psi,\alpha))]$		
Dark corner	$F_{16}(x,y,u;\theta,\psi,\alpha) = 1-\mathbf{min}(F_1(x,y,u;\theta,\alpha), F_1(x,y,u;\theta+\psi,\alpha))$		
Light corner	$F_{17}(x,y,u;\theta,\psi,\alpha) = \mathbf{min}(F_1(x,y,u;\theta,\alpha), F_1(x,y,u;\theta+\psi,\alpha))$		
Dark junction	$F_{18}(x,y,u;\theta,\psi,\alpha) = \mathbf{min}(F_1(x,y,u;\theta,\alpha), F_2(x,y,u;\theta+\psi,\alpha))$		
Light junction	$F_{19}(x,y,u;\theta,\psi,\alpha) = 1-\mathbf{min}(F_1(x,y,u;\theta,\alpha), F_2(x,y,u;\theta+\psi,\alpha))$		
Dark cross	$F_{20}(x,y,u;\theta,\psi,\alpha) = \mathbf{min}(F_2(x,y,u;\theta,\alpha), F_2(x,y,u;\theta+\psi,\alpha))$		
Light cross	$F_{21}(x,y,u;\theta,\psi,\alpha) = 1-\mathbf{min}(F_2(x,y,u;\theta,\alpha), F_2(x,y,u;\theta+\psi,\alpha))$		
Dark termination	$F_{22}(x,y,u;\theta,\psi,\alpha) = 1-\mathbf{min}(F_1(x,y,u;\theta,\alpha), F_3(x,y,u;\theta+\psi,\alpha))$		
Light termination	$F_{23}(x,y,u;\theta,\psi,\alpha) = \mathbf{min}(F_1(x,y,u;\theta,\alpha), F_3(x,y,u;\theta+\psi,\alpha))$		
Dark T-Junction	$F_{24}(x,y,u;\theta,\psi,\alpha) = \mathbf{min}(F_2(x,y,u;\theta,\alpha), F_{22}(x,y,u;\theta+\psi,\alpha))$		
Light T-Junction	$F_{25}(x,y,u;\theta,\psi,\alpha) = \mathbf{min}(F_3(x,y,u;\theta,\alpha), F_{23}(x,y,u;\theta+\psi,\alpha))$		
Light curve	$F_{26}(x,y,u;\theta,\psi,\alpha) = \min[1-\mathbf{max}(F_4(x,y,u;\alpha), F_5(x,y,u;\alpha)), 1-\mathbf{min}(F_1(x,y,u;\theta,\alpha), F_1(x,y,u;\theta+\psi,\alpha))]$		
Dark curve	$F_{27}(x,y,u;\theta,\psi,\alpha) = 1-F_{26}(x,y,u;\theta,\psi,\alpha)$		

al., 1990; Guo et al., 2007; Crosier and Griffin, 2010). These shape groupings can be obtained by combining the core shapes nonlinearly, i.e., via min-max addition operations as in Table 3.3. One can conjecture that these models correspond more closely to patterns in real scenes, i.e., one pattern ends and abruptly another pattern starts, or that they are blended in such a way the lighter (viz. the darker) component dominates.

In the min-max operation, we have one new parameter, the relative angle, called the shape compounding angle or simply the *compounding angle*, ψ, between the two parent models, e.g., F_i, F_j, $i \neq j$. The compounded shapes are also subject to rotations by θ angles, much as in the 1^{st} group models. While nonlinearly com-

pounding, we have chosen shape models from Table 3.2 having similar α parameter values. For example, suppose that for some chosen, q_α, let the values of transition parameter be $(\alpha_{P1}^1, \ldots, \alpha_{P1}^{q_\alpha})$ for parent $P1$ and $(\alpha_{P2}^1, \ldots, \alpha_{P2}^{q_\alpha})$ are for parent $P2$. The values are sorted in ascending order as $\alpha_{Pi}^{(1)} < \ldots < \alpha_{Pi}^{(q_\alpha)}$, $i = 1, 2$, and, transition parameters with the same ordered rank are paired, e.g., $\alpha_{P1}^{(1)}$ with $\alpha_{P2}^{(1)}$, $\alpha_{P1}^{(2)}$ with $\alpha_{P2}^{(2)}$, etc. Note that this matching of scales is needed since parent shapes F_i,F_j must have the same scale as argued in (Saund, 1990). This reduces the combinations to be compounded by the cardinality of transition rate quanta set; anyway, the performance was not too sensitive to the transition rate parameter.

3.1.3 Group III Models

Group III shapes consist of models that are filter-bank functions commonly used in the literature. We used first and second Gaussian derivative patterns in MR8 filter set (Varma and Zisserman, 2005) and also high frequency Gabor patterns (Daugman, 1985).

We used the open source code in (Varma and Zisserman, 2007) to create MR8 filters. MR8 set contains edge and bar filters at 6 orientations and 3 scales of $(\sigma_x, \sigma_y) = \{(1, 3), (2, 6), (4, 12)\}$ for the block size of 49 pixels (Varma and Zisserman, 2007). We similarly employed 6 orientations, but just 2 scales of $(\sigma_x, \sigma_y) = \{(1, 3), (2, 6)\}$ that corresponds to 24 patterns which are more fitting for our patch size of $p = 15$ pixels. We created the Gabor patterns in two phases by employing Hilbert transform using the *Piotr's Computer Vision Matlab Toolbox* (*PMT*) (Dollár, 2014). The Gabor parameters are the standard deviation σ to set the scale, λ to set the eccentricity of the Gaussian mask, and finally ξ to adjust wavelength of the sinusoidal modulation.

We also use in the shape dictionary the *terminations* (end points) of patterns of MR8 and Gabor filters; more specifically, we shift by half the patch size in the direction perpendicular to the maximum gradient direction by some translation vector $v = (v_x, v_y)$, the uncovered regions after shifting are filled with the continuation of the shape that was out of view before the translation. Note that, these termination shapes will be shifted once more by the amount of $u = (u_x, u_y)$ as applied to all other shape models. Group III models, both models of conventional filters, i.e., F_{28}, F_{31}, F_{34}, F_{37} and models to created their terminations, i.e., F_{29}, F_{30}, F_{32}, F_{33}, F_{35}, F_{36}, F_{38}, F_{39}, are given in Table 3.4.

Table 3.4. Group III models: $v = (v_x, v_y)$ denotes the translation vector used to create terminations of shape patterns. (x', y') denotes the translated coordinates by $u = (u_x, u_y)$, i.e., $x' = x + u_x$, $y' = y + u_y$. Notice that $(u_x, u_y) = (0, 0)$ for central shapes.

Descriptive Name	**Shape function**	**Appear.**
Edge kernel	$F_{28}(x, y, u; \theta, \sigma) = \frac{\partial}{\partial y'_\theta}(\frac{1}{\sqrt{2\pi\sigma^2}} \exp - \frac{x'^2_\theta + y'^2_\theta}{2\sigma^2})$	
Edge kernel, termination 1	$F_{29}(x, y, v, u; \theta, \sigma) = \frac{\partial}{\partial y'_\theta}(\frac{1}{\sqrt{2\pi\sigma^2}} \exp - \frac{(x'_\theta + v_x)^2 + (y'_\theta + v_y)^2}{2\sigma^2})$	
Edge kernel, termination 2	$F_{30}(x, y, v, u; \theta, \sigma) = \frac{\partial}{\partial y'_\theta}(\frac{1}{\sqrt{2\pi\sigma^2}} \exp - \frac{(x'_\theta + v_x)^2 + (y'_\theta + v_y)^2}{2\sigma^2})$	
Bar kernel	$F_{31}(x, y, u; \theta, \sigma) = \frac{\partial^2}{\partial y'^2_\theta}(\frac{1}{\sqrt{2\pi\sigma^2}} \exp - \frac{x'^2_\theta + y'^2_\theta}{2\sigma^2})$	
Bar kernel, termination 1	$F_{32}(x, y, v, u; \theta, \sigma) = \frac{\partial^2}{\partial y'^2_\theta}(\frac{1}{\sqrt{2\pi\sigma^2}} \exp - \frac{(x'_\theta + v_x)^2 + (y'_\theta + v_y)^2}{2\sigma^2})$	
Bar kernel, termination 2	$F_{33}(x, y, v, u; \theta, \sigma) = \frac{\partial^2}{\partial y'^2_\theta}(\frac{1}{\sqrt{2\pi\sigma^2}} \exp - \frac{(x'_\theta + v_x)^2 + (y'_\theta + v_y)^2}{2\sigma^2})$	
Gabor kernel, phase 1	$F_{34}(x, y, u; \theta, \lambda, \sigma, \xi) = -\cos\frac{\pi y'_\theta}{\xi} \times \exp(-\frac{x'^2_\theta}{\lambda^2 \times \sigma^2} - \frac{y'^2_\theta}{\sigma^2})$	
Gabor kernel, phase 1, termination 1	$F_{35}(x, y, v, u; \theta, \lambda, \sigma, \xi) = -\cos\frac{\pi(y'_\theta + v_y)}{\xi} \times \exp(-\frac{x'^2_\theta + v_x}{\lambda^2 \times \sigma^2} - \frac{y'^2_\theta + v_y}{\sigma^2})$	
Gabor kernel, phase 1, termination 2	$F_{36}(x, y, v, u; \theta, \lambda, \sigma, \xi) = -\cos\frac{\pi(y'_\theta + v_y)}{\xi} \times \exp(-\frac{x'^2_\theta + v_x}{\lambda^2 \times \sigma^2} - \frac{y'^2_\theta + v_y}{\sigma^2})$	
Gabor kernel, phase 2	$F_{37}(x, y, u; \theta, \lambda, \sigma, \xi) =$Hilbert$(F_{34}(x, y, u; \theta, \lambda, \sigma, \xi))$	
Gabor kernel, phase 2, termination 1	$F_{38}(x, y, v, u; \theta, \lambda, \sigma, \xi) =$Hilbert$(F_{35}(x, y, v, u; \theta, \lambda, \sigma, \xi))$	
Gabor kernel, phase 2, termination 2	$F_{39}(x, y, v, u; \theta, \lambda, \sigma, \xi) =$Hilbert$(F_{36}(x, y, v, u; \theta, \lambda, \sigma, \xi))$	

3.2 Parametrization of the Shape Dictionary

In this section, we discuss the choices of the shape generation parameters and their effect. The parameters considered are: i) Quanta of rotation angles; ii) Quanta of relative angle for shape compounding; iii) Steepness of the hillsides, i.e., the transition rate; iv) Shape position vis-à-vis patch center. We have used Caltech-101 dataset to tune these dictionary parameters in this section, since it is one of the most diverse datasets in terms of inter-class variability. The preprocessing steps described in Chapter 6.1.3 are applied to the Caltech-101 images.

3.2.1 Quantization scheme for Group I and Group II models

We quantize the ranges of rotation angle, θ, and transition rate, α, parameters into q_θ and q_α discrete values, respectively, $\theta \in \{\theta_1, \ldots, \theta_{q_\theta}\}$ and $\alpha \in \{\alpha_1, \ldots, \alpha_{q_\alpha}\}$, and with the addition of these parameters the shape models are denoted as $F(x, y; \theta, \alpha)$. For each pair of $q_\theta \times q_\alpha$ values, i.e., $\{(\theta_1, \alpha_1), \ldots, (\theta_1, \alpha_{q_\alpha}), (\theta_2, \alpha_1), \ldots, (\theta_2, \alpha_{q_\alpha}), (\theta_{q_\theta}, \alpha_1), (\theta_{q_\theta}, \alpha_{q_\alpha})\}$ we obtain a shape variety with a particular orientation and a transition rate. For the compound shapes, we also set q_ψ quanta for the compounding angle parameter, ψ, resulting in $q_\theta \times q_\alpha \times q_\psi$ combinations, so that the shape function becomes: $F(x, y; \theta, \psi, \alpha)$. The quantization scheme applied for each type of generation parameters are described below.

Rotation angle parameter (θ). The shapes that possess inherent symmetry, such as a circular basin or mesa, are obviously rotation invariant, as in F_4, F_5. The shapes that possess two gradients in opposite senses, such as a bar, have 180^o symmetry, and hence can be rotated within the range of $[0, \pi]$, and these correspond to models: F_2, F_3, F_6, F_7, F_{14}, F_{15}, and these will be denoted as *bi-directional*. Finally, the shapes that possess one dominant gradient, such as a ramp, can be rotated over the range $[0, 2\pi]$, and these are the following: F_1, F_8, F_9, F_{10}, F_{11}, F_{12}, F_{13}, F_{16}, F_{17}, F_{18}, F_{19}, F_{22}, F_{23}, F_{24}, F_{27}, and we will refer to them as *uni-directional*.

We experimented with different values of q_θ, that is, splitting the 360^o into $q_\theta^{(uni-dir.)}$ ={4, 8, 12, 16, 20, 24} and the splitting the 180^o into $q_\theta^{(bi-dir.)}$ ={2, 4, 6, 8, 10, 12} slices, which correspond to rotational angle steps of $\{\frac{\pi}{2}, \frac{\pi}{4}, \frac{\pi}{6}, \frac{\pi}{8}, \frac{\pi}{10}, \frac{\pi}{12}\}$, respectively. The transition rate and compounding angle parameter values were fixed at $q_\alpha = 1$ and $q_\psi = 3$, as to be described in the sequel. Rotational angle step size of $\frac{\pi}{4}$, i.e., $q_\theta^{(uni-dir.)} = 8$ and $q_\theta^{(bi-dir.)} = 4$, is commonly used in the literature (Lowe, 2004; Lillholm and Griffin, 2008). However we have found that a finer

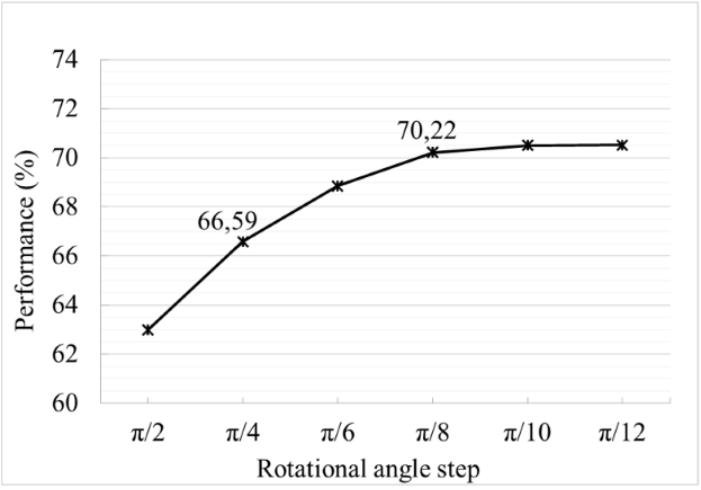


Figure 3.2. Effect of multiple shape orientations for Group I and II models. The vertical axis is the correct category recognition in the Caltech 101 dataset. Rotational angle step size, that is equal to $2\pi/q_\theta^{(uni-dir)}$ or $\pi/q_\theta^{(bi-dir)}$ is changed in the horizontal axis while the other parameters are fixed at $q_\alpha = 1$ and $q_\psi = 3$.

splitting, namely $\frac{\pi}{8}$ radiant separation, i.e., $q_\theta^{(uni-dir.)} = 16$ and $q_\theta^{(bi-dir.)} = 8$, brings about a ~3% performance improvement over splitting by the rotational angle step size of $\frac{\pi}{4}$ as seen in Fig. 3.2.

We did not experiment with smaller rotation angle steps beyond $\frac{\pi}{12}$, since the performance reached a saturation plateau. This is actually an expected outcome since BRIEF features computed on the shape patterns (to be described in Chapter 5.1), were reported to be insensitive to angular shifts smaller than $\sim\pi/9$ degrees (Crosier and Griffin, 2010). Two cases of exception were the cross shape patterns, F_{20} and F_{21}, which had only three quanta for their rotation angles, i.e., $q_\theta = 3$ with $\theta = \{\frac{\pi}{4}, \frac{7\pi}{12}, \frac{11\pi}{12}\}$, to preclude their duplication. These duplications are illustrated in Appendix 2. Notice that such duplications could also be avoided if different compounding angle slices were used, yet we preferred this solution for our case.

The Compounding angle parameter (ψ). We applied uniform quantization on the compounding angle, and we determined that three $q_\psi = 3$ quanta, i.e., $\psi = \{\frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}\}$ suffice, since adding finer quantization levels results in shape patterns that are too close to each other. Sample images of compound shapes with three settings of the compounding are presented in Fig. 3.3. Two cases of exception were the dark and light curved shape patterns, F_{26} and F_{27}, where only two quanta for their compounding angles, i.e., $q_\psi = 2$ with $\psi = \{\frac{\pi}{4}, \frac{\pi}{2}\}$ are used to preclude

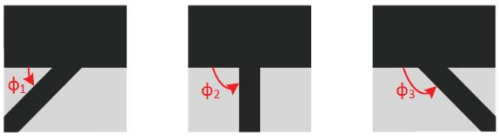


Figure 3.3. Three quantization values for the compounding angle of the dark junction shape pattern, i.e., F_{18}.

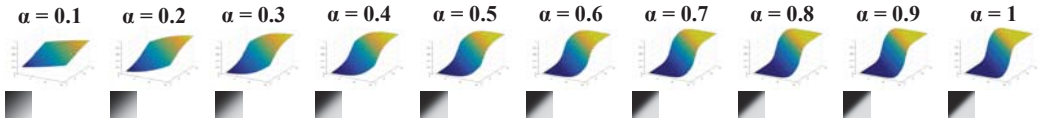
their duplication with mesa and basin shape patterns. The unintended appearance of shapes arise in the generation of Curve shape when the compounding angle quanta of $\psi = \frac{3\pi}{4}$ was applied are illustrated in Appendix 3.

Summary. This scheme of rotating shape models in Group I results in a shape dictionary of size $D_{\theta}^{(\text{Group I})} = \sum_{i=2}^{14} q_{\theta}^{F_i} \times F_i$, where $q_{\theta}^{F_i} \times F_i$ denotes the contribution of shape model F_i in Table 3.2 to the dictionary with $q_{\theta}^{F_i}$ number of rotated variants of it ($q_{\theta}^{F_i}$ denotes the number of quanta of rotation angle applied to the shape model F_i). Specifically, $\mathbf{D}_{\theta}^{(\textbf{Group I})}$ **= 154** ($16 \times F_1 + 8 \times F_2 + 8 \times F_3 + 1 \times F_4 + 1 \times F_5 + 8 \times F_6 + 8 \times F_7 + 16 \times F_8 + 16 \times F_9 + 16 \times F_{10} + 16 \times F_{11} + 16 \times F_{12} + 16 \times F_{13} + 8 \times F_{14}$).

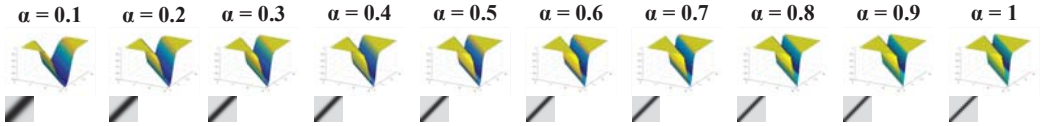
Rotating and compounding schemes applied to shape models in Group II results in a shape dictionary of size $D_{\theta,\psi}^{(\text{Group II})} = \sum_{i=15}^{27} q_{\theta}^{F_i} \times q_{\psi}^{F_i} \times F_i$, where $q_{\theta}^{F_i} \times q_{\psi}^{F_i} \times F_i$ denotes the contribution of shape model F_i in Table 3.3 to the dictionary with $q_{\theta}^{F_i} \times q_{\psi}^{F_i}$ number of rotated and compounded variants of it ($q_{\psi}^{F_i}$ denotes the number of quanta of compounding angle applied to model F_i). Specifically, $\mathbf{D}_{\theta,\psi}^{(\textbf{Group II})}$ **= 394** ($24 \times F_{15} + 48 \times F_{16} + 48 \times F_{17} + 48 \times F_{18} + 48 \times F_{19} + 9 \times F_{20} + 9 \times F_{21} + 16 \times F_{22} + 16 \times F_{23} + 48 \times F_{24} + 48 \times F_{25} + 32 \times F_{26} + 32 \times F_{27}$).

Consequently, the overall quantization in parameters related to orientation attribute of Group I and II models, results in a shape dictionary of size $D = D_{\theta}^{(\text{Group I})} + D_{\theta,\psi}^{(\text{Group II})}$, which is **D = 580**.

Transition rate parameter (α). Since the parameter α takes place in the exponent of the sigmoid function (Eq. 3.1), evaluating it with uniformly quantized values of α yields to an unbalanced set of shape appearances, i.e., too many shapes having rapid transition from background to foreground, as illustrated in Figure 3.4. Moreover, α does not have a linear effect on the appearances that are generated by shape



(a) Surface diagrams and corresponding gray-level shape appearances on the left-bottom of Ramp shape model (F_1) that are generated by uniformly quantized values of α.



(b) Surface diagrams and corresponding gray-level shape appearances on the left-bottom of Valley shape model (F_2) that are generated by uniformly quantized values of α.

Figure 3.4. Appearances of two shape models generated by uniformly quantized values of α, i.e., Ramp (F_1) and Valley (F_2) models that are defined by a polynomial function (given in Table 3.2), which has a degree of 1 and 2, respectively.

models defined by polynomial functions with different degrees. For example, the Valley shape, which is generated by the polynomial function that has a degree of 2, saturates to rapid transition from background to foreground at a smaller value of α, while the Ramp shape, which is generated by the polynomial that has a degree of 1, saturates to rapid transition at a higher value of α. Therefore, we cannot use uniform quantization for the parameter α, as well as, we'd better use different quantized values of α for the shape models having different degrees.

As a solution, we determined discrete values of α for each shape model by clustering the BRIEF features as follows:

i. Densely sample the variable α on its range, specifically 10^4 samples are drawn from the uniform distribution with probability density function in Eq. 3.2. Note that one can use another distribution, i.e., exponential distribution, to sample the α values from, anyway, clustering the BRIEF features of the shape patterns would overcome the biased attitude caused by uniform quantization. We used the range of $[a, b] = [0.01, 1]$, since sufficient variations of appearances could be achieved in this range for our patch size $p = 15$ pixels.

$$f(x) = \begin{cases} \frac{1}{b-a}, & \text{for } a \leqslant x \leqslant b, \\ 0, & \text{otherwise} \end{cases} \tag{3.2}$$

ii. Generate corresponding shapes, e.g., $F_i(x, y; \theta, \psi, \alpha)$ with the sampled values of α for some models i and for fixed θ and ψ,

iii. Cluster these shape patterns using K-medoids. Similar to K-Means, K-Medoids tries to determine clusters by minimizing the total distance, i.e. Hamming distance in our case, between each item in a cluster to the cluster center. Differently from K-Means, the center is certainly a member of the cluster in K-Medoids, whereas mean of the cluster members is used as the center in K-Means. We use *KMeans++* algorithm (Arthur and Vassilvitskii, 2007) to determine the initial medoids, which briefly selects cluster centers to assure as they are farthest from each other. Using these initial medoidal values, we run *Partitioning Around Medoids* (*PAM*) algorithm (Kaufman and Rousseeuw, 2009) iteratively to obtain final medoidal values.

iv. Adopt as quanta of the α parameter the values of the cluster medoidal shapes, i.e., $q_\alpha = K$.

We experimented with different values of q_α, that is, $q_\alpha = \{1, 2, 3, 4\}$ for all models of Group I and II, while rotation angle and compounding angle parameter values were fixed at $q_\theta^{(uni-dir.)} = 16$, $q_\theta^{(bi-dir.)} = 8$, and $q_\psi = 3$. Given the insensitivity of the performance results to higher number of quanta of this parameter (see Fig. 3.5), we decided to adopt $q_\alpha = 2$. In fact, the performance gain is a modest ~2% for $q_\alpha = 2$ vis-à-vis $q_\alpha = 1$ and larger values of q_α did not yield any significant advantage. Notice that in this terminology, $q_\alpha = 1$ is a single α value that is actually the value of the medoid of the single cluster, i.e., all BRIEF features generated by 10^4 number of α values. Thus, it is the one that yields to minimum tightness measure, i.e., total distances between items to medoid, computed for the cluster.

Adopting $q_\alpha = 2$ resulted in a shape dictionary of size $D = D_{\theta,\alpha}^{(\text{Group I})} + D_{\theta,\psi,\alpha}^{(\text{Group II})} = 2 \times (D_\theta^{(\text{Group I})} + D_{\theta,\psi}^{(\text{Group II})})$, which is $D = 1160$ (2×580).

3.2.2 Quantization scheme applied to Group III models

For the MR8 filterbank, we generated the edge and bar patterns in 6 orientations as in (Varma and Zisserman, 2005), but in lowest two scales $(\sigma_x, \sigma_y) = \{(1, 3), (2, 6)\}$ which resulted in 24 shape patterns which is appropriate for our block size, that is $p = 15 \times 15$. We use the open source code published in (Varma and Zisserman, 2007) to generate MR8 filter set.

We generated Gabor patterns in 6 orientations by the PMT toolbox published in (Dollár, 2014). MR8 bar patterns appear as Gabor-like structures when proper parameter settings were used, so in order to preclude duplications we just generate

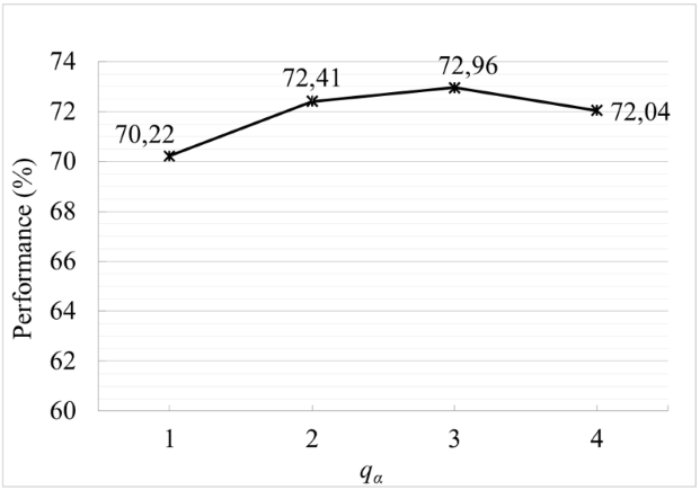


Figure 3.5. Effect of having multiple transition slopes, q_α, for Group I and II models, while the other parameters fixed at $q_\theta^{(\text{uni-dir.})} = 16$ and $q_\theta^{(\text{bi-dir.})} = 8$ (i.e., rotational angle step is $\pi/8$), and $q_\psi = 3$.

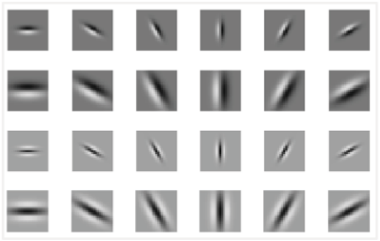
odd-phased Gabor kernel patterns at the scale $\sigma = 2.6$. The odd-phased pattern is enabled by the Hilbert transform which introduces 90-degree phase shift to the sinusoidal components in the Gabor waveforms. We also generate even- and odd-phased patterns with a higher scale of $\sigma = 6.6$. They all constitute 18 shape patterns. Here, we use $\lambda = 1.8$ for the elongation of the Gaussian mask and $\xi = 1.1$ for the wavelength of the modulating sinusoid.

The 42 shape patterns created by the MR8 and Gabor filter generators in Group III models with the mentioned scale and orientation parameter values are visualized in Fig. 3.6. We also create terminations of these shapes by shifting them in the amount of $v = (v_x, v_y)$, so the set of shapes in Group III triples from 42 to 126 patterns. Thus, the size of shape dictionary including rotated, scaled, even- and odd-phased variants of Group III models is $D_{\theta,\sigma,\lambda,\xi,v}^{(\text{Group III})} = 126$.

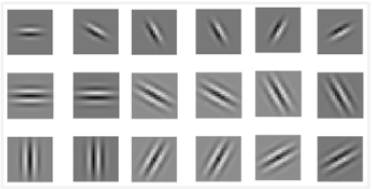
3.2.3 Enriching the dictionary by shift operations

For $p = 15 \times 15$ sized patches, we found it adequate to execute only one shift size, by 3 pixels, as this corresponds to 43% of one symmetrical half of the patch size. Generating shape patterns shifted by the amount $u = (u_x, u_y)$, u_x and u_y, i.e., $x' = x + u_x$, $y' = y + u_y$ and $(u_x, u_y) = (0, 0)$ for the non-shifted shapes, the shape function becomes: $F(x, y, u; \theta, \psi, \alpha)$.

For the patterns F_1, F_2, F_3, F_{10} and F_{14}, whose shape functions are given in



(a) MR8 edge and bar patterns generated at scales of $(\sigma_x, \sigma_y) = \{(1,3),(2,6)\}$ and at six orientations.



(b) Gabor patterns generated with $\xi = 1.1$ and $\lambda = 1.8$, (the first row indicates Hilbert transformed Gabor patterns generated at scale $\sigma = 2.6$, second and third rows indicate Gabor kernels and their Hilbert transformed version sequentially at scale $\sigma = 6.6$) and at six orientations.

Figure 3.6. Shape patterns generated by Group III models, block size is $p = 15 \times 15$.

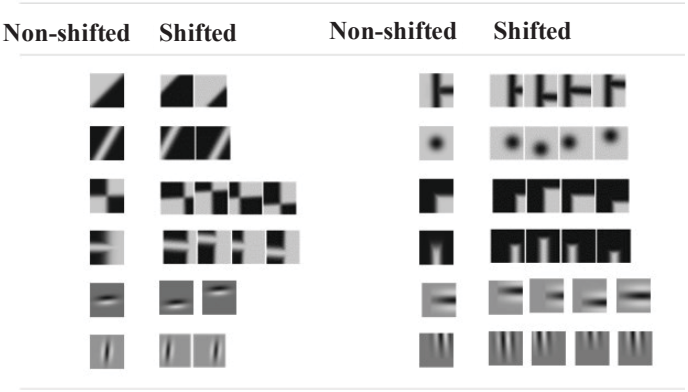
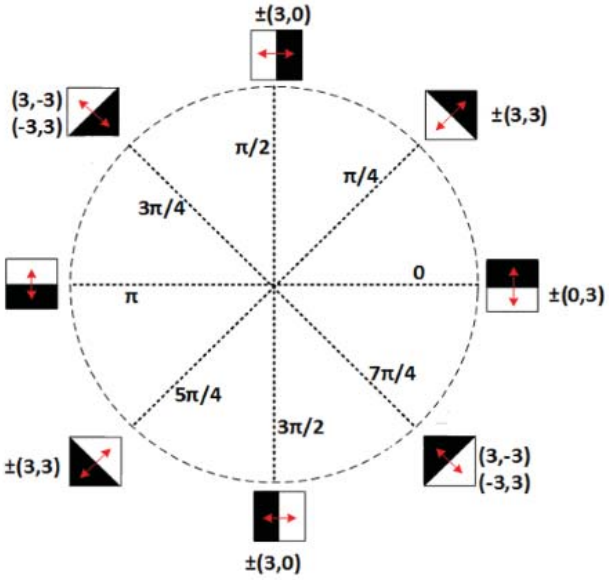


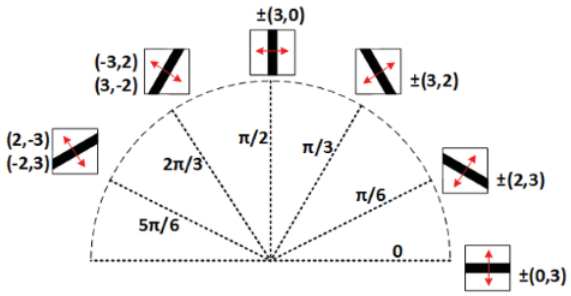
Figure 3.7. Some examples to non-shifted and shifted shape patterns from Groups I to III.

Table 3.2, a meaningful shift can be done along the sense of the gradient. These shapes that have only one dominant gradient direction can be shifted in two senses as illustrated in Figure 3.8. On the other hand, shapes in Table 3.2, and Table 3.3 that have two or more dominant directions, such as a corner, and terminations (end points) in Group III models are shifted along four directions with $(u_x, u_y) = \{(3,0); (0,3); (-3,0); (0,-3)\}$. Shape patterns in Fig. 3.6 that are created by MR8 and Gabor filter generators in Group III models are shifted along the two senses as applied to the valley/ridge patterns, to preclude duplicates with their termination patterns. Some examples to non-shifted and shifted patterns are given in Fig. 3.7.

This scheme of shifting shape models in Group I to III results in a shape dictionary of size $\mathbf{D}^{(\text{Group I, II, III})}_{\theta,\psi,\alpha,\sigma,\lambda,\xi,u,v} = \mathbf{6122}$.



(a) Shifting applied to ramp shape patterns having different rotation angles.



(b) Shifting applied to valley/ridge shape patterns having different rotation angles.

Figure 3.8. Shifting of shape patterns that have one dominant gradient direction. Ramp and valley/ridge shapes are given as example. Shifting applied along the directions indicated by a red double-arrow. One shift by 3 pixels was adequate for the $p = 15 \times 15$ sized patches.

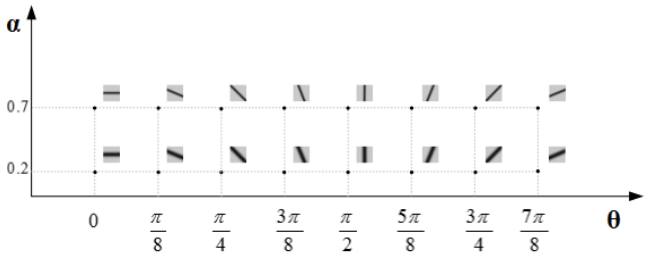
Table 3.5. Shape dictionaries obtained by different schemes of parametrization of the shape models.

Dictionary	Description	Size
$\mathbf{D}^{(\text{Group I, II, III})}_{\text{no quantization}}$	Shape dictionary obtained when no quantization is applied on the range of parameters. This scheme yields to describe image patches in terms of 39 core shape models without discriminating parametric variants of them.	39
$\mathbf{D}^{(\text{Group I, II, III})}_{(\alpha,\sigma)}$	Shape dictionary obtained when ranges of the parameters α and σ are quantized into q_α and q_σ discrete values, respectively, which are determined as in Section 3.2, and N samples of the parameters θ, ψ, and u are randomly chosen in their ranges.	76
$\mathbf{D}^{(\text{Group I, II, III})}_{(\theta,\psi)}$	Shape dictionary obtained when ranges of the parameters θ and ψ are quantized into q_θ and q_ψ discrete values, respectively, which are determined as in Section 3.2, and N samples of the parameters α, σ, and u are randomly chosen in their ranges.	652
$\mathbf{D}^{(\text{Group I, II, III})}_{(\theta,\psi),(\alpha,\sigma)}$	Shape dictionary obtained when ranges of the parameters θ, ψ, α and σ are quantized into q_θ, q_ψ, q_α and q_σ discrete values, respectively, which are determined as in Section 3.2, and N samples of the the parameter u are randomly chosen in its range	1286
$\mathbf{D}^{(\text{Group I, II, III})}_{(\theta,\psi),(\alpha,\sigma),u}$	Shape dictionary obtained when all parameters are quantized in their ranges. This parametrization scheme have been used in SymPaD construction.	6122

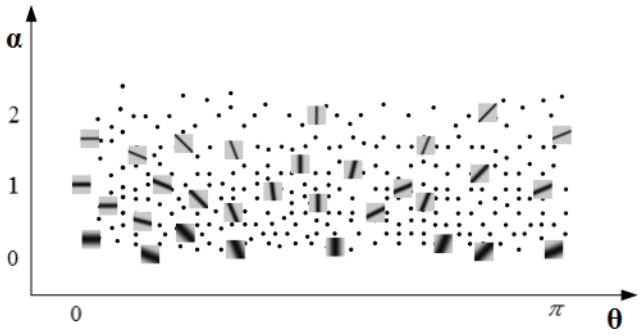
3.3 Overview of the Shape Dictionary

Having discussed the parametrization of the core shapes, we want to address the question of whether dictionaries incorporating more and more shape varieties will perform better and if so, what would be the most parsimonious dictionary. Thus, we ran an experiment for category recognition problem at Caltech-101 dataset with five dictionaries that are summarized in Table 3.5. We worked by Caltech-101 images at 75 pixels on the longest side and the preprocessing steps presented in Chapter 6.1.3 are applied on the images.

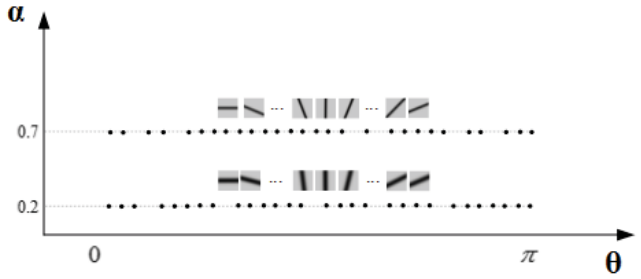
We present some illustrations in Fig. 3.9 to explain how we applied different quantization schemes to obtain the differently parametrized shape dictionaries that are presented in Table 3.5. We use the shape model *Valley* in these illustrations and



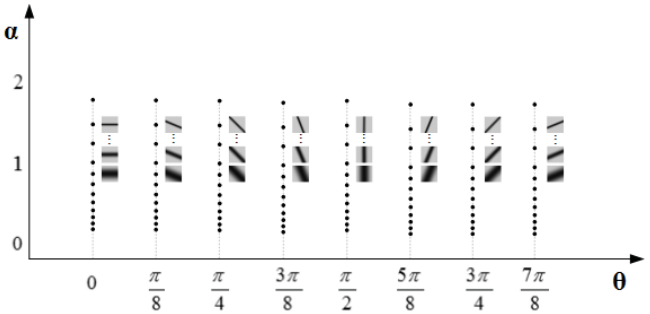
(a) Ranges of θ and α are quantized into $q_\theta = 8$ and $q_\alpha = 2$ discrete values. This scheme of quantization have been used in SymPaD construction.



(b) Nor α, neither θ is quantized, but N samples are drawn from the whole range of parameters.



(c) Range of α is quantized into $q_\alpha = 2$ discrete values, and N random samples are drawn from uniform distribution in the range where Valley is defined, i.e., $[0, \pi]$.



(d) Range of θ is quantized into $q_\theta = 8$ discrete values, and N random samples are drawn from exponential distribution with the scale parameter of the distribution is set to $\mu = \alpha^*$ where α^* is the value determined by KMedoids (K=1) for the Valley shape as explained in Section 3.2.1.

Figure 3.9. Four schemes of quantization of the parameter ranges in (α, θ) plane which is illustrated for an example shape model, i.e., Valley.

we applied quantization to ranges of its two parameters to be as an example, yet same schemes are applied to other shape models in Group I to III and for other parameters of the models to construct the dictionaries in Table 3.5. Ranges of two parameters θ and ψ can be demonstrated on the (θ,α) plane as in Figure 3.9. We present the quantization on the ranges of only two parameters θ and α for a plain explanation, yet one should consider quantization at *3-Dimensional* (*3D*) space when one more parameter, e.g., u, was incorporated.

Fig. 3.9.(a) shows the distribution of quantized values on the (θ, α) plane which demonstrates the parametrization scheme that have been used to construct the SymPaD dictionary, i.e., $D^{(\text{Group I, II, III})}_{(\theta,\psi),(\alpha,\sigma),u}$, in size $D = 6122$. Ranges of parameters θ and α of Valley model are quantized into $q_\theta = 8$ and $q_\alpha = 2$ discrete values which yields 16 variants of Valley pattern incorporate to the shape dictionary. Thus, by this scheme, a test patch will be encoded in terms of 16 variants of *Valley* by hard-voting (to be explained in Chapter 5.2, Eq. 5.3).

One should randomly sample N values from the (θ, α) plane when no quantization is applied on the ranges of the parameters as demonstrated in Fig. 3.9.(b). Generating N shape patterns from the randomly sampled N values of parameters for each shape model, a testing image patch can be encoded in terms of 39 core shape models, i.e., from Group I to III, by the method called *common voting* (Jiang et al., 2005). According to this method, N nearest neighbours of a test patch is found among $39 \times N$ shape patterns and votes on 39 core models among these N nearest neighbours are counted. Then, the test patch is encoded by the type of the shape model having majority of the votes. Thus, all differently oriented or with different transition rated variants of a shape model constitutes to a single dictionary element, for example, N samples of parameter values chosen in (θ, α) plane that are shown in Fig. 3.9.(b) represent a single core model, which is *Valley*, at the shape dictionary. This scheme of parametrization results with the dictionary denoted by $\mathbf{D}^{(\text{Group I, II, III})}_{\text{no quantization}}$ in Table 3.5, which is in size of $D = 39$.

Continuing to the same methodology, one can quantize ranges of some particular parameters into some discrete values, and not apply any quantization to the ranges of the remaining parameters, i.e., means would just choose N random samples in their ranges. In this way, we can observe the effect of some particular parametrizations on the performance. We illustrate such scheme of quantizing, for the ranges of parameters α and θ in Fig. 3.9.(c) and (d), respectively. Once the dictionary is obtained by this scheme, we encode a test patch by common voting among its N-nearest neighbours retrieved. However, this time the dictionary size will be

$39 \times q_{\omega_1} \times \ldots \times q_{\omega_i}$, when quantization is applied to the ranges of i parameters $\omega_1, \ldots, \omega_i$, and N samples are drawn randomly from the ranges of parameters in the set of $\{\Gamma | \Gamma = \gamma \setminus \omega_1, \ldots, \omega_i\}$, where $\gamma = \{\theta, \psi, \alpha, \sigma, \ldots\}$, whatever applicable. Then, for example, quantizing only the α range into $q_\alpha = 2$ discrete values of the Valley shape as in Fig. 3.9.(c) would yield 2 variants of it incorporate to the shape dictionary, and quantizing only the θ range into $q_\theta = 8$ of the Valley shape as in Fig. 3.9.(d) would yield 8 variants of it incorporate to the shape dictionary.

The distributions where we randomly choose N samples of the parameters to constrcut the dictionaries in Table 3.5 are as follows: (i) α samples are drawn from the exponential distribution with the scale parameter, i.e., mean of the distribution, is set to α^* that is determined by K-Medoids clustering when $K = 1$, as explained in Section 3.2.1. (ii) θ samples are drawn from uniform distribution in the range where the shape models are defined, i.e., $[0, \pi]$ or $[0, 2\pi]$. (iii) u samples, i.e., u_x and u_y position from the patch center in x and y axis, are drawn from uniform distribution in the range of half of the patch size, i.e., $[-3, 3]$, (iii) σ samples used in Group III model functions are drawn from the uniform distribution in the default ranges where the MR8 and Gabor filters have been used. In our experiments we choose $N = 50$ samples as we employed in our previous paper in (Aslan et al., 2015).

We present performance results obtained by these five dictionaries given at Table 3.5 in Figure 3.10. The shape dictionary which is not parametrized, i.e., $D^{(\text{Group I, II, III})}_{\text{no quantization}}$, yields to the performance of 36.14%, and applying quantization to ranges of (α,σ), i.e., as in the dictionary $D^{(\text{Group I, II, III})}_{(\alpha,\sigma)}$, improves the performance to 45.68%. Quantizing the ranges of θ and ψ, that is in $D^{(\text{Group I, II, III})}_{(\theta,\psi)}$, yields to the performance of 68.92%, which doubles up the performance of the dictionary $D^{(\text{Group I, II, III})}_{\text{no quantization}}$. Thus, we observe that highest improvement on the performance is obtained by quantizing the ranges of the parameters θ and ψ. Quantizing ranges of both (θ, ψ) and (α,σ) parameter pairs yields to the performance of 71.33% and finally quantizing the shifting parameter we obtain the SymPaD dictionary in size of $D = 6122$ which yield to the performance of 75.29%. We observe that while the size of the final dictionary, i.e., $D^{(\text{Group I, II, III})}_{(\theta,\psi),(\alpha,\sigma),u}$ is increased by ten times of the dictionary $D^{(\text{Group I, II, III})}_{(\theta,\psi)}$, the gain in performance was in ~5% which makes us to conjecture that there is probably some redundancy in the final dictionary.

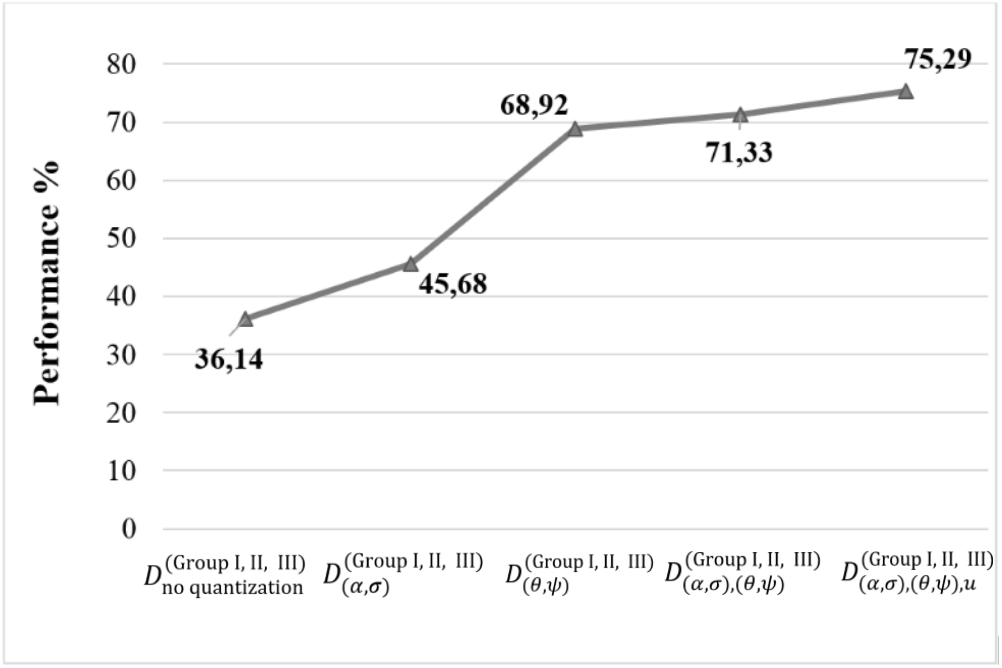


Figure 3.10. Performance of shape dictionaries in Table 3.5 on Caltech-101 dataset.

4. PRUNING THE SHAPE DICTIONARY

Once we have built a dictionary as a collection of shape patterns guided by image qualitative criteria, the next step is to prune it into a more compact dictionary without sacrificing on the performance. This will alleviate not only the computational effort, but also, possibly, improve the classification performance. There are several ways we can proceed to obtain a more concise dictionary:

- **Unsupervised clustering:** We can essentially apply vector quantization to the descriptors of the shapes, and prune the initial dictionary to some desired goal dimension in an unsupervised manner. Clustering relies on a notion of closeness, measured by a given objective function. The goal is trying to create tight clusters by minimizing the given objective function. A very common technique of unsupervised clustering has been *K-Means* algorithm.

- **Discriminative clustering:** The goal of these techniques is trying to separate data into clusters with high discrimination. *Maximum Margin Clustering* (*MMC*) introduced by (Xu et al., 2004) is one of the first proposed techniques. MMC adopts the *Support Vector Machine* (*SVM*) cost function used in linear classification as a clustering criterion and tries to find clusters that are separated by large margins, so that they are most linearly separable. Another type of implementation of discriminative clustering is achieved by combining *Linear Discriminant Analysis* (*LDA*), a dimensionality reduction technique, with clustering techniques (De la Torre and Kanade, 2006; Ding and Li, 2007).

- **Spectral clustering:** These techniques are specifically developed for graphs where data points are represented by vertices that are connected by weighted undirected edges. The goal is finding partitions on the graph such that edges from different clusters have low weigth. Integrated *KMeans - Laplacian* (*KL*) clustering is a successful implementation proposed by (Wang et al., 2009) where the similarity-based kernel matrix and Laplacian matrix are combined.

- **Feature selection techniques:** Shape atoms can be selected by wrapper or filter feature selection techniques (Pohjalainen et al., 2015). Wrapper techniques add or remove features based on their contribution to the classification problem. Techniques employing greedy search such as *Sequential Backward Elimination* (*SBE*) (Marill and Green, 1963), *Sequential Forward Selection* (*SFS*) (Whitney, 1971), *Sequential Floating Forward Selection* (*SFFS*) (Pudil et al., 1994), and ensemble methods such as Adaboost (Freund et al., 1999) are in this group. Wrapper techniques are computationally very intensive and they

have a higher risk of overfitting. Filter techniques are generally preprocessing algorithms and feature selection procedure is independent from the classification algorithm. Ranking scores are computed for the features according to intrinsic characteristics of data, e.g., some probabilistic dependence measures, rather than classification accuracy criteria. Common measures used in filter techniques are (Peng et al., 2005; Fleuret, 2004; Kwak and Choi, 2002; Battiti, 1994), pointwise mutual information (Yang and Pedersen, 1997), and Pearson product-moment correlation coefficients (Hall, 1999). Filter techniques ignore feature dependencies, yet they are much faster than the wrappers and can scale to high-dimensional data.

In this work, we opted for the feature selection technique using the mutual information principle (Kwak and Choi, 2002; Battiti, 1994). Therefore, we rank the shape models according to mutual information scores between the shape models and the categories, and select the highest scoring subsets.

We compute the *Mutual Information* (*MI*) between the occurrence probability, w_i, of dictionary atoms in images, w_i, $i = 1, \ldots, D$, of a D-dimensional dictionary and image categories (classes, objects etc.), $y \in \{1, 2, \ldots, C\}$ in a C-category image dataset. The score w_i, the i^{th} element of the signature vector W (to be described in Chapter 5.3, Eq. 5.5), is the empirical distribution of the i^{th} shape pattern score on the training images. Higher dependency between occurrence probability of a shape pattern and category labels yields a higher mutual information score, which indicates that that particular shape pattern is probably significant to discriminate the related categories. Thus, we rank the shape patterns according to their MI score and select the highest R ones to form the pruned dictionary.

The algorithmic stages of the implementation are given in Appendix 4. Briefly:

i. Choose a training set of images on which to calculate the shape occurrence probabilities w_i where $i = 1, \ldots, D$ and D is the number of shape patterns;

ii. Discretize the components of the vector w_i (Eq. 5.5), $i = 1, \ldots, D$, using their empirical distribution and name the discretized vector as X_i; notice that this discretization can be quite coarse, even a two-level quantization may suffice;

iii. Compute the mutual information s_i of the i^{th} shape pattern as in Eq. 4.1. In Eq. 4.1, X_i denotes the discretized signature component of the shape pattern i, Y denotes the category label of the training images, i.e., $y \in 1, \ldots, C$ and

C is the number of categories in the image set. In this work it was sufficient to apply binary quantization to the signature components, hence $x \in 0, 1$;

$$s_i = \text{MI}(X_i, Y) = \Sigma_{x\in\{0,1\}}\Sigma_{y\in\{1,2,\ldots,C\}} \Pr[X_i = x, Y = y] \log \frac{\Pr[X_i = x, Y = y]}{\Pr[X_i = x]\Pr[Y = y]} \tag{4.1}$$

iv. Rank the shape patterns, i.e., dictionary atoms, according to their MI-scores from highest to lowest, and select the top R shape patterns as in Eq. 4.2. In Eq. 4.2, γ denotes the parameterization of the shape model F_i whatever applicable, i.e., $\gamma \in \{\theta, \psi, \alpha, \sigma, \ldots\}$, $s_{(i)}$ denotes the mutual information score of the i^{th} shape pattern in the rank order.

$$\text{Pruned dictionary} = \cup_{i=1}^{R}\{F_i(x, y, u; \gamma) | s_{(i)} \leq s_{(R)},\ i = 1, \ldots, R\} \tag{4.2}$$

We computed the ranked and pruned dictionaries for each of the four different image datasets, i.e., COIL, ALOI-View, Caltech-101, and ZuBuD, separately. These datasets will be introduced in Chapter 6, yet we used different experimental settings for these datasets in this chapter than we will use in Chapter 6 experiments. In Chapter 6, we follow the settings that have been used in the literature studies, however the number of training images per category used in some of these settings might not suffice for MI-score computation, e.g., 8 images are sampled per category into training set in SETUP-2 to evaluate performance on ALOI-250 and COIL-100 images. Thus, we randomly sampled 30 images from each category to construct the training sets for MI score computation as in the standard setting of Caltech-101 experiments, and the remaining images are incorporated into the testing set. This scheme is employed for all datasets except ZuBuD, since ZuBuD has much fewer number of images, and hence we sample just 5 images from each category of it. We used the lowest resolution Caltech images, i.e., 75-pixels on the longest side. The preprocessing operations applied on the dataset images are explained in Chapter 6.1. Number of categories for Caltech, Coil, ALOI and ZuBuD are, respectively, $C = 100, 100, 250,$ and 201, while the number of training images for MI computation are, 30, 30, 30, and 5.

Table 4.1. Performance results obtained on Caltech dataset by highest ranked **R** shape patterns where MI score is computed on samples discretized by different techniques.

	Performance	
Discretization techniques	**R = 256**	**R = 512**
Binary quantization by median thresholding	70.3%	73.3%
Binary quantization by mean thresholding	69.9%	73.2%
3-level quantization by EFB	69.9%	73.2%
10-level quantization by EFB	70.1%	72.4%

4.1 Discretization

To discretize the components w_i of the signature vector, i.e., the shape occurrence probabilities, various methods of feature discretization have been employed in the literature (Kotsiantis and Kanellopoulos, 2006; Dougherty et al., 1995; Holte, 1993; Kerber, 1992). Unsupervised methods do not make use of class membership information such as *Equal Frequency Binning* (*EFB*) and *Equal Width Binning* (*EWB*) (Kotsiantis and Kanellopoulos, 2006; Dougherty et al., 1995), while supervised methods use class labels to accomplish discretization (Holte, 1993; Kerber, 1992). We chose the unsupervised discretization technique called EFB, and ran experiments with quantization levels between 3 to 10 discrete values, as well as two-level quantization where the slicing level was chosen as the sample *mean* or as sample *median*. This initial experiment is run on the Caltech dataset merely. We used $R = 256$ and $R = 512$, as the pruned dictionary sizes and selected R highest ranking shape patterns according to MI scores. The classification performance obtained by various techniques are presented in Table 4.1. We observe that unsupervised discretization into *binary* values worked quite well, even slightly better (~1%) than discretization by EFB into 3 and 10 levels. Furthermore for the case of binary quantization, *median* was slightly better than *mean* thresholding.

4.2 Analysis of the Pruned Dictionary

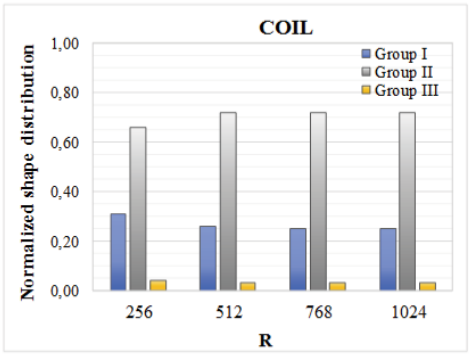
We obtained pruned dictionaries with sizes $R = \{256, 512, 768, 1024\}$ out of the $D = 6122$ shapes for four different image datasets. Shape patterns of these pruned dictionaries are illustrated in Appendices 5 to 8. In this subsection, we investigate the type of shape patterns that could take part in the pruned dictionaries predominantly. These patterns have the highest MI scores, thus we conjecture that

these are more significant to discriminate the image categories. For a modest presentation, we portioned these R number of shape patterns in two ways: (i) We collapsed R shape patterns to three groups of models that were mentioned at Sections 3.1.1, 3.1.2 and 3.3, i.e., Group I, II, and III, respectively. This provides a *coarser analysis*, but brings a general intuition about the pruned dictionary; (ii) We collapsed R shape patterns to 19 groups of models. Each group includes all parametrizations, i.e., rotations, offsets from center, compounding angles, transition rates, and light/dark contractions of the main shape appearances, i.e., ramps, lines, blobs, elliptical blobs, corners, junctions, etc. This type of grouping provides a *finer analysis*.

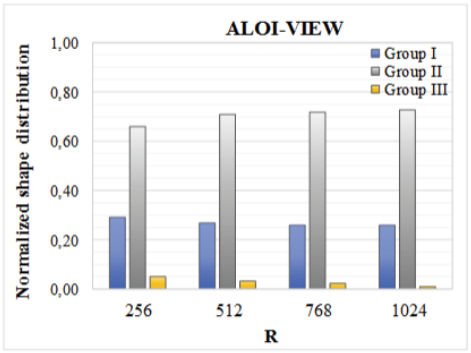
Coarser analysis. The portions of the pruned dictionary obtained from the three shape model groups (Group I, II and III) are shown in Fig. 4.1. Group II models, i.e., the compounded models, are predominantly represented in the four final dictionaries with sizes $R = \{256, 512, 768, 1024\}$. This is indicative of the fact that this set contains discriminative shape patterns. This is to be expected, since Group II shape patterns are spatially organized forms of Group I patterns, and they are more unlikely to occur by accident but prone to occur on the interest parts of the objects as asserted in the literature (Marr, 1976; Tenenbaum and Witkin, 1983; Lowe, 1984; Horaud et al., 1990). The dictionaries pruned according to the three image datasets, i.e., COIL, ALOI-VIEW and ZuBuD, do not differ significantly from each other. However, for the smaller dictionaries, i.e., R = 256, 512, Group I models are represented in higher probability when pruning is achieved according to Caltech dataset than it was for the remaining three datasets in the smaller dictionaries; nevertheless Group II models were still the most predominant ones in these smaller dictionaries.

In Figure 4.2, we present normalized MI scores that were accumulated over three groups of shape models and averaged over four datasets. We did not obtain this chart in Figure 4.2 from the pruned dictionary, but we considered the initial dictionary with 6122 atoms to present general trend in MI scores throughout whole dictionary elements. Thus, we accumulated MI scores of patterns collapsed in three group of models and normalized them by dividing each portion to sum of all scores. According to this chart, MI scores of 2^{nd} group are quite higher than the MI scores of the other two groups. Moreover, MI scores of groups computed on different dataset were so close to each other. We present the bar chart in Fig. 4.2 in order to justify that Group II patterns could take part in the ranked portions of pruned dictionary with high majority, since their MI scores were quite higher than the others.

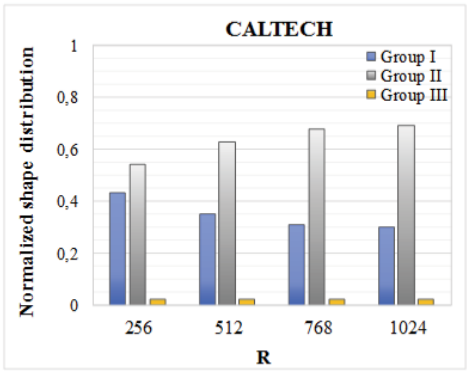
In order to demonstrate which group of models represent which regions of



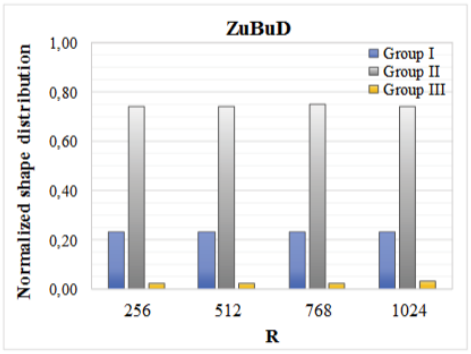
(a) Population of R highest ranked patterns when MI scores are computed according to COIL-100 images.



(b) Population of R highest ranked patterns when MI scores are computed according to ALOI-View images.



(c) Population of R highest ranked patterns when MI scores are computed according to Caltech-101 images.



(d) Population of R highest ranked patterns when MI scores are computed according to ZuBuD images.

Figure 4.1. Populations of highest ranked patterns from the three group of models.

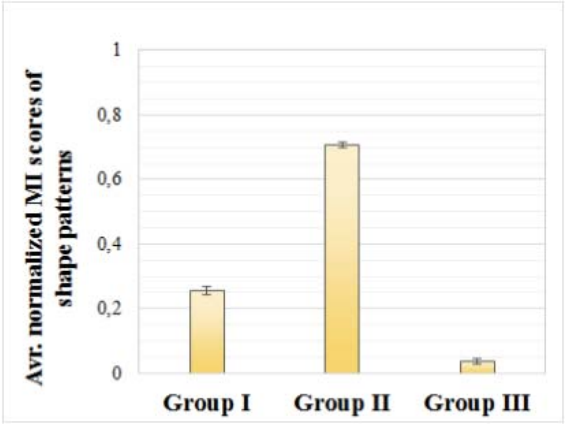


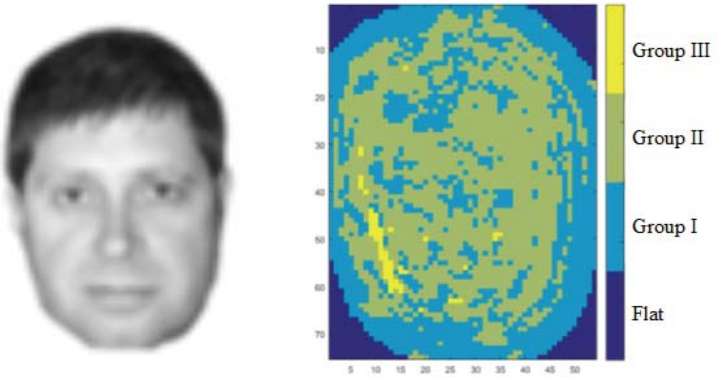
Figure 4.2. Normalized MI scores of shape patterns accumulated over three groups of models. Average and standard deviation of values obtained on four datasets are presented.

the images, we randomly sampled four images from two datasets, i.e., Caltech-101 and COIL-100, and compute label maps of them that are illustrated in Fig. 4.3. In order to obtain these label maps, we first labelled each pixel in terms of its nearest neighbour among $D = 6122$ shape patterns and we then categorised these labels in terms of three groups of models. We observe in Fig. 4.3 that pixels at the interest parts of the objects are mostly labelled as Group II category, thus we can conjecture that they are more likely to to be significant to discriminate object categories, though Group I labels appear on the pixels that mostly locate on or around the borders of the objects.

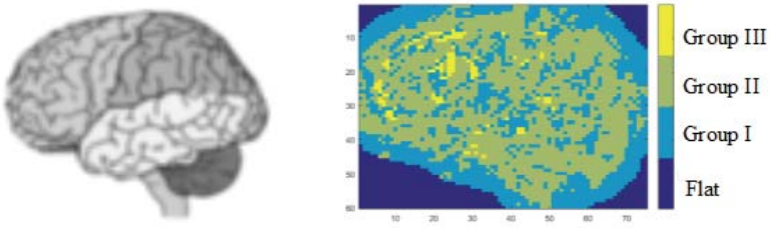
Finer analysis. Figures 4.5 and 4.6 show the percentage of top shape models of the dictionaries pruned according to four image datasets. Note that only 19 shapes appear below the bars in these figures; these are the shapes collapsed from the R selected patterns of the pruned dictionary, when we have accumulated their hit probabilities over all its parametrizations, i.e., rotations, offsets from center, compounding angles, transition rates and light/dark contractions.

We observe that corners (F_{16}, F_{17}), junctions (F_{18}, F_{19}), T-junctions (F_{24}, F_{25}) from Group II models set and ramps (F_1) from Group I model set are the most significant shape patterns in all datasets. Actually, the distribution of shapes are similar in the largest pruned dictionary with size R=1024, yet some differences exist from dataset to dataset in smaller ones. For example, ramp-like shapes, i.e., (F_{10}), are more dominant in R=256 when it is pruned with respect to Caltech images than it is to other dataset images. Corner shape is predominant in R=256 pruned with respect to Caltech, COIL, ALOI-View datasets, while T-junction and junctions hapes are predominant when R=256 is pruned with respect to ZuBuD dataset. We observe that the least occurring shapes in the pruned dictionary are mesa and basin (F_4, F_5) and parallel valleys/ridges shapes that were generated by the transcendental function F_{14}.

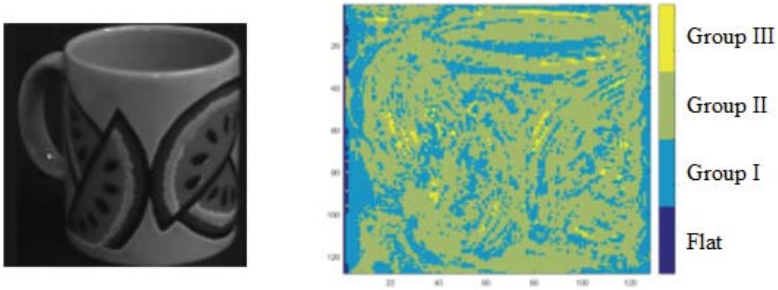
In Figure 4.4, we present normalized MI scores of 19 type of models that are averaged over four datasets. We considered the MI scores of the 6122 shape patterns, i.e., the unpruned dictionary, to obtain this chart, as we did at Fig. 4.2. We observe that junctions and corners from Group II models have the highest MI scores. Among the patterns generated by the Group I models, ramps, and ramp-like models in Table 3.2, i.e., F_1, F_8, F_{10}, F_{11}, F_{12}, F_{13}. Consequently, it is not surprise that the several kind of junctions and corners take part in the top ranked portions of the pruned dictionary in Figures 4.5 and 4.6 with high majority, since these have the



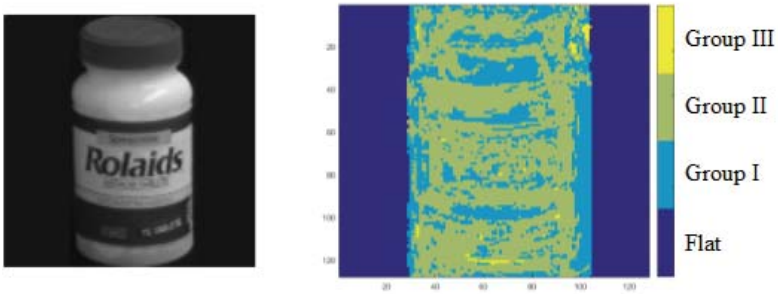
(a) Pixel label map in terms of Flat and three groups of models of the image randomly chosen from Caltech-101 dataset. The original image size is 75×54 pixels.



(b) Pixel label map in terms of Flat and three groups of models of the image randomly chosen from Caltech-101 dataset. The original image size is 60×75 pixels.



(c) Pixel label map in terms of Flat and three groups of models of the image randomly chosen from Coil-100 dataset. The original image size is 128×128 pixels.



(d) Pixel label map in terms of Flat and three groups of models of the image randomly chosen from Coil-100 dataset. The original image size is 128×128 pixels.

Figure 4.3. Pixel label maps of four images randomly chosen from Caltech-101 and Coil-100 datasets.

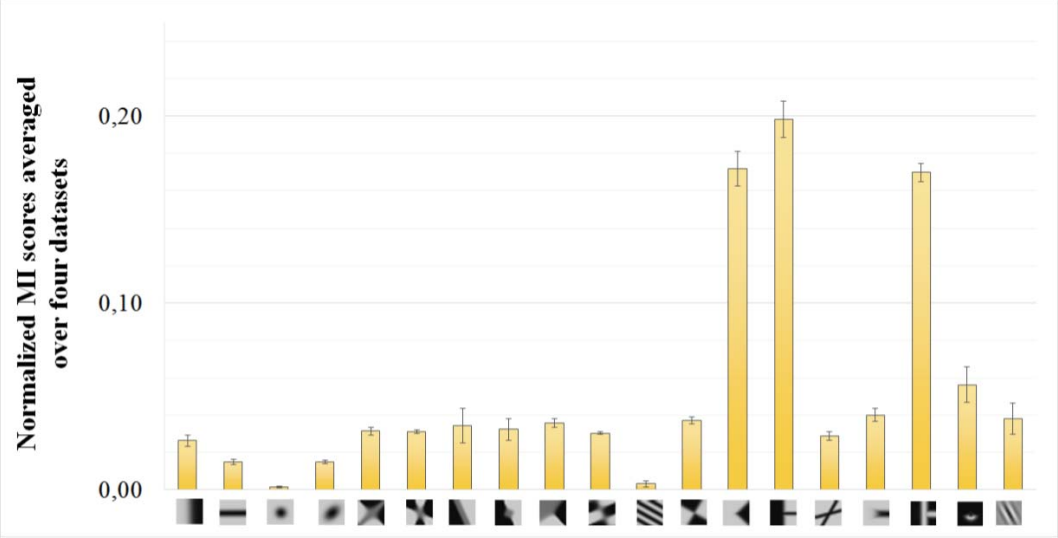


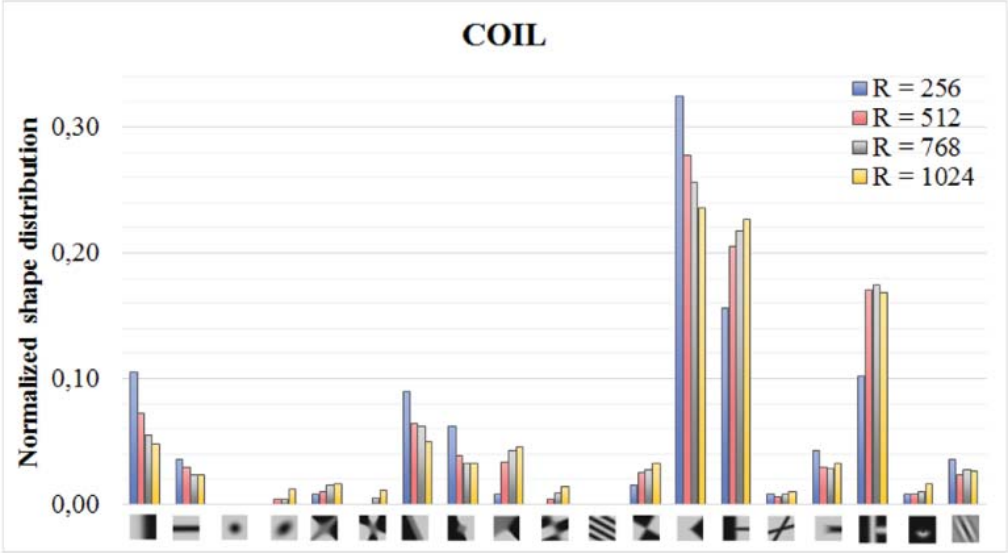
Figure 4.4. Normalized MI scores of shape patterns accumulated over 19 model types. Average and standard deviation of values obtained on four datasets are presented.

highest MI scores.

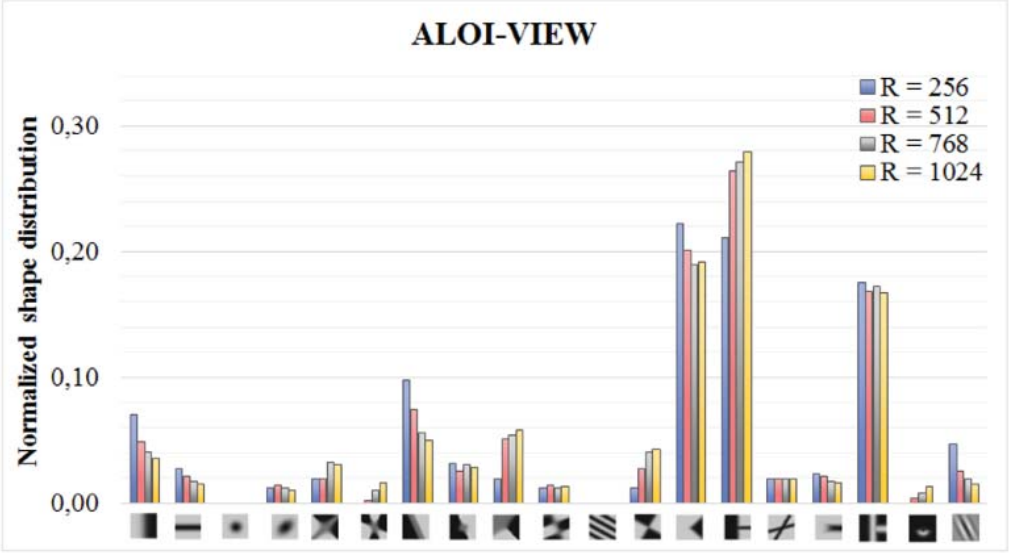
We present the label maps of four images computed for 19 type of shape models in Figures 4.7 and 4.8. In order to obtain these maps, we first labelled each pixel in terms of its nearest neighbour among $D = 6122$ shape patterns and we then categorised these initial labels in terms of 19 main models, thus all different parametrization, i.e., offsets from center, compounding angles, transition rates and light/dark contractions, of a main model is labelled by a single color-code. We observe that the interest regions of the objects at the interior parts of object borders are mostly labelled by greenish to yellow color-codes. These indicate the matches of saddle, corner, and junction shapes on these regions.

4.2.1 Performance evaluation of the pruned dictionaries

We present the performance results obtained by the initial dictionary with $D = 6122$ atoms, and pruned dictionaries with respect to four image datasets and having $R = \{256, 512, 768, 1024\}$ atoms in Fig. 4.9. Performances obtained at evaluation on ALOI-250 and COIL-100 images is quite high even with R=256 sized dictionary, probably 30 training images per category also had positive effect on this result. It is important to note that no decrease in performance is observed although ~90% of the dictionary was pruned (768 patterns selected out of 6122), and even a slight improvement, i.e., ~1%, is achieved for Caltech-101.

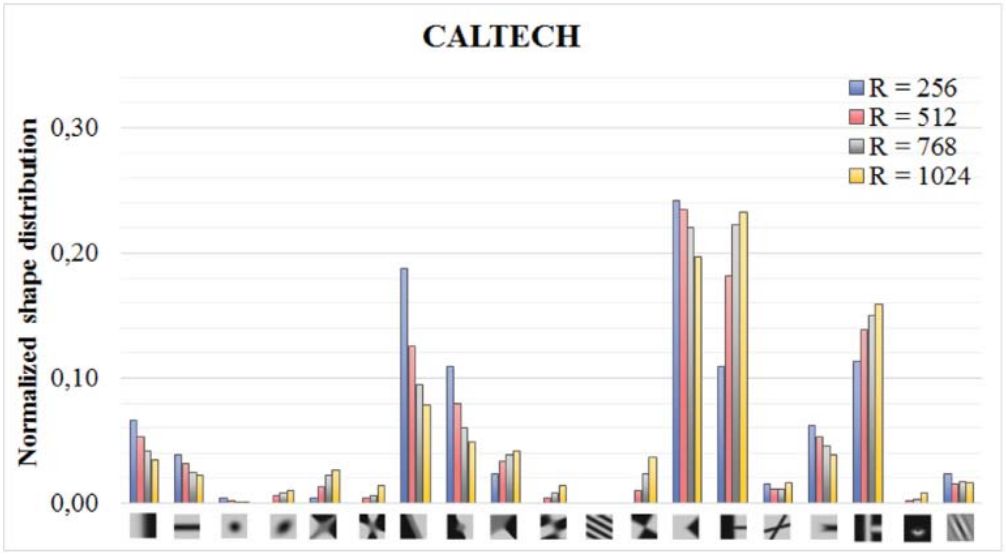


(a) R patterns are selected by MI scores computed by using COIL-100 images.

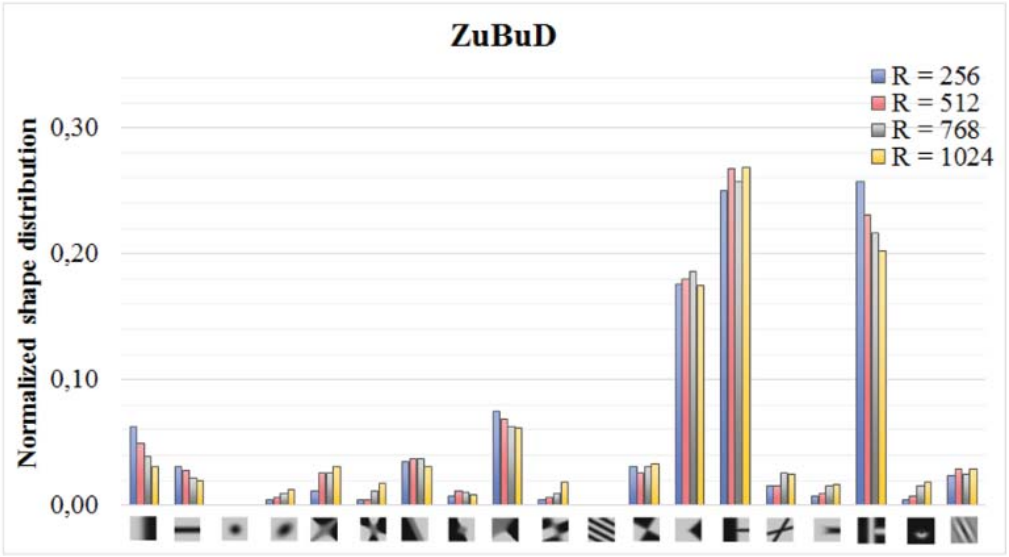


(b) R patterns are selected by MI scores computed by using ALOI-View images.

Figure 4.5. Populations of the R highest ranking shape patterns, grouped according to main shape models. R patterns are selected by MI scores computed by using the COIL-100 and ALOI-View images.

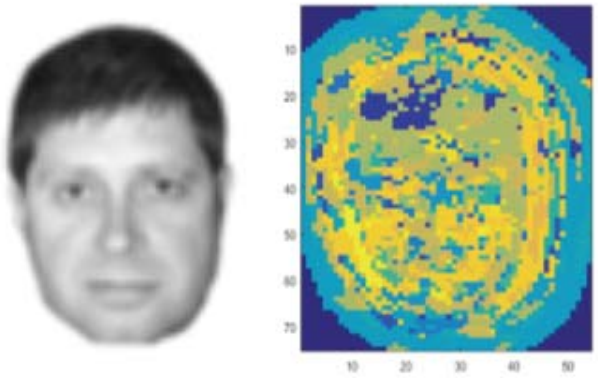


(a) R patterns are selected by MI scores computed by using Caltech-101 images.

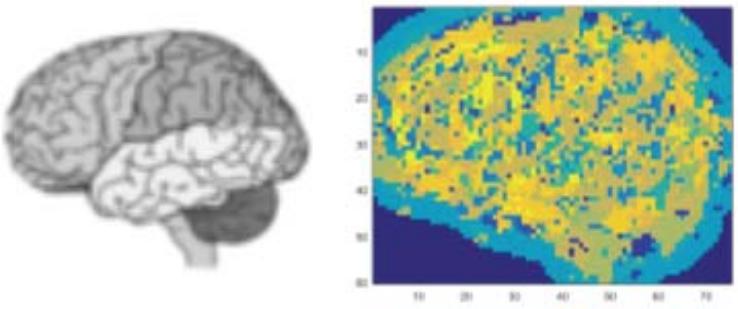


(b) R patterns are selected by MI scores computed by using ZuBuD images.

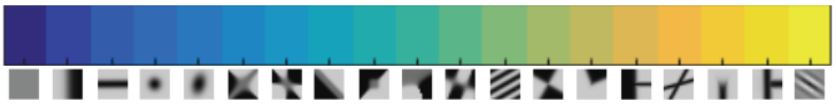
Figure 4.6. Populations of the R highest ranking shape patterns from the shape models. R patterns are selected by MI scores computed by using the Caltech-101 and ZuBuD images.



(a) Pixel label map of a randomly chosen image in terms of Flat and 19 type of models. The original image size is 75×54 pixels.

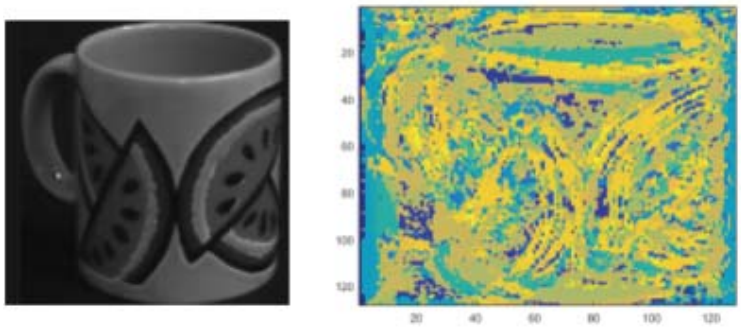


(b) Pixel label map of a randomly chosen image in terms of Flat and 19 type of models. The original image size is 60×75 pixels.

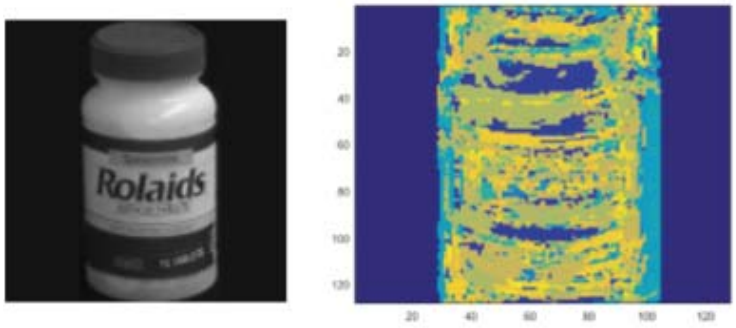


(c) Color-code labels that are used to indicate the corresponding shape models.

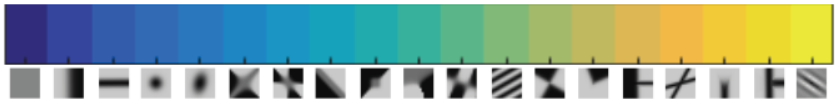
Figure 4.7. Pixel label maps of two images randomly chosen from Caltech-101 dataset.



(a) Pixel label map of a randomly chosen image in terms of Flat and 19 type of models. The original image size is 128×128 pixels.



(b) Pixel label map of a randomly chosen image in terms of Flat and 19 type of models. The original image size is 128×128 pixels.



(c) Color-code labels that are used to indicate the corresponding shape models.

Figure 4.8. Pixel label maps of two images randomly chosen from Coil-100 dataset.

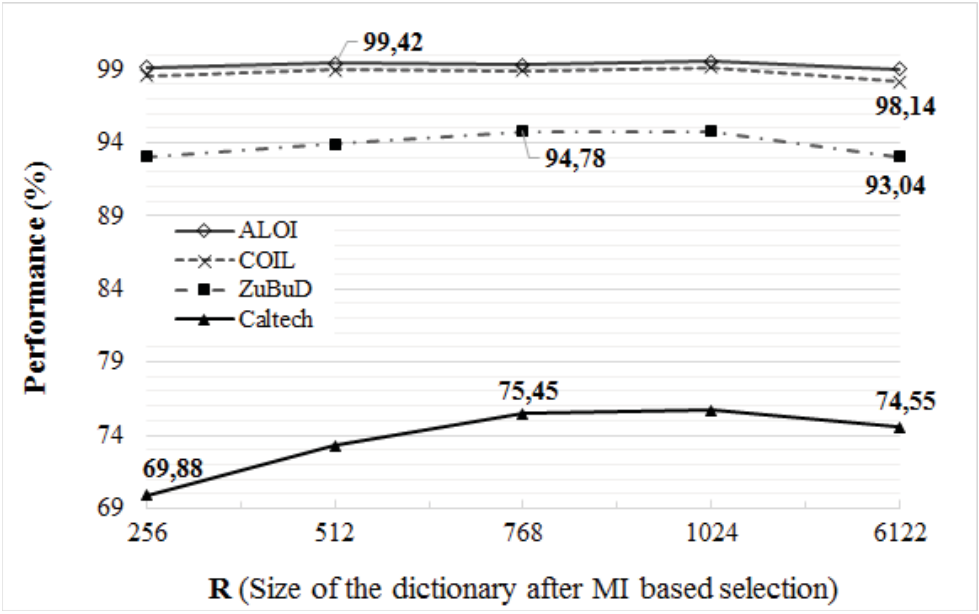


Figure 4.9. Recognition performance with dictionaries of various sizes. Dictionary atoms are selected according to their MI scores on the Caltech, COIL, ALOI, and ZuBuD images.

5. EXTRACTION OF IMAGE DESCRIPTORS

Given a shape dictionary, an image descriptor is typically a function of the expression strength or of the occurrence frequencies of the dictionary atoms. The image descriptors will be instrumental in comparing images for classification and for recognition. To this effect, one must first determine the degree of the similarity between image models and local appearances. Once the similarity scores are computed for the local appearances of images, an image level descriptor, that is a unique signature of the image is extracted. The pathway from feature extraction to image classification/recognition is illustrated in Figure 5.1. In this chapter we introduce the main components that take part in this pathway. Definitions and notations used throughout this chapter are given in Table 5.1.

5.1 Feature Extraction

To measure similarity between a test patch and one of the dictionary elements, we have used BRIEF features (Calonder et al., 2012). BRIEF features have the advantages of being binary strings, hence they are computationally efficient. Furthermore they have been shown to be robust to illumination variations and their performance is on a par with other high-performance features such as SURF (Bay et al., 2006), which are non-binary, hence computationally more complex. BRIEF features, however, are not robust to geometrical variations such as rotation and viewpoint changes. This sensitivity is somewhat mitigated by the fact that our dictionary contains shape models at several angular orientations.

BRIEF is computed by a defined test τ between a pair of points on a patch p sized $S \times S$ in Eq. 5.1. $I(p, x)$ denotes the intensity value of a patch, which is typically smoothed, p on the location of $x = (u_1, v_1)^T$ and $I(p, y)$ denotes the intensity value of the same patch on some other location $y = (u_2, v_2)^T$ (Calonder et

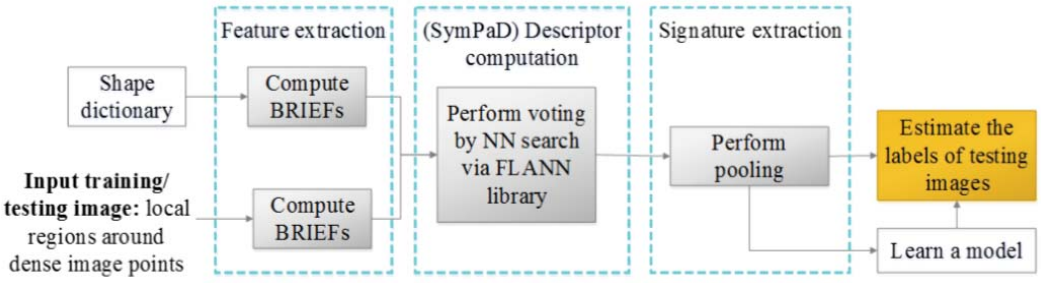


Figure 5.1. Main components that take part in the SymPaD framework

Table 5.1. Definitions and notations used throughout Chapter 5.

Symbol	Definition
p	Denote a generic image patch in pixels
$S \times S$	Size of an image patch
$I(p, x)$	The intensity value of patch p on the location of $x = (u, v)^T$
$\tau(p; x, y)$	Intensity comparison test between two image points of $x = (u1, v1)^T$ and $y = (u_2, v_2)^T$ on an image patch p
n_d	Number of pairs incorporated to intensity comparison test in the computation of BRIEF featur.
b_p	BRIEF feature vector computed on an image patch p
b_{s_i}	BRIEF feature vector computed on the SymPaD shape pattern s_i
$X_p = [\ldots]$	SymPaD vector computed on the image patch p
β	Smoothing factor used at the localized soft-voting kernel
K	Neighborhood size take place at localized soft voting computation
N	Number of points on the test image that SymPaD vector is computed for
w_i	Occurrence probability of shape pattern i on a test image
$W = [w_1, \ldots, w_D]$	Image signature
$l = 0, \ldots, L$	Spatial pyramid levels

al., 2012).

$$\tau_{(p;x,y)} := \begin{cases} 1, & \text{if } I(p,x) < I(p,y) \\ 0, & \text{otherwise} \end{cases} \tag{5.1}$$

This test is repeated for randomly chosen n_d pairs located in (x_i, y_i), $1 \leqslant i \leqslant n_d$, which results in a bit string of 0 and 1's of length n_d as the patch descriptor, In practice, the decimal conversion of this bit string is used, as in Eq. 5.2.

$$\Sigma_{1 \leqslant i \leqslant n_d} 2^{i-1} \tau(p; x_i, y_i) \tag{5.2}$$

Three factors are crucial for the computation of BRIEF features: i) the spatial arrangement of the point pairs within the patch, that is, analysis window, ii) the number of n_d of pairwise pixel comparisons, iii) the type of filtering as a preprocessing step to reduce noise sensitivity. We summarize the anaylsis related to these factors made in (Calonder et al., 2012) and our findings below:

i. Spatial arrangement. (Calonder et al., 2012) examined the following five types of spatial arrangements where the pixel pairs were sampled. Illustrations of these arrangements are presented in Figure 5.2.

- **GI.** $(X, Y) \sim$ i.i.d. Uniform$(-\frac{S}{2}, \frac{S}{2})$: The coordinates of pairs are randomly sampled from a uniform distribution.
- **GII.** $(X, Y) \sim$ i.i.d. Gaussian$(0, \frac{1}{25}S^2)$: The coordinates of pairs are randomly sampled from the Gaussian distribution with zero mean and standard deviation value which was experimentally found to be the best.
- **GIII.** $(X) \sim$ i.i.d. Gaussian$(0, \frac{1}{25}S^2)$, $(Y) \sim$ i.i.d. Gaussian$(0, \frac{1}{100}S^2)$: The x_i coordinate values are sampled from the Gaussian distribution centered in the patch origin, whereas the y_i coordinate values are sampled from the x_i centered Gaussian distribution. This geometry provides the chosen pairs to be closer to each other.
- **GIV.** (x_i, y_i) pairs are uniformly randomly sampled from the locations of a coarse polar grid.
- **GV**. $\forall i \, : \, x_i = (0, 0)^T$ are chosen on the origin and the y_i points are determined as all possible n_d values on the polar grid.

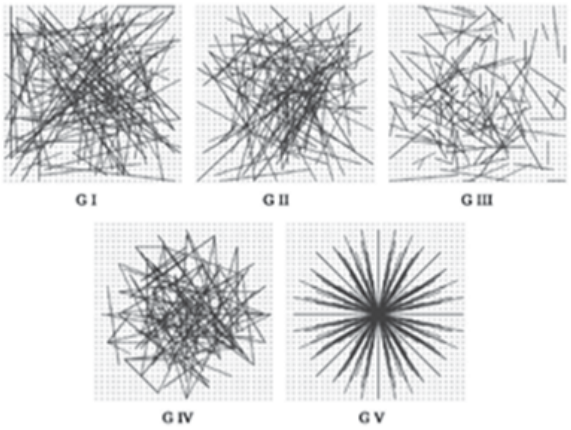


Figure 5.2. Test geometries for BRIEF. Image credits to (Calonder et al., 2012)

Table 5.2. Recognition performance results on the Caltech dataset obtained by six different test geometries. The pruned dictionary R=512 is used in the experiments.

Test geometry for BRIEF	**Performance**
GI	74.4 % $\pm$ 0.6
GII	71.9 % $\pm$ 0.5
GIII	70.9 % $\pm$ 0.6
GIII-2	70.3 % $\pm$ 0.6
GIV	70.1 % $\pm$ 0.4
GV	65.8 % $\pm$ 0.9

While (Calonder et al., 2012) recommended choosing the coordinates of the test sampling pairs from a Gaussian distribution, specifically the distribution in GII, (Galvez-Lopez and Tardos, 2011) reported better performance results by the sampling pattern generated by GIII with the distribution parameters $(X) \sim$ i.i.d. Gaussian$(0, \frac{1}{25}S^2)$, $(Y) \sim$ i.i.d. Gaussian$(0, \frac{4}{625}S^2)$. We name this sampling arrangement as **GIII-2**. We experimented the sampling patterns, GI, GII, GIII, GIV and GV, proposed by (Calonder et al., 2012) and GIII-2 proposed by (Galvez-Lopez and Tardos, 2011) for category recognition on the Caltech dataset with the pruned shape dictionary with $R = 512$ atoms. The performance results are in Table 5.2. Our experiments indicated that a uniform distribution, i.e. GI sampling pattern, works better than the others.

ii. Descriptor length. (Calonder et al., 2012) examined the descriptor length, $n_d = 128$, 256, and 512 in the context of speed, storage efficiency and recognition rate, and it is reported that $n_d = 256$ yields near optimal results. They deduced this conclusion by the $n_d = 256$ pairs sampled on a patch size of

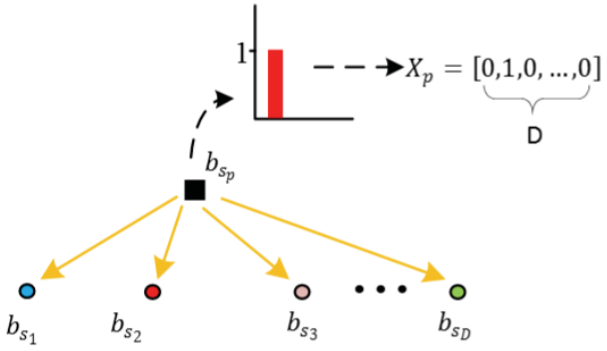


Figure 5.3. Illustration of an instance of the SymPaD vector computation for some image patch p using hard-voting.

$S = 48$ pixels. In our framework, we use patch size of $S = 15$ and it is apparent that sampling $n_d = 256$ pairs would provide sufficient information for our case, thus we used this default feature length $n_d = 256$.

iii. Smoothing filter. Reported performance results in (Calonder et al., 2012) obtained by Gaussian and Box filtering methods at the preprocessing stage were very close to each other, whereas the latter is much faster, so we similarly preferred to use Box filtering. The default parameters for box filter size is given as 9×9 for a patch sized 48×48 in (Calonder et al., 2012). In *Symbolic Patch Dictionary* (*SymPaD*), since we use patch size of $S = 15 \times 15$, we apply box filtering with size 3×3 as a preprocessing step.

5.2 Descriptor Computation

Let $H = \{0, 1\}$ and the n_d-dimensional Hamming space H^{n_d} consists of binary vectors or bit strings of length n_d. Each point $x \in H^{n_d}$ in this space is a string $x = (x_0, x_1, \ldots, x_{n_d})$ consists of 0's and 1's and the Hamming distance $d_H(x, y)$ between two given points $x, y \in H^{n_d}$ is the number of positions where these strings differ from each other, i.e., $d_H(x, y) = \sum_{i=1}^{n_d} |x_i - y_i|$. Let the illustration in Figure 5.3 represent the n_d-dimensional Hamming space, H^{n_d}. The i^{th} shape pattern, i.e., i^{th} dictionary atom, and a test patch p are represented by their corresponding n_d-dimensional BRIEF features in this space, denoted as $b_{s_i} \in H^{n_d}$, $i = 1, \ldots, D$, and $b_p \in H^{n_d}$, respectively.

As illustrated in Figure 5.3, in order to compute a descriptor vector, i.e., $X_p = [x_{(p,1)}, x_{(p,2)}, \ldots, x_{(p,D)}]$ for a test patch p, one can search its nearest neighbour among D number of shape patterns and code the position corresponding to its nearest neighbour by 1 and other positions by 0. Thus, the patch/image descriptor

$X_p = [x_{(p,1)}, x_{(p,2)}, \ldots, x_{(p,D)}]$, becomes the D-dimensional normalized histogram of occurrences of the dictionary elements computed as in Eq. 5.3 by the scheme of *hard-voting*.

$$x_{(p,i)} = \begin{cases} 1, & \text{if } i = \underset{i \in \{1,\ldots,D\}}{\operatorname{argmin}} \; d_H(b_p, b_{s_i}) \\ 0, & \text{otherwise} \end{cases} \tag{5.3}$$

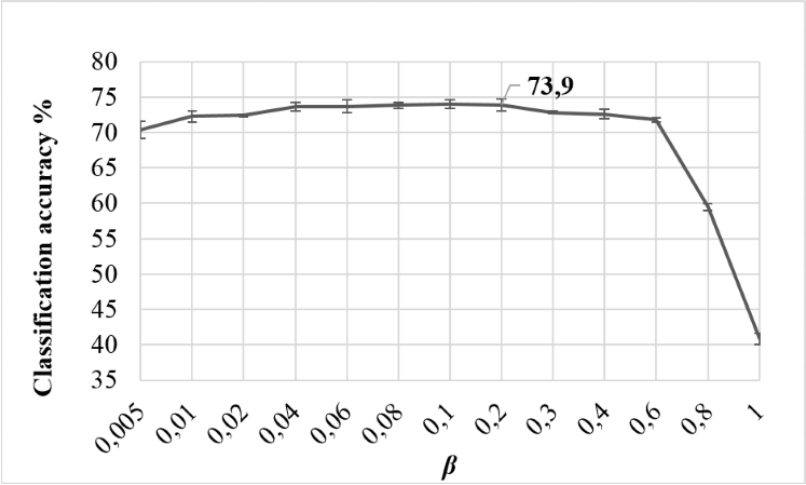
Descriptor vectors of image patches can also be encoded by multiple dictionary elements using a kernel function as in Eq. 5.4 in the *soft-voting* scheme. Note that all the D dictionary atoms are used in the formula of Eq. 5.4.

$$x_{(p,i)} = \frac{\exp(-\beta d_H(b_p, b_{s_i}))}{\sum_{i=1}^{D} \exp(-\beta d_H(b_p, b_{s_i}))} \tag{5.4}$$

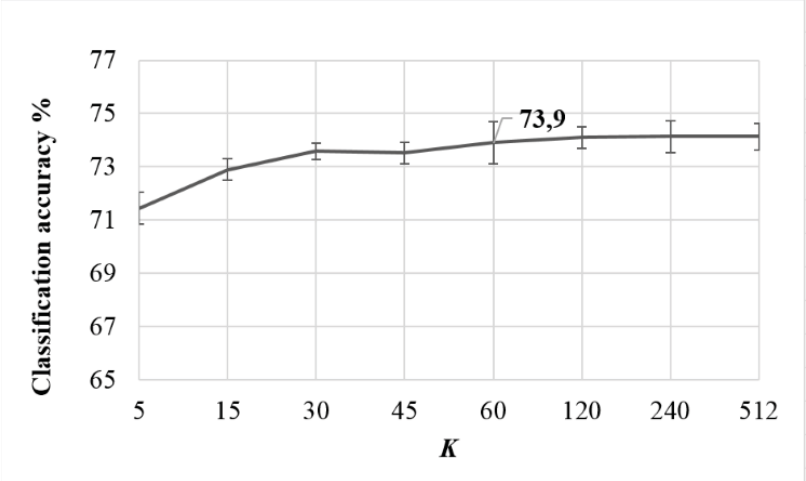
(Liu et al., 2011) argued that soft-voting is computationally more efficient than recently proposed sparse coding schemes (Yang et al., 2009; Yu et al., 2009; Wang et al., 2010), yet its main drawback is classification accuracy obtained is behind performance of sparse coding schemes. (Liu et al., 2011) improved the performance of soft voting scheme by a simple solution, i.e., *localized soft-voting*, that assigns continuous weights to a subset of shape models proportional to their similarity (Hamming distance for our case). For example, if one is to consider K closest shape models, called the *KNN* set, then a possible voting could be as in Eq. 5.5 where β denotes the smoothing factor.

$$x_{(p,i)} = \begin{cases} \frac{\exp(-\beta d(b_p, b_{s_i}))}{\sum_{j \in \mathbf{KNN}} \exp(-\beta d(b_p, b_{s_i}))}, & \text{if } i \in \text{KNN} \\ 0, & i \notin \text{KNN} \end{cases} \tag{5.5}$$

We experimented hard-voting and localized soft-voting schemes on the category recognition accuracy at Caltech dataset. The test setting that will be introduced in Chapter 6.1.3 is used. We implement the nearest neighbor linear search via FLANN library (Muja and Lowe, 2012) using Hamming distance in our experiments. Pruned dictionary of size $R = 512$ is used in these experiments. For the localized soft voting implementation, we initially investigate the impact of the parameters of β and K, on the category recognition accuracy at Caltech dataset. The classification accuracy obtained for varying β and K are presented in Figures 5.4(a) and 5.4(b). $\beta = 0.2$ and $K = 60$ was appropriate for our case.



(a) The impact of β on the classification performance at Caltech dataset for localized soft-voting scheme where K was fixed as $K = 60$.



(b) The impact of neighborhood size on the classification performance at Caltech dataset for localized soft-voting scheme where β was fixed to $\beta = 0.2$

Figure 5.4. Category recognition results at the Caltech-101 dataset when the image patches are described by localized soft-voting of SymPaD patterns.

Table 5.3. Comparison of voting methods on Caltech-101 dataset.

Method	**Performance**
Localized soft-voting (Liu et al., 2011)	74.2 ± 0.8
Localized soft-voting (**SymPaD**)	$\mathbf{73.9 \pm 0.8}$
Soft-voting (Liu et al., 2011)	72.6 ± 0.7
Soft-voting (Boureau et al., 2010a)	69.0 ± 0.8
Soft-voting (Van Gemert et al., 2008)	64.1 ± 1.2
Hard-voting (**SymPaD**)	$\mathbf{74.4 \pm 0.6}$
Hard-voting (Lazebnik et al., 2006)	64.6 ± 0.8
Hard-voting (Boureau et al., 2010a)	64.3 ± 0.9

The performance results we obtained by hard-voting and localized soft-voting are given in Table 5.3. Additionally, we give performance results reported in the literature using the same voting methods in Table 5.3. The highest classification accuracy we obtain by localized soft-voting followed by max-pooling (max-pooling is recommended for soft-voting schemes in the literature), is 73.9% $\pm$ 0.8 whereas we obtain performance of 74.4% $\pm$ 0.6 by hard-voting followed by average-pooling. (Boureau et al., 2010a) reported that soft-voting brings 4% performance gain over hard-voting under the same experiment settings. However, we observe from Table 5.3 that depending on the experimental settings, i.e., length of the dicitonary used, type of the classifier used, etc., the performance results obtained by the same method differ remarkably, for example (Liu et al., 2011) reported the performance of soft-voting as 72.6% $\pm$ 0.7, while (Van Gemert et al., 2008) reported it as 64.1% $\pm$ 1.2. Actually, although soft-voting is generally accepted as a more effective method than hard-voting, (Van Gemert et al., 2010) demonstrated that the performances of these methods highly depend on the dimension of the feature and the size of the dictionary used. Lower dimensional features yield to lower codeword ambiguity where performances of hard and soft-voting is reported as close to each other in (Van Gemert et al., 2010). For the same dimensionality of features used, performances of hard and soft voting are very close to each other and converge when the size of the dictionary is gradually increased to some particular values. However, performance of hard-voting degrades when size of the dictionary proceeds to increase to higher and higher values, whereas the performance of soft voting does not get affected significantly. For the dictionary size of $R = 512$ and feature length of BRIEF in $n_d = 256$, we found that hard-voting slightly outperforms localized soft-voting as presented in Table 5.3. Thus, we decided to proceed by the hard-voting scheme.

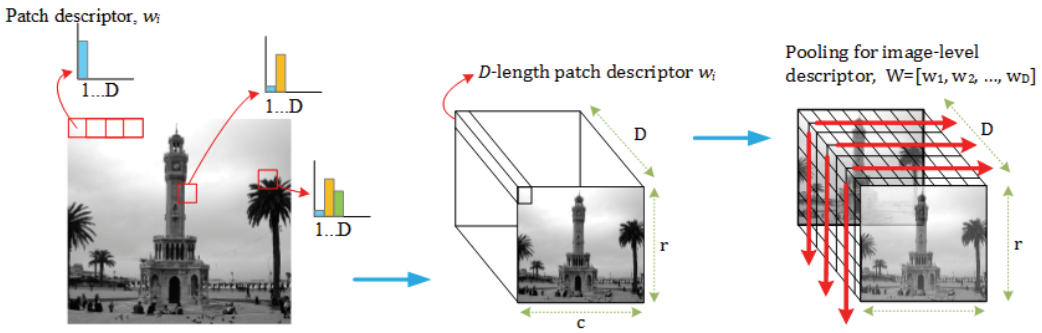


Figure 5.5. From patch descriptors to image signature.

5.3 Image Signature Extraction

Once similarity decisions are computed for all of the $p = 1, \ldots, N$ test points on the image, the image signature $W = [w_1, w_2, \ldots, w_D]$ is computed by average pooling over the $X_p = [x_{(p,1)}, x_{(p,2)}, \ldots, x_{(p,D)}]$ patch descriptors, as in Eq. 5.6. In our experiments, we worked with element-wise square rooted image signatures, since it yielded a modest performance improvement. The process is illustrated in Figure 5.5.

$$w_i = \frac{\Sigma_{p=1}^{N} x(p,i)}{N}, \; i =, \ldots, D \tag{5.6}$$

We preferred sum pooling since it is more appropriate to be used in conjunction with hard-voting coding scheme, i.e., max pooling of hard-voted descriptors yields to less discriminative image signatures.

We also tried a version where some spatial information of the occurrence locations of dictionary atoms was taken into account via *Spatial Pyramid Matching* (SPM) (Lazebnik et al., 2006). According to this method, 2D image space is divided into a sequence of grids at resolutions $l = 0, \ldots, L$, such that the grid has for a total of 2^{2l} cells at level l. After computing descriptor vectors at each cell at all resolution levels, they are appropriately weighted and concatenated in a long vector, to form the image signature. A toy example in Figure 5.6 demonstrates how the discrimination power of the image signatures improves by incorporating the spatial layout of features into the final descriptor. In this example, the image is assumed to contain three types of features. We see that although the probability distribution of the three feature types is equal in the whole image, their occurrence probabilities differ from each other in the subimages, i.e., smaller cells.

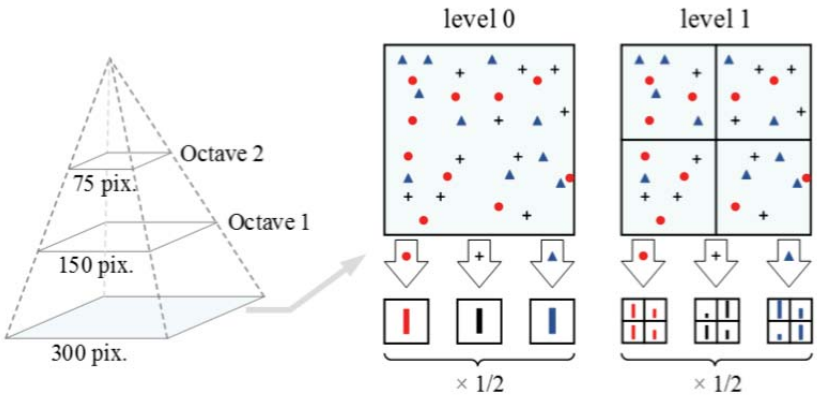


Figure 5.6. A toy example for 2-level spatial-pyramid constructed for a single image scale. There are three types of features in the image, i.e., circles, crosses and triangles. Level 1 is obtained by dividing the image at Level 0 into four cells. The colored bars below demonstrate the density of each feature type found in each cell of the grid. Each spatial histogram is weighted by ½ to obtain the image signature as adopted in the original paper (Lazebnik et al., 2006).

We applied the same scheme of normalization as in (Lazebnik et al., 2006), i.e., we divided the number of occurrences of dictionary atoms in each cell by their total occurrences in the whole image. These normalized histograms of the 2^{2l} cells computed for level l are then concatenated to obtain the spatial histogram for level l. Then, the level $l = 0$ and the higher levels $l = 1, \ldots, L-1$ spatial histograms are weighted as $1/2^L$ and $1/2^{L-l+1}$, respectively (Lazebnik et al., 2006). This weighting scheme provides the higher levels to be weighted more. Finally, spatial histograms of all levels are concatenated to obtain the image signature with dimensionality $D\Sigma_{l=0}^{L} 4^l$, where D is the dictionary size. Thus for example in Table 6.3 (to be presented in Chapter 6), one has signature size D if $L = 0$ is used and if $L = 0$ and $L = 1$ are both used, the signature dimension becomes $5D$, and $L = 2$ denotes SPM applied at levels $l = 0$, $l = 1$, and $l = 2$ so that image signatures have length $21R$.

5.4 Class/Category Recognition

Once the signatures are computed on training and testing images, we feed them to a classifier in order to recognize the class/category labels of the testing images. The general block diagram of the procedure is given in Figure 5.7.

We use a linear SVM classifier using *One-vs-One* (*OVO*) approach. SVMs are two-class classifiers and two methods are used to accomplish multiclass classification with SVM: (i) *One-vs-All* (*OVA*), (ii) One-vs-One (Galar et al., 2011).

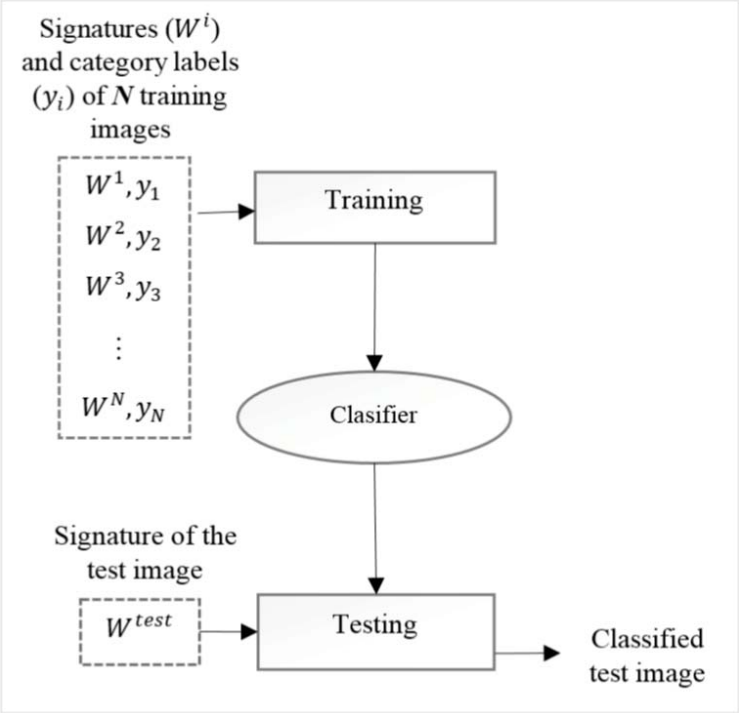


Figure 5.7. A general block diagram for classification of a test image.

In OVA method, for a C class problem, C number of binary classifiers are trained by considering samples of a single class as positives and samples of all remaining classes as negatives. The decision for class of a test sample is given regarding to the output of the C decision functions, that is the one maximizes the decision function output. OVA method has been widely used, however a comparative analysis made in (Hsu and Lin, 2002; Milgram et al., 2006) and reported that One-vs-One method has some superiority in recognition performance.

In OVO method (also known as pairwise method), $C \times (C-1)/2$ number of classifiers are trained by samples of every class pairs, by considering samples of one as positives, and samples of other as negatives. There are various methods to give the final decision. We followed the *Max Wins* voting strategy, that when a class label is assigned to a testing sample by a classifier the vote of that class increases by 1. The class with maximum vote determines the label of the testing sample.

6. EXPERIMENTS

In this chapter we present evaluation of SymPaD performance for three types of image understanding problems, i.e., category recognition, object recognition, and image retrieval. The benchmark datasets that we experimented on are as follows:

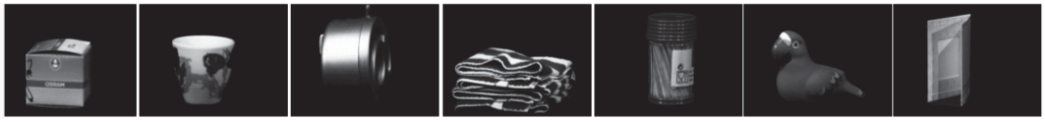
- ***Columbia Object Image Library* (*COIL*) 100** (Nene et al., 1996), contains 7200 color images of 100 objects under in-plane rotation, with a pose interval of 5 degrees, and with 128×128 resolution.
- ***Amsterdam Library Of Images* (*ALOI*) VIEW** (Geusebroek et al., 2005), is a similar dataset to COIL-100 but has a much larger extent, i.e., contains 72000 images of 1000 objects under in-plane rotation, with interval of 5 degrees.
- **Caltech-101** (Fei-Fei et al., 2006), is one of the most diverse datasets in terms of inter-class variability and it has become almost a de facto test standard for category recognition algorithms. It consists of 101 object categories each containing from 31 to 800 images.
- ***Zurich Buildings Dataset* (*ZuBuD*)** (Shao et al., 2003b), is a testbed for image retrieval that consists of 1005 color images of 201 buildings. Images of buildings are taken in five different viewpoints and they may also contain occlusions. The dataset contains an additional set of 115 query images under different imaging conditions and matching is performed against 1005 database images with 1120 images in total.

Randomly chosen example images from four benchmark datasets are presented in Figure 6.1.

We compared the performance of the proposed dictionary, i.e., SymPaD, with the following alternative dictionaries:

- *Visual dictionaries that we learnt from training images.* These experiments are accomplished to compare our model-driven method to data-driven approach of dictionary construction. We trained the dictionaries by the technique of K-SVD (Aharon et al., 2006) from given input image patches. We chose this method, since K-SVD has been succesfully applied in a variety of image processing tasks such as image denoising (Elad and Aharon, 2006; Protter and Elad, 2009), compression of facial images (Bryt and Elad, 2008),

(a) Example images to randomly chosen COIL-100 objects.



(b) Example images to randomly chosen ALOI-VIEW objects.

(c) Example images to randomly chosen Caltech-101 categories.

(d) Example images of a building in ZuBuD dataset. The first image is the query image which contains occlusion. Images of the same building in the dataset are the remaining ones.

Figure 6.1. Example images from benchmark datasets.

image and video restoration (Mairal et al., 2008b,a). We used the toolboxes of KSVD-Box, v13 and OMP-Box, v10 published in (Rubinstein R., 2009) at the implementation.

- *State-of-the-art model-driven visual dictionary construction methods.* We compared SymPaD with the existing model-driven methods, namely BIFs (Crosier and Griffin, 2010), oBIFs (Lillholm and Griffin, 2008), and BIF-columns (Crosier and Griffin, 2010) (see Appendix 1). We run the the open source codes in (Griffin et al., 2015), on the benchmark datasets with the default parameter settings mentioned in (Lillholm and Griffin, 2008; Crosier and Griffin, 2010).

- In addition, we report for comparative purposes the performance scores of recent data-driven methods from the literature.

We first provide some notes about our implementation of methods. We then present the experimental settings and preprocessing stages that we employed for each datasets. Finally, we present the performance results obtained and comparison analysis in the following sections of this chapter.

6.1 Experimental Settings

We worked with square rooted image signatures for all methods, i.e., SymPaD, K-SVD, BIFs, oBIFs, and BIF-columns, since it yielded a modest performance improvement, i.e., 1% - 2%, and we used a linear SVM classifier using OVO for all methods. In addition to performance results obtained by standard BoVW pooling, denoted by $L = 0$, that is performed on the whole image, we also present performance results obtained by employing SPM (Lazebnik et al., 2006). More details related to SPM implementation are given in Chapter 5.3.

Some arrangements related to implementations are:

i. SymPaD experiments. In the implementation of SymPaD, we discard the votes on the visual word Flat as it is also opted out in BIFs and oBIFs (Lillholm and Griffin, 2008; Crosier and Griffin, 2010). The performances are evaluated by using the four pruned dictionaries of sizes R = 256, 512, 768, 1024 constructed as in Chapter 4.

ii. K-SVD experiments. The dictionaries that we learned with K-SVD algorithm did not include any Flat patches. Thus, to run experiments under the same conditions with SymPaD, we added the Flat shape pattern into the dictionary that was learned by K-SVD and we discard its votes at the image signature extraction stage similar to we did in SymPaD framework.

iii. BIF-columns experiments. We implement SPM for BIF-columns somewhat differently than SPM (Lazebnik et al., 2006) applied to other methods, i.e., SymPaD, BIFs and oBIFs. In the standard implementation of SPM, that we applied to methods SymPaD, BIFs and oBIFs, the descriptors are computed once on the whole image (we do not compute them separately for each cell), and these descriptors are simply pooled in each cell of the spatial pyramid, then the pooled vectors in each cell are normalized and concatenated. However, we compute BIF-columns on each cell of the spatial pyramid separately, we then concatenate all to obtain the image signature. We equally weight the BIF-column vectors of the cells of level l by $1/2^{2l}$, so the sum of the concatenated vectors of level l become 1. Finally the same weighting scheme as in (Lazebnik et al., 2006) applied to concatenate vectors of different levels as explained in Chapter 5.3.

6.1.1 COIL-100 settings

We converted colored COIL-100 images to gray-scale at the preprocessing stage. For a more exhaustive analysis, we followed three experimental setups that have been used in the literature:

i. **SETUP**$_1$. We use 9-fold cross-validation as in (Hamsici and Martinez, 2009; Jayasumana et al., 2013). Images in each object class are randomly divided into 9 groups, and in each iteration, one of the groups is used for testing, and the remaining ones are used for training. The average and standard deviation of the nine recognition rates are given in the final result.

ii. **SETUP**$_2$. We regularly chose 8 images with 45 degree viewing angle between for each object to make the training set, the remaining 64 images of each object are incorporated into the testing set as in (Yang et al., 2000; Marée et al., 2004; Naik and Murthy, 2007; Obdrzalek and Matas, 2002).

iii. **SETUP**$_3$. We made a comparison with *Salient Bayes* (*SalBayes*), HMAX and SIFT methods, whose performances are reported in (Elazary and Itti, 2010). 18 images with 20 degree viewing angle between each are selected regularly for the training set and the remaining ones to testing set in SETUP$_3$ setting.

6.1.2 ALOI-VIEW settings

We followed SETUP$_2$ and SETUP$_3$ settings as in COIL-100 experiments which also have been used in the literature for ALOI-VIEW dataset. We implement experiments with SETUP$_2$ setting on the uniformly randomly sampled 250 objects of ALOI among 1000 as have been done in (Naik and Murthy, 2007). We implement experiments with SETUP$_3$ setting on images of 1000 objects as in (Elazary and Itti, 2010). We use published gray-scale ALOI-VIEW images in the quarter resolution with size 192×144 in experiments with SETUP$_2$ setting, and smoothed with a Gaussian filter with $\sigma = \sqrt{2}$ and down-sampled to half-size, i.e., 96×72, in SETUP$_3$ setting.

6.1.3 CALTECH-101 settings

A widely used experimental setup for Caltech dataset is constructing a training set by randomly chosen 15 or 30 images and a testing set by randomly chosen 50 images from the remaining ones in each category. If less than 50 images are left after

the training set in a category, all the remaining ones are used in testing. This process is repeated 10 times and the average and the standard deviation of the recognition performances are presented. We sampled 30 images from each category to construct the training set.

Many studies in the literature treated image classification for the Caltech-101 as a scene matching problem and used image-based representations (Bo et al., 2013; Boureau et al., 2011; Chatfield et al., 2011; He et al., 2014; Lazebnik et al., 2006; Yang et al., 2009; Wang et al., 2010; Zeiler and Fergus, 2014; Pu et al., 2016), while some others argued that for a better classification performance it is necessary to home in on the object instance since Caltech-101 images are not constrained in pose and have significant background clutter (Gu et al., 2009; Law et al., 2014; Li et al., 2010; Yang et al., 2015; Zhu et al., 2014). Motivated by the reported improvements in the recognition accuracy on the Caltech 101 dataset, we adopted the second approach and made the assumption that the object foreground has been segmented from the background in a preprocessing stage. Therefore, we first crop a rectangular *Region of Interest* (*RoI*), resize it by setting to 300 pixels the longer side of the rectangle via bilinear interpolation. We also remove any background clutter by filling the non-object parts with a constant intensity value. In our experiments, we did not include the *Faces-easy* and *Background* classes, since *Faces-easy* is redundant in the presence of the *Faces* class when foreground segmentation was employed, and the Background class is not intended for object recognition. We have converted the Caltech dataset images to gray-scale images.

Following the scheme employed by algorithms on similar shape/category recognition problems, we used a Gaussian pyramid with two scales where images were smoothed with a Gaussian filter with $\sigma = \sqrt{2}$ and down-sampled to half-size in each step. Hence, the image sizes became 150 pixels and 75 pixels on the longest side in the subsequent levels of the scale pyramid. Thus, the Gaussian pyramid consists of three layers. We also incorporated the information about spatial layout of occurrences of dictionary shapes in images of every scale of the Gaussian pyramid by applying SPM (Lazebnik et al., 2006). We used three layers in this spatial pyramid, denoted by $L = 2$, where SPM applied at levels $l = 0$, $l = 1$, and $l = 2$.

6.1.4 ZuBuD settings

We have converted the ZuBuD dataset images to gray-scale images, smoothed them with a Gaussian filter and downsampled the original resolution of 640×480 to 160×120. We follow the experimental procedure in (Deselaers et al., 2004;

Goedemé et al., 2004; Marée et al., 2007; Obdržálek and Matas, 2003; Shao et al., 2003a) and investigate the recall rate $r_Z = \frac{n_Z}{N}$, where n_z is the number of the correct matches in the first Z retrieved images and N is the number of possible correct matches. For the ZuBuD dataset, since every query image has 5 corresponding images in the dataset, one has $N = \min(Z, 5)$. We present the average precision results for $Z = 1$.

6.2 Comparison to K-SVD

In the dictionary learning technique of K-SVD, the matrix factorization problem in Eq.6.1 is solved iteratively where $X = [x_1, \ldots, x_n] \in \mathbb{R}^{p \times n}$ is the training set of p-dimensional images or image patches, $D = [d_1, \ldots, d_R]$ the matrix of dictionary atoms represented by $d_i \in \mathbb{R}^p$ and $A = [\alpha_1, \ldots, \alpha_n]$ is the matrix of decomposition coefficients represented by α_i in $\mathbb{R}^R$.

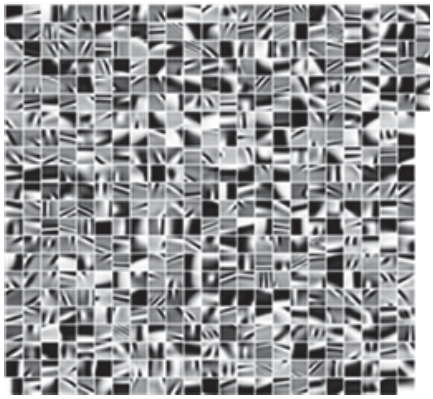
$$\min_{D,A} \frac{1}{2}\|X - DA\|_F^2 \qquad \text{subject to } \forall i, \|\alpha_i\|_0 \leqslant T_0 \tag{6.1}$$

Two stages are used in each iteration: (i) in the sparse coding stage, the sparse coefficients matrix $A = [\alpha_1, \ldots, \alpha_n]$, is computed by a pursuit matching method, i.e., *Orthogonal Matching Pursuit* (*OMP*), with the constraint that each coefficient vector have maximum T_0 nonzero elements; (ii) in the dictionary update stage, each dictionary element represented by d_i is updated sequentially to better represent the patches that use it.

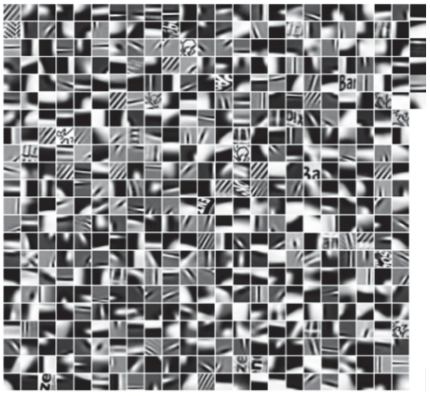
We learned four visual dictionaries with $R = 512$ atoms from $10^6/(N \times C)$ randomly sampled $p = 15 \times 15$ sized image patches from each of the training set images of four datasets. Here, N denotes the number of randomly chosen images from each category, and C denotes the number of categories in the dataset, so the dictionary is learned from $n = 10^6$ image patches for each dataset. The learned dictionaries are illustrated in Figure 6.2.

6.2.1 Distance between shape dictionaries

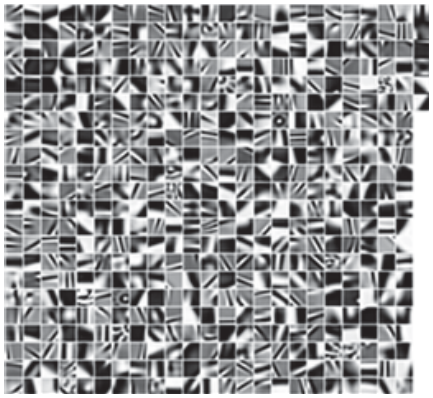
It's intriguing to investigate the degree of similarities between sets of shape patterns, i.e., dictionaries. We computed distances between visual dictionaries learned by K-SVD from four datasets and SymPaD dictionaries that are constructed and pruned with respect to four datasets. (Chevallier et al., 2014) proposed a novel criterion for dictionary assessment which is reported as robust to outliers, sensitive to



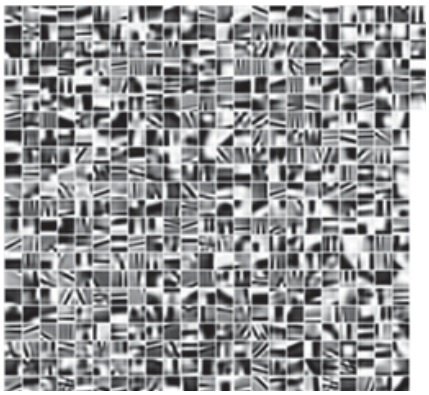
(a) Shape patterns learned from COIL-100 images



(b) Shape patterns learned from ALOI-View images



(c) Shape patterns learned from Caltech-101 images



(d) Shape patterns learned from ZuBuD images

Figure 6.2. Visual dictionary with $R = 512$ shape patterns that are learned by K-SVD from 10^6 patches of four image datasets.

	K-SVD$_{\text{COIL}}$	K-SVD$_{\text{ALOI}}$	K-SVD$_{\text{CALTECH}}$	K-SVD$_{\text{ZuBuD}}$	SymPaD dictionaries
K-SVD$_{\text{COIL}}$	10.0×10^{-15}	0.439	0.289	0.438	0.658
K-SVD$_{\text{ALOI}}$		6.0×10^{-15}	0.449	0.386	0.614
K-SVD$_{\text{CALTECH}}$			12.4×10^{-15}	0.372	0.655
K-SVD$_{\text{ZuBuD}}$				5.1×10^{-15}	0.450

Figure 6.3. Hausdorff distances between visual dictionaries.

small variations and efficient at low *Signal Noise Ratio* (*SNR*). The proposed metric is defined in two steps as follows:

1. An Euclidean-based distance, named as the *ground-metric* and denoted by d_E in Eq. 6.2, is defined between dictionary atoms;

$$d_E(\phi, \psi) = 2 \times (1 - |\langle \phi, \psi \rangle|) \tag{6.2}$$

2. A *set-metric*, e.g., Hausdorff or Wasserstein metric, is defined based on the atom-to-atom *ground-metric*. In this work, we proceed by the Hausdorff distance, denoted by d_H in Eq. 6.3, to compute distance between two dictionaries where both of the dictionaries Φ and Ψ include R atoms and assumed that each atom was L_2 normalized previously, i.e., $\Phi = \{\phi_i \in \mathbb{R}^p : \|\phi_i\|_2 = 1, i = 1, \ldots, R.$

$$d_H(\Phi, \Psi) = \max(\max_{\phi \in \Phi} \min_{\psi \in \Psi} d_E(\phi, \psi), \max_{\psi \in \Psi} \min_{\phi \in \Phi} d_E(\phi, \psi)) \tag{6.3}$$

The Hausdorff distance matrix between the mentioned visual dictionaries is presented in Fig. 6.3. It is noted that, among the learned dictionaries via K-SVD from the four datasets, the most similar one to the pruned SymPaD dictionaries (that are illustrated in Appendix 5 to 8) is the one learned from the ZuBuD dataset according to Fig. 6.3. We also observe in Fig. 6.2 that learned dictionary from ZuBuD includes more generic shape patterns, whereas learned dictionaries from other datasets include more category-specific patterns, such as car tyre, or text parts.

We compared the SymPaD dictionaries vis-a-vis data-driven (K-SVD) dictionaries. Bag-of-Words pooling with $L = 0$ (1×1), i.e., without SPM, is employed in these experiments; the image signatures have length $R = 512$. The performance results are presented in Table 6.1. We observe that SymPaD dictionaries outperform

Table 6.1. Performance comparison to visual dictionaries learned by K-SVD. Standard Bag-of-Words pooling, i.e., $L = 0(1 \times 1)$ is employed, The length of all dictionaries is $R = 512$. SymPaD dictionaries are pruned according to corresponding dataset images as in Chapter 4.

	Performance	
Dataset and the experimental setting	**Dictionary learned by K-SVD**	**SymPaD model-based dictionary**
COIL-100, SETUP_1	$99.6\% \pm 0.2$	**99.9 % ± 0.1**
COIL-100, SETUP_2	92.3 %	**95.5%**
COIL-100, SETUP_3	97.9 %	**99.2%**
ALOI-VIEW, SETUP_2 (250 objects)	96.5%	**97.9 %**
CALTECH-101 (75 pix. images)	$66.2\% \pm 0.5$	**74.42% ± 0.56**
ZuBuD	93.0%	**94.78%**

the dictionaries learned by K-SVD in all experiments. Especially, in the category recognition experiment on the challenging Caltech dataset, SymPaD dictionary outperformed K-SVD by ~8%.

We computed Hamming distance based similarity matrices calculated from BRIEF features of shape patterns in order to show the degree of redundancy inherent in these dictionaries. The shape atoms are ordered according to the following scheme: We compute the mutual BRIEF distances of shape atoms. For each atom, we rank its distance to the $(R-1)$ other atoms, and sum the distance scores of the R_D most distant atoms ($R_D = 50$ in our case). Heat maps of these similarity matrices are presented in Appendix 6. The larger these sum distances are overall the atoms, the less redundant is the dictionary. We also conjecture that more redundant dictionary results in a less discriminative description. We observe that the dictionary learned by K-SVD from the Caltech dataset is more redundant than the pruned SymPaD dictionary.

6.3 Comparison to BIFs, OBIFs And BIF-Columns

The implementational details about BIFs, oBIFs and BIF-columns are given in Appendix 1. Briefly, BIFs technique uses a visual dictionary composed of seven shape prototypes, i.e., *flat*, *ramp*, *circular mesa*/*basin*, *valley*, *ridge*, and *saddle*. By quantizing the orientation ranges of BIFs with angular step size of $\pi/4$, i.e., ramp into eight quanta, valley/ridge and saddle into four quanta, (Lillholm and Griffin, 2008) defined a new bag of words scheme with 23 visual words rather than 7. This set of shape prototypes are named as oBIFs (oriented BIFs). Since the BoW-style

descriptors by seven shape types yields to a quite coarse description, (Crosier and Griffin, 2010) proposed to use BIF-columns representation, which considers co-occurrences of BIFs in multiscale. BIF-Column representation is simply a histogram descriptor with $6^4 = 1296$ bins.

The object and category recognition and image retrieval performance results obtained by the methods of SymPaD, BIFs, oBIFs and BIF-columns are given in Tables 6.2 to 6.6. We applied the mentioned methods at the single scale of images of COIL-100, ALOI-VIEW and ZuBuD, whereas we worked in multiple scales of images of challenging Caltech dataset. In addition to performances obtained for each scale of Caltech images, we also present performance result of concatenated image signatures of all scales for Caltech dataset in Table 6.5.

We outperform BIFs, oBIFs and BIF-columns in all experiments, probably that is because we use a larger variety of shape models. The important outcomes of these experiments can be listed as:

- In the experiments on COIL-100 dataset, we obtain 100% performance with the selected $R = 768$ and $R = 1024$ dictionary atoms in the standard bag-of-words scheme, that is $L = 0$ for SETUP$_1$ configuration. SPM with $L = 1$ (2×2) improved performance ~3% for the smallest sized dictionary, i.e., $R = 256$, with regard to $L = 0$ for SETUP$_2$. Performance results was beyond 99% with $L = 0$ and $L = 1$ schemes for SETUP$_3$.

- In the experiments on ALOI-VIEW dataset, the use of SPM ($L = 1$) improves performance by ~1% when smaller sized dictionaries are used, however, for larger sizes of the shape dictionary, SPM is not crucial anymore.

- In the experiments on the Caltech-101 dataset, it is observed that SPM brings significant performance gain similarly as reported in the literature, i.e. ~12% for SymPaD, ~15% for BIFs, oBIFs and BIF-columns. For the SymPad case we see two opposite tendencies: In the absence of any spatial pyramid, the performance improves as lower resolution images are used. In the presence of spatial pyramid ($L = 1$ and $L = 2$), the performance is better with higher resolution images.

- *Joint use of multiscale and SPM:* For the multiscale performance of SymPaD on the Caltech dataset presented in Table 6.5, we concatenated the signatures of each scale of with equal weights of $1/3$ (since there are 3 scales) for the L=0, L=1, and L=2 schemes, into a long vector, in length of $3R$, $15R$, and

Table 6.2. Performance comparison of *SymPaD* to *BIFs*, *oBIFs* and *BIF-columns* for object recognition on **COIL-100** dataset in three experimental setups. $L = 0$ denotes standard BoVW scheme, $L = 1$ denotes SPM applied at levels $l = 0$ and $l = 1$.

Method	**R**	**SETUP$_1$**	**SETUP$_2$**		**SETUP$_3$**	
		$L = 0(1 \times 1)$	$L = 0(1 \times 1)$	$L = 1(2 \times 2)$	$L = 0(1 \times 1)$	$L = 1(2 \times 2)$
SymPaD	256	$99.8\% \pm 0.2$	93.4%	96.4%	98.9%	99.6%
	512	$99.9\% \pm 0.1$	**99.5%**	97.1%	99.2%	99.7%
	768	**100%**	95.8%	**97.2%**	**99.5%**	**99.9%**
	1024	**100%**	96.0%	**97.2%**	**99.5%**	**99.9%**
BIFs	6	$58.6\% \pm 1.5$	49.9%	77.5%	57.1%	85.9%
oBIFs	22	$94.8\% \pm 0.7$	81.3%	89.3%	91.9%	97.7%
BIF-columns	1296	$99.4\% \pm 0.3$	92.9%	95.8%	97.7%	98.9%

Table 6.3. Performance comparison of *SymPaD* to *BIFs*, *oBIFs* and *BIF-columns* for object recognition on **ALOI-VIEW** dataset in two experimental setups. $L = 0$ denotes standard BoVW scheme, $L = 1$ denotes SPM applied at levels $l = 0$ and $l = 1$.

Method	**R**	**SETUP$_2$ (250 categories)**		**SETUP$_3$ (1000 categories)**	
		L=0(1×1)	L=1(2×2)	L=0(1×1)	L=1(2×2)
SymPaD	256	97.1%	98.1%	98.8%	99.2%
	512	97.9%	98.4%	99.2%	99.6%
	768	97.9%	98.5%	**99.2%**	99.8%
	1024	**98.1%**	**98.6%**	**99.2%**	**99.9%**
BIFs	6	50.0%	84.7%	44.2%	87.4%
oBIFs	22	88.2%	95.5%	93.0%	97.7%
BIF-columns	1296	91.6%	94.6%	96.1%	97.9%

$63R$ respectively. The performance improves significantly, i.e., ~8%, for L=0 scheme with the aid of multiscale; however, when spatial pyramid scheme is used in conjunction with multiscaling, the performance gain we obtain becomes marginal, i.e., ~1%.

- In the experiments on the ZuBuD dataset, we observe that SPM implementation does not make difference in the performance, yet we outperform BIF-columns by ~4% with image signatures in almost its half-length, i.e., $R = 512$ vs. $R = 1296$.

Table 6.4. Comparison of *SymPaD* to *BIFs*, *oBIFs* and *BIF-columns* for category recognition on **Caltech-101** dataset under different scales and for different spatial pyramids. $L = 0$ denotes standard BoVW scheme, $L = 1$ denotes SPM applied at levels $l = 0$ and $l = 1$, $L = 2$ denotes SPM applied at levels $l = 0$, $l = 1$ and $l = 2$

Method	**R**	**Resolution** (size on the longer side)	**L=0**(1×1)	**L=1**(2×2)	**L=2**(4×4)
SymPaD	256	300 pix	66.8 ± 1.1	80.7 ± 0.6	$\mathbf{84.7 \pm 0.5}$
		150 pix	69.1 ± 0.7	$\mathbf{80.8 \pm 0.6}$	83.9 ± 0.5
		75 pix	$\mathbf{70.5 \pm 0.5}$	78.6 ± 0.6	81.8 ± 0.7
	512	300 pix	70.8 ± 0.8	82.1 ± 0.8	$\mathbf{85.1 \pm 0.6}$
		150 pix	73.0 ± 0.8	$\mathbf{82.5 \pm 0.5}$	84.5 ± 0.7
		75 pix	$\mathbf{74.4 \pm 0.6}$	81.0 ± 0.7	82.4 ± 0.3
	768	300 pix	72.5 ± 0.7	82.4 ± 0.7	$\mathbf{85.3 \pm 0.6}$
		150 pix	74.7 ± 1.0	$\mathbf{83.4 \pm 0.6}$	84.7 ± 0.8
		75 pix	$\mathbf{76.4 \pm 0.6}$	81.3 ± 0.4	83.2 ± 0.3
	1024	300 pix	73.4 ± 0.7	82.5 ± 0.8	$\mathbf{85.6 \pm 0.7}$
		150 pix	75.2 ± 0.7	$\mathbf{82.9 \pm 0.5}$	84.4 ± 0.8
		75 pix	$\mathbf{76.5 \pm 0.6}$	81.7 ± 0.5	82.8 ± 0.6
BIFs	6	300 pix	23.2 ± 0.4	47.7 ± 0.8	66.2 ± 0.7
		150 pix	24.1 ± 0.5	48.6 ± 1.1	67.1 ± 0.8
		75 pix	20.8 ± 0.6	45.4 ± 0.8	64.9 ± 0.4
oBIFs	22	300 pix	49.4 ± 0.6	72.0 ± 0.8	80.6 ± 0.4
		150 pix	49.1 ± 0.7	72.8 ± 0.8	81.8 ± 0.9
		75 pix	48.2 ± 1.2	72.1 ± 0.3	80.9 ± 0.6
BIF-columns	1296	300 pix	45.2 ± 0.7	60.8 ± 0.8	69.8 ± 1.0
		150 pix	45.2 ± 0.6	62.4 ± 0.4	70.5 ± 0.4
		75 pix	45.2 ± 0.9	61.9 ± 0.8	69.5 ± 0.9

Table 6.5. Category recognition performance results of SymPaD when spatial histograms are concatenated over scale on **Caltech-101** dataset. Best performance results obtained in single scale for each R are taken from Table6.4

Method	**R**	**L=0**(1×1)		**L=1**(2×2)		**L=2**(4×4)	
		single scale	scale concat.	single scale	scale concat.	single scale	scale concat.
SymPaD	256	70.5 ± 0.5	78.6 ± 0.8	80.8 ± 0.6	85.0 ± 0.6	84.7 ± 0.5	86.3 ± 0.2
	512	74.4 ± 0.6	81.2 ± 0.5	82.5 ± 0.5	85.9 ± 0.5	85.1 ± 0.6	86.8 ± 0.2
	768	76.4 ± 0.6	82.3 ± 0.9	83.4 ± 0.6	86.6 ± 0.6	85.3 ± 0.6	86.9 ± 0.8
	1024	76.5 ± 0.6	82.4 ± 0.6	82.9 ± 0.5	86.2 ± 0.4	85.6 ± 0.7	$\mathbf{87.1 \pm 0.4}$

Table 6.6. Performance comparison of *SymPaD* to *BIFs*, *oBIFs* and *BIF-columns* for image retrieval on **ZuBuD** dataset. $L = 0$ denotes standard BoW scheme, $L = 1$ denotes SPM applied at levels $l = 0$ and $l = 1$.

Method	**R**	**L=0** (1×1)	**L=1** (2×2)
SymPaD	256	93.0 %	**94.8** %
	512	**94.8** %	94.8 %
	768	94.8 %	94.8 %
	1024	94.8 %	94.8 %
BIFs	6	23.5	64.4
oBIFs	22	76.5	89.6
BIF-columns	1296	90.4	89.6

6.4 Comparisons with the State-of-the-Art

6.4.1 Object recognition on COIL-100

Comparison of our results with the performance results of data-driven approaches on COIL-100 dataset from the literature is given in Table 6.7. We outperformed existing literature methods of *Rotation Invariant Kernels* (RIK) (Hamsici and Martinez, 2009) and Manifold Kernel SVM (Jayasumana et al., 2013) that are contour-based descriptors extracted from the shape contour. *Local Affine Frames* (*LAFs*) (Obdrzalek and Matas, 2002) which defines on affine covariant features, was better in SETUP_2 setting and in SETUP_3 it was equal to ours.

6.4.2 Object recognition on ALOI-VIEW

We present performance comparisons with recent works on ALOI-VIEW dataset from the literature in Table 6.8. *Multi-Colored Region Descriptor* (*M-CORD*) (Naik and Murthy, 2007), using shape and color clues of images in SETUP_2 configuration, slightly outperforms SymPaD which is based on shape clues merely. SymPaD outperforms the SIFT, HMAX and SalBayes methods (Elazary and Itti, 2010) in SETUP_3 setting significantly.

6.4.3 Category recognition on CALTECH-101

The performance comparison with data-driven approaches are presented in Table 6.9. Since we home in on the object instance, we additionally present the literature work adopting the same approach for a fair comparison.

In one of the earlier methods, SIFT features are computed around image points

Table 6.7. Performance comparison of *SymPaD* to *State-of-the-Art* methods for object recognition on **COIL-100** dataset in three experimental setup.

Method	**SETUP$_1$**	**SETUP$_2$**	**SETUP$_3$**
HMAX (Elazary and Itti, 2010)	-	-	77.0%
SIFT (Elazary and Itti, 2010)	-	-	87.2%
RIK (Hamsici and Martinez, 2009)	95.8%	-	-
Manifold Ker. SVM (Jayasumana et al., 2013)	97.0%	-	-
SNOW/intensity (Yang et al., 2000)	-	85.1%	92.3%
SNOW/edges (Yang et al., 2000)	-	89.2%	94.1%
SalBayes (Elazary and Itti, 2010)	-	-	97.2%
Extra Trees (Marée et al., 2004)	-	92.5%	99.5%
SymPaD (L=0, R=1024)	**100%**	96.0%	99.5%
SymPaD (L=1, R=1024)	-	97.2%	**99.9%**
Sub-windows (Marée et al., 2004)	-	98.5%	99.6%
M-CORD-Edge (Naik and Murthy, 2007)	-	99.0%	**99.9%**
LAFs (Obdrzalek and Matas, 2002)	-	**99.4%**	**99.9%**

Table 6.8. Performance comparison of *SymPaD* to *State-of-the-Art* methods for object recognition on **ALOI-VIEW** dataset

Method	**SETUP$_2$ (250 objects)**	**SETUP$_3$ (1000 objects)**
SIFT (Elazary and Itti, 2010)	-	71.0%
HMAX (Elazary and Itti, 2010)	-	80.8%
SalBayes (Elazary and Itti, 2010)	-	89.7 %
SymPaD (L=0, R=1024)	98.1%	99.2%
SymPaD (L=1, R=1024)	98.6%	**99.9%**
M-CORD-Edge (Naik and Murthy, 2007)	99.6%	-
M-CORD-Cluster (Naik and Murthy, 2007)	**99.7%**	-

detected by Hessian-affine detector on the foreground images of Caltech-101 dataset in (Kinnunen et al., 2011) and visual dictionary is learned by quantizing the feature space. Among the methods computing SIFT or HOG descriptors on the entire image, performance improvements are obtained by employing SPM (Lazebnik et al., 2006), soft-voting via sparse coding methods (Yang et al., 2009; Wang et al., 2010), and Fisher kernel coding by learning the dictionary with GMM clustering (Chatfield et al., 2011). More recent studies that reported state-of-the-art performance results employ deep learning architectures (Bo et al., 2013; He et al., 2014; Zeiler and Fergus, 2014). Extremely discriminative features could be learned from the data in deep learning architectures, however, they require enormous amounts of training data.

We also observe that leveraging segmentation results to object recognition provides comparative or better recognition performance than the methods dealing with the entire image region. (Gu et al., 2009) computes features of contour shape and geometric blur (Berg and Malik, 2001) on the segmented foreground and (Yang et al., 2015) utilizes a saliency detection algorithm that yield the salient regions detected on the object instance and uses SIFT features densely on these detected regions. (Gu et al., 2009) and (Yang et al., 2015) outperform the SPM implementation on the entire image region in (Lazebnik et al., 2006), without incorporating any localization information in (Gu et al., 2009) and with employing the same scheme with (Lazebnik et al., 2006) in (Yang et al., 2015), by ~9% and ~13% respectively. (Law et al., 2014) did not employ any segmentation or saliency detection algorithm but simply trivialized the unimportant regions that mostly belong to non-object regions via thresholding the densely extracted SIFT norm magnitude. (Law et al., 2014) outperformed the advanced methods of Fisher kernel (Chatfield et al., 2011) and sparse coding (Yang et al., 2009) (Wang et al., 2010) that work on the entire image region with high image representation and costly sparse coding optimization by ~1% and ~5% in performance respectively.

With the assumption that object instances were segmented properly in the preprocessing, our SymPaD method employing simply hard-voting of BRIEF features and utilizing SPM and multiscaling, performed better than the data-driven schemes using SIFT/HOG features (Lazebnik et al., 2006; Yang et al., 2009; Wang et al., 2010; Chatfield et al., 2011; Law et al., 2014; Kinnunen et al., 2011; Yang et al., 2015), dictionary learning methods of Kmeans, GMM, MI-KSVD (Lazebnik et al., 2006; Bo et al., 2013; Chatfield et al., 2011; Law et al., 2014; Kinnunen et al., 2011; Yang et al., 2015), or employing sparse coding/fisher kernel (Yang et al., 2009; Wang et al., 2010; Bo et al., 2013; Chatfield et al., 2011). We also did better than ConvNet method in (Zeiler and Fergus, 2014).

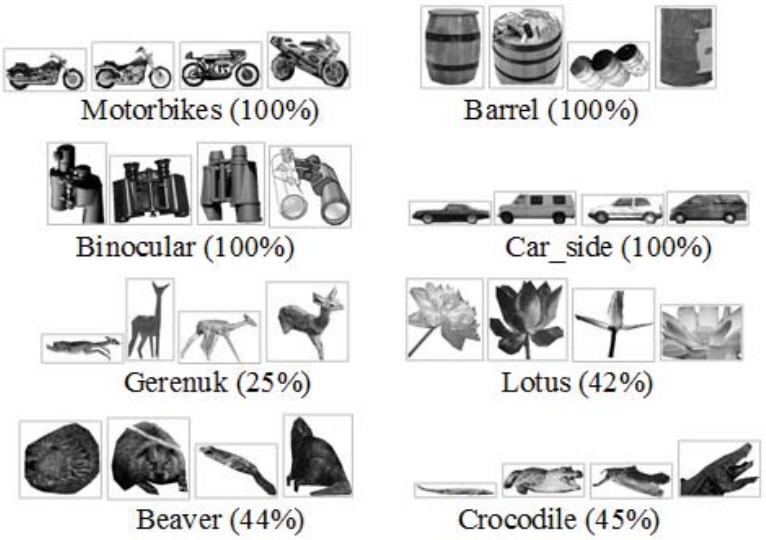


Figure 6.4. Example images from categories recognized in best and worst rates ($L = 2$, $R = 1024$, multiscale concatenation is applied).

We present the performance results from (Zhu et al., 2014) and (Li et al., 2010) which are obtained by utilizing segmented object instances via ground-truth information as in our case. We differ from (Zhu et al., 2014) in that they extract dense SIFT features, and employ costly sparse coding. By using SymPaD dictionary and hard-voting we obtain competitive performance results with (Zhu et al., 2014) and (Li et al., 2010) with even $R = 256$ dictionary atoms. Performance obtained on segmented object instance is improved by ~1% in (Li et al., 2010) over (Zhu et al., 2014) by extracting a more variety of features, i.e., appearance features that are bag of words of gray-level SIFT and color SIFT densely on the foreground images, and shape features, that are three pyramid HOGs, one computed on the contour of the foreground, and the other two operate on edges detected by globalPB inside the foreground.

The categories with best and worst performance, as recognized by the multiscale SymPaD with $R = 1024$, $L = 2$ are presented in Fig. 6.4. Our method performs well on categories with strong spatial layout priors such as *binocular*, *car_sid*e, *motorbikes*, but less successful on categories with large intra-class variation and high diversity on pose such as *gerenuk*, *lotus*, *beaver* and *crocodile*.

6.4.4 Image retrieval on ZUBUD

Performance figures from the recent literature are given in Table 6.10. We outperform (Shao et al., 2003a; Deselaers et al., 2004; Goedemé et al., 2004). We

Table 6.9. Performance comparison of *SymPaD* to *State-of-the-Art* methods for category recognition on **Caltech-101** dataset. GT denotes the methods applied on the foreground images segmented by Ground Truth segmentation masks, Sal/Seg. denotes the methods applied on the salient regions or regions detected by proposed segmentation algorithms in each.

Method	**Foreground**		**Entire image**	**Performance**
	GT	Sal/Seg		
SIFT, VQ (SOM), Hard Voting (Kinnunen et al., 2011)	√			20%
Dense SIFT, VQ (KMeans), Hard Voting, SPM (Lazebnik et al., 2006)			√	64.6%±0.8
Contour shape (Region), Geometric Blur (Point) (Gu et al., 2009)		√		73.1%
Dense SIFT, SC, SPM (Yang et al., 2009)			√	73.2%±0.5
Dense HOG, LLC, SPM (Wang et al., 2010)			√	73.4%
Dense SIFT, VQ (KMeans), SPM (Yang et al., 2015)		√		77.3%±0.6
Dense SIFT, GMM, Fisher kernel (Chatfield et al., 2011)			√	77.8%±0.6
Dense SIFT, VQ (KMeans), (Hard/Soft-voting), (Avg/Max) pooling (Law et al., 2014)		√		78.5%±1.0
MI-KSVD, SC (Bo et al., 2013)			√	82.5%±0.5
SymPaD (R=256, SPM, scale concatenation, Hard-voting, Avr-pooling)	√			86.3%±0.2
ConvNet (Zeiler and Fergus, 2014)			√	86.5%±0.5
SymPaD (R=1024, SPM, scale concatenation, Hard-voting, Avr-pooling)	√			87.1%±0.4
Dense SIFT, SC, SPM (Zhu et al., 2014)	√			88.3%
Dense SIFT, Dense color SIFT, HOG, VQ (Kmeans) (Li et al., 2010)	√			89.3%
ConvNet using Bayesian deep learning (Pu et al., 2016)			√	93.2%
ConvNet with SPM (SPP-net) (He et al., 2014)			√	93.4%±0.5

Table 6.10. Performance comparison of *SymPaD* to *State-of-the-Art* methods for image retrieval on **ZuBuD** dataset.

Method	Performance
HPAT (Shao et al., 2003a)	86.1%
Feature histograms of relational kernel, color histograms & tamura texture histograms (Deselaers et al., 2004)	89.6%
Fast wide baseline matching (Goedemé et al., 2004)	92.0%
SymPaD (L=0, R=1024)	94.8%
Random subwindows (Marée et al., 2007)	95.7%
DCT 15 coeffs (Obdržálek and Matas, 2003)	100%

obtain competitive results with (Marée et al., 2007) that use randomized tree ensembles. Computing the DCT coefficients on the local affine frames, (Obdržálek and Matas, 2003) obtained superior performance gain.

7. CONCLUSIONS

In this thesis, we have dealt with the problem of creating effective and representative visual dictionaries for image understanding applications, e.g., image category recognition and image retrieval. We have explored visual dictionaries from a model-driven perspective and we have developed a novel dictionary scheme that we call as **SymPaD** (**Sym**bolic **Pa**tch **D**ictionary). The main contributions and novelties of the thesis work can be summarized as follows.

- We have shown that useful shape models can be generated using polynomial and transcendental functions in the argument of sigmoids. These function mappings can generate gray-level surfaces of geometrical primitive shapes in light images, and 3D surfaces in depth images, such as ramps, valleys, ridges, elongated and circular mesas and summits, Gabor-like shapes, etc. To advance the current model-driven methods, this type of modelling allows the incorporation of judiciously selected higher-order intrinsic image structures into the visual dictionary.

- We have defined the shape compounding operation as a nonlinear step using min-max gray-level addition. We were inspired from the fundamental literature works that employ perceptual grouping concepts for a more discriminating image representation, i.e., (Marr, 1976; Tenenbaum and Witkin, 1983; Lowe, 1984; Saund, 1990; Horaud et al., 1990), in the course of designing these compounding models. We have observed that nonlinear combinations via min-max addition proves to be very useful; these nonlinearly combined shapes achieved highest Mutual Information scores and are seen to be predominant in the pruned dictionaries.

- We have found useful to enrich shape dictionaries with geometrical transformations such as rotations, scalings, translations and luminance reversals (video reversal). With this method, referred to as parametrization in the thesis, our dictionaries could be expanded two orders of magnitude in size.

- When parametrizing a dictionary, we have seen that the performance has different sensitivities to the parameter sampling interval. For example, the fineness of rotation intervals affects the performance much more as compared to the sampling step size of the intensity transition rate from background to foreground of primitive shapes. Rotational angle step of $\pi/4$ is commonly used in the literature (Lowe, 2004; Lillholm and Griffin, 2008). However we have

found that a finer splitting, namely $\pi/8$ radiant separation brings about a 3% performance improvement.

- We have generated a visual dictionary using all possible parametrizations, i.e., rotations, offsets from center, compounding angles, and transition rates of the shape models based and then we have pruned and ranked it based on the Mutual Information scores. Even though the dictionary was reduced to one tenth of its full size, not only we did not compromise on the performance, but we even observed a slight improvement of 1% in category recognition. Our main observation is that corners, junctions, T-junctions and ramp-like shapes are the most relevant shape patterns for image datasets.

- We have compared the performance of SymPaD dictionary vis-à-vis two other dictionary generation methods: As a representative of the data-driven methods, we chose the dictionary learning method called K-SVD (Aharon et al., 2006); as a representative of the state-of-the-art model-driven methods, we chose BIFs, oBIFs and BIF-columns (Griffin, 2007; Griffin and Lillholm, 2007; Griffin et al., 2009; Crosier and Griffin, 2010; Lillholm and Griffin, 2008). The experimental results showed that;

  - ⋆ SymPaD performs significantly better than the BIFs class of model-driven methods;
  - ⋆ SymPaD outperformed K-SVD by ~8% in category recognition and by ~1.6%±1.1 in the remaining datasets of object recognition and image retrieval when the same encoding, pooling and classification tools were used.

- Performance comparisons with recent data-driven methods for object recognition have revealed that SymPaD method employing hard-voting on binary features, pyramid pooling and multiscaling performed better than the methods using SIFT/HOG features, learning dictionary by Kmeans/GMM/MI-KSVD or employing sparse coding / Fisher kernel. On the other hand, M-CORD-Edge that benefits from colored region descriptors and methods employing deep learning architectures performed better than SymPad method on the same set of test databases.

An interesting future application area of the SymPaD approach can be employing SymPaD for classification of multispectral (or hyperspactral) images. Then, the shape dictionary should be constructed by multiband primitives. These primitive

models can be defined regarding to some prior knowledge learned from a training dataset of multiband images.

In this thesis work, we have observed that *nonlinear combinations* of a set of basic shape primitives, i.e., *ramp*, *valley*, *ridge*, *circular pit* and *hill*, via min-max addition operations provides effective image representation. This compounding scheme of models reminds us of hierarchical feature learning scheme provided in *Convolutional Neural Networks* (*CNNs*). Extensively reviewed in (Bengio, 2009), CNNs employ deep learning architectures consist of alternating convolution layers and pooling layers. Convolution layers generate feature maps by linear convolution filters followed by a nonlinear activation function, a.k.a. transfer function. Pooling layers employ subsampling by max or sum operations, so some degree of invariance to translation and rotation is achieved. Alternating such layers in a deep architecture yields to learn feature hierarchies in increased complexity and abstraction (Kavukcuoglu et al., 2010; Bengio et al., 2013; Farabet et al., 2013; Agostinelli et al., 2014; Simpson, 2015). The most abstract representations are less invariant to local changes of the input data and more succesful to detect the categories, hence they have greater discrimination power (Bengio et al., 2013). Learning extreme representative features from data, CNNs achieve superior performance gains in many machine vision applications.

In SymPaD, we have proceeded until second level of feature abstraction by compounding basic shape models. However, higher levels of abstraction can also be achieved by new compounding definitions over the existing shape models, i.e., Group I, II and III models. To do so we can facilitate from the CNNs architecture that achieves increased levels of abstraction by activation functions and pooling operations.

An activation function in CNNs determines whether and in what magnitude an input signal is allowed to progress further through the following layers of the network. The same methodology is viable in our case, i.e., we decide whether and in what magnitude input shape models are allowed to take part at the formation of the final model by the compounding functions. The foremost activation functions in the literature that have been used at CNNs are presented at Table 7.1. The performance evaluation of these nonlinear functions can be further explored in the extensive survey of (Mishkin et al., 2016).

Table 7.1. Foremost non-linear activation functions at the literature that have been used at CNNs.

Name	**Formula**
ReLU (Jarrett et al., 2009)	$f(x) = \max(x, 0)$
LReLU (Maas et al., 2013)	$f(x) = \max(x, \alpha x)$, $\alpha \approx 0.01$
maxout (Goodfellow et al., 2013)	$f(x) = \max(W_1 x + b_1, W_2 x + b_2)$
APL (Agostinelli et al., 2014)	$f(x) = \max(x, 0) + \sum_{s=1}^{S} \alpha_i^s \max(0, -x + b_i^s)$
RReLU (Xu et al., 2015)	$f(x) = \max(x, \alpha x), \alpha =$ random$(0.1, 0.5)$
PReLU (He et al., 2015)	$f(x) = \max(x, \alpha x), \alpha$ is learnable
ELU (Clevert et al., 2015)	$f(x) = x,$ if $x \geqslant 0,$ else $\alpha(e^x - 1)$

REFERENCES

Agostinelli, F., Hoffman, M., Sadowski, P. and Baldi, P., 2014. Learning activation functions to improve deep neural networks. *arXiv preprint arXiv:1412.6830*.

Aharon, M., Elad, M. and Bruckstein, A., 2006. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE transactions on signal processing*, 54(11):4311–4322.

Alahi, A., Ortiz, R. and Vandergheynst, P., 2012. Freak: Fast retina keypoint. In Computer vision and pattern recognition (CVPR), 2012 IEEE conference on. Ieee, p. 510–517.

Arthur, D. and Vassilvitskii, S., 2007. k-means++: The advantages of careful seeding. In Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms. Society for Industrial and Applied Mathematics, p. 1027–1035.

Aslan, S., Akgül, C.B., Sankur, B. and Tunalı, T., 2015. SymPaD: Symbolic Patch Descriptor. In Proceedings of the 10th International Conference on Computer Vision Theory and Applications (VISIGRAPP 2015). p. 266–271.

Barlow, H.B., 1953. Summation and inhibition in the frog's retina. *The Journal of physiology*, 119(1):69.

Battiti, R., 1994. Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on neural networks*, 5(4):537–550.

Bay, H., Tuytelaars, T. and Van Gool, L., 2006. Surf: Speeded up robust features. In European conference on computer vision. Springer, p. 404–417.

Bengio, Y., 2009. Learning Deep Architectures for AI. *Foundations and Trends® in Machine Learning*, 2(1):1–127.

Bengio, Y., Courville, A. and Vincent, P., 2013. Representation Learning: A Review and New Perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828.

Berg, A.C. and Malik, J., 2001. Geometric blur for template matching. In Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on. IEEE, volume 1, p. I–607.

Bo, L., Ren, X. and Fox, D., 2013. Multipath sparse coding using hierarchical matching pursuit. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. p. 660–667.

REFERENCES (continued)

Boureau, Y.L., Bach, F., LeCun, Y. and Ponce, J., 2010a. Learning mid-level features for recognition. In Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. IEEE, p. 2559–2566.

Boureau, Y.L., Le Roux, N., Bach, F., Ponce, J. and LeCun, Y., 2011. Ask the locals: multi-way local pooling for image recognition. In 2011 International Conference on Computer Vision. IEEE, p. 2651–2658.

Boureau, Y.L., Ponce, J. and LeCun, Y., 2010b. A theoretical analysis of feature pooling in visual recognition. In Proceedings of the 27th international conference on machine learning (ICML-10). p. 111–118.

Bryt, O. and Elad, M., 2008. Compression of facial images using the K-SVD algorithm. *Journal of Visual Communication and Image Representation*, 19(4):270–282.

Calonder, M., Lepetit, V., Ozuysal, M., Trzcinski, T., Strecha, C. and Fua, P., 2012. BRIEF: Computing a local binary descriptor very fast. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(7):1281–1298.

Candes, E., Demanet, L., Donoho, D. and Ying, L., 2006. Fast discrete curvelet transforms. *Multiscale Modeling & Simulation*, 5(3):861–899.

Cen, F., Jiang, Y., Zhang, Z., Tsui, H.T., Lau, T.K. and Xie, H., 2004. Robust registration of 3-D ultrasound images based on gabor filter and mean-shift method. In Computer Vision and Mathematical Methods in Medical and Biomedical Image Analysis, Springer, p. 304–316.

Chatfield, K., Lempitsky, V.S., Vedaldi, A. and Zisserman, A., 2011. The devil is in the details: an evaluation of recent feature encoding methods. In BMVC. volume 2, pp. 8.

Chevallier, S., Barthélemy, Q. and Atif, J., 2014. On the need for metrics in dictionary learning assessment. In 2014 22nd European Signal Processing Conference (EUSIPCO). IEEE, p. 1427–1431.

Clevert, D.A., Unterthiner, T. and Hochreiter, S., 2015. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*.

Crosier, M. and Griffin, L.D., 2010. Using basic image features for texture classification. *International Journal of Computer Vision*, 88(3):447–460.

Csurka, G., Dance, C., Fan, L., Willamowski, J. and Bray, C., 2004. Visual categorization with bags of keypoints. In Workshop on statistical learning in computer vision, ECCV. Prague, volume 1, p. 1–2.

REFERENCES (continued)

Dalal, N. and Triggs, B., 2005. Histograms of oriented gradients for human detection. In 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). IEEE, volume 1, p. 886–893.

Dantone, M., Gall, J., Leistner, C. and Van Gool, L., 2014. Body parts dependent joint regressors for human pose estimation in still images. *IEEE transactions on pattern analysis and machine intelligence*, 36(11):2131–2143.

Daugman, J.G., 1980. Two-dimensional spectral analysis of cortical receptive field profiles. *Vision research*, 20(10):847–856.

Daugman, J.G., 1985. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *JOSA A*, 2(7):1160–1169.

De la Torre, F. and Kanade, T., 2006. Discriminative cluster analysis. In Proceedings of the 23rd international conference on Machine learning. ACM, p. 241–248.

Deselaers, T., Keysers, D. and Ney, H., 2004. Classification error rate for quantitative evaluation of content-based image retrieval systems. In Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on. IEEE, volume 2, p. 505–508.

Ding, C. and Li, T., 2007. Adaptive dimension reduction using discriminant analysis and k-means clustering. In Proceedings of the 24th international conference on Machine learning. ACM, p. 521–528.

Do, M.N. and Vetterli, M., 2005. The contourlet transform: an efficient directional multiresolution image representation. *IEEE Transactions on image processing*, 14(12):2091–2106.

Dollár, P., 2014. Piotr's Computer Vision Matlab Toolbox (PMT). *h ttp://vision. ucsd. edu/ pdollar/toolbox/doc/index. html*.

Donoho, D., 1998b. Orthonormal Ridgelets and Linear Singularities Tech. Report, Dept. of Stat.

Donoho, D.L., 1998a. Fast edgelet transform and applications. *Manuscript, September*.

Dou, J. and Li, J., 2014. Modeling the background and detecting moving objects based on Sift flow. *Optik-International Journal for Light and Electron Optics*, 125(1):435–440.

REFERENCES (continued)

Dougherty, J., Kohavi, R., Sahami, M. et al., 1995. Supervised and unsupervised discretization of continuous features. In Machine learning: proceedings of the twelfth international conference. volume 12, p. 194–202.

Elad, M. and Aharon, M., 2006. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image processing*, 15(12):3736–3745.

Elazary, L. and Itti, L., 2010. A Bayesian model for efficient visual search and recognition. *Vision research*, 50(14):1338–1352.

Farabet, C., Couprie, C., Najman, L. and LeCun, Y., 2013. Learning Hierarchical Features for Scene Labeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1915–1929.

Fei-Fei, L., Fergus, R. and Perona, P., 2006. One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence*, 28(4):594–611.

Field, D.J., 1994. What is the goal of sensory coding? *Neural computation*, 6(4):559–601.

Figueras i Ventura, R., Vandergheynst, P. and Frossard, P., 2006. Low rate and flexible image coding with redundant representations. *IEEE Transactions on Image Processing*, 15(EPFL-ARTICLE-87334):726–739.

Fleuret, F., 2004. Fast binary feature selection with conditional mutual information. *Journal of Machine Learning Research*, 5(Nov):1531–1555.

Freeman, W.T. and Adelson, E.H., 1991. The design and use of steerable filters. *IEEE Transactions on Pattern analysis and machine intelligence*, 13(9):891–906.

Freund, Y., Schapire, R. and Abe, N., 1999. A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, 14(771-780):1612.

Fulkerson, B., Vedaldi, A. and Soatto, S., 2008. Localizing objects with smart dictionaries. In European Conference on Computer Vision. Springer, p. 179–192.

Galar, M., Fernández, A., Barrenechea, E., Bustince, H. and Herrera, F., 2011. An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes. *Pattern Recognition*, 44(8):1761–1776.

REFERENCES (continued)

Galvez-Lopez, D. and Tardos, J.D., 2011. Real-time loop detection with bags of binary words. In 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, p. 51–58.

Geman, D. and Koloydenko, A., 1999. Invariant statistics and coding of natural microimages. In IEEE Workshop on Statistical and Computational Theories of Vision. Citeseer.

Geusebroek, J.M., Burghouts, G.J. and Smeulders, A.W., 2005. The Amsterdam library of object images. *International Journal of Computer Vision*, 61(1):103–112.

Goedemé, T., Tuytelaars, T. and Van Gool, L., 2004. Fast wide baseline matching for visual navigation. In Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on. IEEE, volume 1, p. I–24.

Goodfellow, I.J., Warde-Farley, D., Mirza, M., Courville, A.C. and Bengio, Y., 2013. Maxout networks. *ICML (3)*, 28:1319–1327.

Griffin, L.D., 2007. The second order local-image-structure solid. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(8):1355–1366.

Griffin, L.D. and Lillholm, M., 2007. Feature category systems for 2nd order local image structure induced by natural image statistics and otherwise. In Electronic Imaging 2007. International Society for Optics and Photonics, p. 649209–649209.

Griffin, L.D., Lillholm, M., Crosier, M. and van Sande, J., 2009. Basic image features (bifs) arising from approximate symmetry type. In International Conference on Scale Space and Variational Methods in Computer Vision. Springer, p. 343–355.

Griffin, L.D. et al., 2015. Basic Image Features (BIFs) implementation. Available at: `https://github.com/GriffinLab/BIFs`. Date accessed: 10.09.2016.

Gu, C., Lim, J.J., Arbeláez, P. and Malik, J., 2009. Recognition using regions. In Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE, p. 1030–1037.

Guo, C.e., Zhu, S.C. and Wu, Y.N., 2007. Primal sketch: Integrating structure and texture. *Computer Vision and Image Understanding*, 106(1):5–19.

Hall, M.A., 1999. Correlation-based feature selection for machine learning. Ph.D. thesis, The University of Waikato.

REFERENCES (continued)

Hamsici, O.C. and Martinez, A.M., 2009. Rotation invariant kernels and their application to shape analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(11):1985–1999.

Hartline, H.K., 1938. The response of single optic nerve fibers of the vertebrate eye to illumination of the retina. *American Journal of Physiology–Legacy Content*, 121(2):400–415.

He, K., Zhang, X., Ren, S. and Sun, J., 2014. Spatial pyramid pooling in deep convolutional networks for visual recognition. In European Conference on Computer Vision. Springer, p. 346–361.

He, K., Zhang, X., Ren, S. and Sun, J., 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the IEEE International Conference on Computer Vision. p. 1026–1034.

Heiler, M. and Schnörr, C., 2005. Natural image statistics for natural image segmentation. *International Journal of Computer Vision*, 63(1):5–19.

Holte, R.C., 1993. Very simple classification rules perform well on most commonly used datasets. *Machine learning*, 11(1):63–90.

Horaud, R. and Skordas, T., 1989. Stereo correspondence through feature grouping and maximal cliques. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(11):1168–1180.

Horaud, R., Veillon, F. and Skordas, T., 1990. Finding geometric and relational structures in an image. In European Conference on Computer Vision. Springer, p. 374–384.

Hsu, C.W. and Lin, C.J., 2002. A comparison of methods for multiclass support vector machines. *IEEE transactions on Neural Networks*, 13(2):415–425.

Huang, Y., Wu, Z., Wang, L. and Tan, T., 2014. Feature coding in image classification: A comprehensive study. *IEEE transactions on pattern analysis and machine intelligence*, 36(3):493–506.

Hubel, D.H. and Wiesel, T.N., 1962. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, 160(1):106–154.

Hyvärinen, A., Karhunen, J. and Oja, E., 2004. Independent component analysis, volume 46. John Wiley & Sons.

REFERENCES (continued)

Jaccard, N., Szita, N. and Griffin, L., 2015. Segmentation of phase contrast microscopy images based on multi-scale local Basic Image Features histograms. *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, p. 1–9.

Jaccard, N., Szita, N. and Griffin, L.D., 2014. Trainable Segmentation of Phase Contrast Microscopy Images Based on Local Basic Image Features Histograms. In MIUA. p. 47–52.

Jarrett, K., Kavukcuoglu, K., Lecun, Y. et al., 2009. What is the best multi-stage architecture for object recognition? In 2009 IEEE 12th International Conference on Computer Vision. IEEE, p. 2146–2153.

Jayasumana, S., Salzmann, M., Li, H. and Harandi, M., 2013. A framework for shape analysis via hilbert space embedding. In Proceedings of the IEEE International Conference on Computer Vision. p. 1249–1256.

Jiang, L., Zhang, H. and Su, J., 2005. Learning k-nearest neighbor naive bayes for ranking. In International Conference on Advanced Data Mining and Applications. Springer, p. 175–185.

Julesz, B., 1981. Textons, the elements of texture perception, and their interactions. *Nature*, 290(5802):91–97.

Jurie, F. and Triggs, B., 2005. Creating efficient codebooks for visual recognition. In Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1. IEEE, volume 1, p. 604–610.

Kaufman, L. and Rousseeuw, P.J., 2009. Finding groups in data: an introduction to cluster analysis, volume 344. John Wiley & Sons.

Kavukcuoglu, K., Sermanet, P., Boureau, Y.L., Gregor, K., Mathieu, M. and Cun, Y.L., 2010. Learning convolutional feature hierarchies for visual recognition. In Advances in neural information processing systems. p. 1090–1098.

Kerber, R., 1992. Chimerge: Discretization of numeric attributes. In Proceedings of the tenth national conference on Artificial intelligence. Aaai Press, p. 123–128.

Kinnunen, T. et al., 2011. Bag-of-Features Approach to Unsupervised Visual Object Categorisation. *Acta Universitatis Lappeenrantaensis*.

Koekoek, R. and Swarttouw, R.F., 1996. The Askey-scheme of hypergeometric orthogonal polynomials and its q-analogue. *arXiv preprint math/9602214*.

REFERENCES (continued)

Koenderink, J.J. and van Doorn, A.J., 1987. Representation of local geometry in the visual system. *Biological cybernetics*, 55(6):367–375.

Kotsiantis, S. and Kanellopoulos, D., 2006. Discretization techniques: A recent survey. *GESTS International Transactions on Computer Science and Engineering*, 32(1):47–58.

Kuffler, S.W., 1953. Discharge patterns and functional organization of mammalian retina. *Journal of neurophysiology*, 16(1):37–68.

Kwak, N. and Choi, C.H., 2002. Input feature selection for classification problems. *IEEE Transactions on Neural Networks*, 13(1):143–159.

Law, M.T., Thome, N. and Cord, M., 2014. Bag-of-words image representation: Key ideas and further insight. In Fusion in Computer Vision, Springer, p. 29–52.

Lazebnik, S., Schmid, C. and Ponce, J., 2006. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06). IEEE, volume 2, p. 2169–2178.

Le Pennec, E. and Mallat, S., 2005. Sparse geometric image representations with bandelets. *IEEE transactions on image processing*, 14(4):423–438.

Lee, A.B., Pedersen, K.S. and Mumford, D., 2001. The complex statistics of high-contrast patches in natural images.

Lee, D.D. and Seung, H.S., 1999. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791.

Leutenegger, S., Chli, M. and Siegwart, R.Y., 2011. BRISK: Binary robust invariant scalable keypoints. In 2011 International conference on computer vision. IEEE, p. 2548–2555.

Li, F., Carreira, J. and Sminchisescu, C., 2010. Object recognition as ranking holistic figure-ground hypotheses. In Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. IEEE, p. 1712–1719.

Li, J. and Allinson, N.M., 2008. A comprehensive review of current local features for computer vision. *Neurocomputing*, 71(10):1771–1787.

Lillholm, M. and Griffin, L.D., 2008. Novel image feature alphabets for object recognition. In ICPR. Citeseer, p. 1–4.

REFERENCES (continued)

Liu, L., Wang, L. and Liu, X., 2011. In defense of soft-assignment coding. In 2011 International Conference on Computer Vision. IEEE, p. 2486–2493.

Lowe, D., 1984. Perceptual Organization and Visual Recognition. Technical Report, DTIC Document.

Lowe, D.G., 2004. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110.

Maas, A.L., Hannun, A.Y. and Ng, A.Y., 2013. Rectifier nonlinearities improve neural network acoustic models. In Proc. ICML. volume 30.

Mairal, J., Bach, F. and Ponce, J., 2014. Sparse modeling for image and vision processing. *arXiv preprint arXiv:1411.3230*.

Mairal, J., Bach, F., Ponce, J. and Sapiro, G., 2010. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11(Jan):19–60.

Mairal, J., Elad, M. and Sapiro, G., 2008a. Sparse representation for color image restoration. *IEEE Transactions on image processing*, 17(1):53–69.

Mairal, J., Ponce, J., Sapiro, G., Zisserman, A. and Bach, F.R., 2009. Supervised dictionary learning. In Advances in neural information processing systems. p. 1033–1040.

Mairal, J., Sapiro, G. and Elad, M., 2008b. Learning multiscale sparse representations for image and video restoration. *Multiscale Modeling & Simulation*, 7(1):214–241.

Marée, R., Geurts, P., Piater, J. and Wehenkel, L., 2004. A generic approach for image classification based on decision tree ensembles and local sub-windows.

Marée, R., Geurts, P. and Wehenkel, L., 2007. Content-based image retrieval by indexing random subwindows with randomized trees. In Asian Conference on Computer Vision. Springer, p. 611–620.

Marill, T. and Green, D., 1963. On the effectiveness of receptors in recognition systems. *IEEE transactions on Information Theory*, 9(1):11–17.

Marr, D., 1976. Early processing of visual information. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 275(942):483–519.

Marr, D., 1982. Vision: A computational investigation into the human representation and processing of visual information. *Henry Holt and Co., New York*.

REFERENCES (continued)

Martin, D.R., Fowlkes, C.C. and Malik, J., 2004. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE transactions on pattern analysis and machine intelligence*, 26(5):530–549.

Mikolajczyk, K. and Schmid, C., 2005. A performance evaluation of local descriptors. *IEEE transactions on pattern analysis and machine intelligence*, 27(10):1615–1630.

Milgram, J., Cheriet, M. and Sabourin, R., 2006. "One Against One" or "One Against All": Which One is Better for Handwriting Recognition with SVMs? In Tenth International Workshop on Frontiers in Handwriting Recognition. Suvisoft.

Mishkin, D., Sergievskiy, N. and Matas, J., 2016. Systematic evaluation of CNN advances on the ImageNet. *arXiv preprint arXiv:1606.02228*.

Morgan, M.J., 2011. Features and the 'primal sketch'. *Vision research*, 51(7):738–753.

Muja, M. and Lowe, D.G., 2012. Fast matching of binary features. In Computer and Robot Vision (CRV), 2012 Ninth Conference on. IEEE, p. 404–410.

Murray, N. and Perronnin, F., 2014. Generalized max pooling. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. p. 2473–2480.

Naik, S. and Murthy, C., 2007. Distinct Multicolored Region Descriptors for Object Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(29):1291–1296.

Nasrabadi, N.M. and King, R.A., 1988. Image coding using vector quantization: A review. *IEEE Transactions on communications*, 36(8):957–971.

Nene, S.A., Nayar, S.K., Murase, H. et al., 1996. Columbia object image library (COIL-20). Technical Report, Technical report CUCS-005-96.

Newell, A.J. and Griffin, L.D., 2011. Natural image character recognition using oriented basic image features. In Digital Image Computing Techniques and Applications (DICTA), 2011 International Conference on. IEEE, p. 191–196.

Newell, A.J. and Griffin, L.D., 2014. Writer identification using oriented basic image features and the delta encoding. *Pattern Recognition*, 47(6):2255–2265.

REFERENCES (continued)

Newell, A.J., Morgan, R.M., Griffin, L.D., Bull, P.A., Marshall, J.R. and Graham, G., 2012. Automated texture recognition of quartz sand grains for forensic applications. *Journal of forensic sciences*, 57(5):1285–1289.

Obdrzalek, S. and Matas, J., 2002. Object Recognition using Local Affine Frames on Distinguished Regions. In BMVC. Citeseer, volume 1, pp. 3.

Obdržálek, Š. and Matas, J., 2003. Image retrieval using local compact DCT-based representation. In Joint Pattern Recognition Symposium. Springer, p. 490–497.

Peng, H., Long, F. and Ding, C., 2005. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on pattern analysis and machine intelligence*, 27(8):1226–1238.

Perronnin, F. and Dance, C., 2007. Fisher kernels on visual vocabularies for image categorization. In 2007 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, p. 1–8.

Perronnin, F., Sánchez, J. and Mensink, T., 2010. Improving the fisher kernel for large-scale image classification. In European conference on computer vision. Springer, p. 143–156.

Pham, D.S. and Venkatesh, S., 2008. Joint learning and dictionary construction for pattern recognition. In Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on. IEEE, p. 1–8.

Pohjalainen, J., Räsänen, O. and Kadioglu, S., 2015. Feature selection methods and their combinations in high-dimensional classification of speaker likability, intelligibility and personality traits. *Computer Speech & Language*, 29(1):145–171.

Protter, M. and Elad, M., 2009. Image sequence denoising via sparse and redundant representations. *IEEE Transactions on Image Processing*, 18(1):27–35.

Pu, Y., Yuan, X., Stevens, A., Li, C. and Carin, L., 2016. A deep generative deconvolutional image model. In Proceedings of the 19th International Conference on Artificial Intelligence and Statistics. p. 741–750.

Pudil, P., Novovičová, J. and Kittler, J., 1994. Floating search methods in feature selection. *Pattern recognition letters*, 15(11):1119–1125.

Rubinstein, R., Bruckstein, A.M. and Elad, M., 2010. Dictionaries for sparse representation modeling. *Proceedings of the IEEE*, 98(6):1045–1057.

REFERENCES (continued)

Rublee, E., Rabaud, V., Konolige, K. and Bradski, G., 2011. ORB: An efficient alternative to SIFT or SURF. In 2011 International conference on computer vision. IEEE, p. 2564–2571.

Saund, E., 1990. Symbolic construction of a 2-D scale-space image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(8):817–830.

Shao, H., Svoboda, T., Tuytelaars, T. and Van Gool, L., 2003a. HPAT indexing for fast object/scene recognition based on local appearance. In International Conference on Image and Video Retrieval. Springer, p. 71–80.

Shao, H., Svoboda, T. and Van Gool, L., 2003b. Zubud-zurich buildings database for image based recognition. *Computer Vision Lab, Swiss Federal Institute of Technology, Switzerland, Tech. Rep*, 260:20.

Simpson, A.J., 2015. Abstract Learning via Demodulation in a Deep Neural Network. *arXiv preprint arXiv:1502.04042*.

Sun, J., Zheng, N.N., Tao, H. and Shum, H.Y., 2003. Image hallucination with primal sketch priors. In Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on. IEEE, volume 2, p. II–729.

Tenenbaum, J.M. and Witkin, A., 1983. On the role of structure in vision. *Human and machine vision*, p. 481–543.

Van Gemert, J.C., Geusebroek, J.M., Veenman, C.J. and Smeulders, A.W., 2008. Kernel codebooks for scene categorization. In European conference on computer vision. Springer, p. 696–709.

Van Gemert, J.C., Veenman, C.J., Smeulders, A.W. and Geusebroek, J.M., 2010. Visual word ambiguity. *IEEE transactions on pattern analysis and machine intelligence*, 32(7):1271–1283.

Varma, M. and Zisserman, A., 2005. A statistical approach to texture classification from single images. *International Journal of Computer Vision*, 62(1-2):61–81.

Varma, M. and Zisserman, A., 2007. The Maximum Response (MR) Filter Banks. Available at: `http://www.robots.ox.ac.uk/~vgg/research/texclass/filters.html`. Date accessed: 10.09.2016.

Vilnrotter, F., Nevatia, R. and Price, K.E., 1981. Structural analysis of natural textures. In 1981 Technical Symposium East. International Society for Optics and Photonics, p. 246–253.

REFERENCES (continued)

Wang, F., Ding, C.H. and Li, T., 2009. Integrated KL (K-means-Laplacian) Clustering: A New Clustering Approach by Combining Attribute Data and Pairwise Relations. In SDM. SIAM, p. 38–48.

Wang, J., Yang, J., Yu, K., Lv, F., Huang, T. and Gong, Y., 2010. Locality-constrained linear coding for image classification. In Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. IEEE, p. 3360–3367.

Whitney, A.W., 1971. A direct method of nonparametric measurement selection. *IEEE Transactions on Computers*, 100(9):1100–1103.

Winn, J., Criminisi, A. and Minka, T., 2005. Object categorization by learned universal visual dictionary. In Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1. IEEE, volume 2, p. 1800–1807.

Wright, J., Yang, A.Y., Ganesh, A., Sastry, S.S. and Ma, Y., 2009. Robust face recognition via sparse representation. *IEEE transactions on pattern analysis and machine intelligence*, 31(2):210–227.

Xu, B., Wang, N., Chen, T. and Li, M., 2015. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*.

Xu, L., Neufeld, J., Larson, B. and Schuurmans, D., 2004. Maximum margin clustering. In Advances in neural information processing systems. p. 1537–1544.

Yang, J., Yu, K., Gong, Y. and Huang, T., 2009. Linear spatial pyramid matching using sparse coding for image classification. In Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE, p. 1794–1801.

Yang, L., Hu, Q., Zhao, L. and Li, Y., 2015. Salience based hierarchical fuzzy representation for object recognition. In Image Processing (ICIP), 2015 IEEE International Conference on. IEEE, p. 4873–4877.

Yang, M., Zhang, L., Yang, J. and Zhang, D., 2010. Metaface learning for sparse representation based face recognition. In 2010 IEEE International Conference on Image Processing. IEEE, p. 1601–1604.

Yang, M.H., Roth, D. and Ahuja, N., 2000. Learning to recognize 3D objects with SNoW. In European Conference on Computer Vision. Springer, p. 439–454.

Yang, Y. and Pedersen, J.O., 1997. A comparative study on feature selection in text categorization. In ICML. volume 97, p. 412–420.

REFERENCES (continued)

Yu, K., Zhang, T. and Gong, Y., 2009. Nonlinear learning using local coordinate coding. In Advances in neural information processing systems. p. 2223–2231.

Zeiler, M.D. and Fergus, R., 2014. Visualizing and understanding convolutional networks. In European Conference on Computer Vision. Springer, p. 818–833.

Zhang, Q. and Li, B., 2010. Discriminative K-SVD for dictionary learning in face recognition. In Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. IEEE, p. 2691–2698.

Zhu, F., Jiang, Z. and Shao, L., 2014. Submodular object recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. p. 2457–2464.

Zhu, J., 2002. Image compression using wavelets and JPEG2000: a tutorial. *Electronics & Communication Engineering Journal*, 14(3):112–121.

CURRICULUM VITAE

Sinem ASLAN

Address: International Computer Institute, Izmir/TURKEY
E-mail: sinem.aslan@ege.edu.tr

Personal Information
Nationality: Turkish
Birth Place and Date: Ankara, 04.05.1980

Education
M.Sc.: 2003-2007, Ege University, International Computer Institute
B.Sc.: 1998-2002, Ankara University, Electronics Engineering Department

Foreign Languages
Turkish : First Language
English : Advanced
French: Beginner

Experience
Research Assistant, 2004 – cont.
Ege University, International Computer Institute, İzmir/Turkey

Visiting Researcher, 2014-2015
Boğaziçi University, Signal and Image Processing (BUSIM) Laboratory under the supervising of Prof. Dr. Bülent Sankur.

Translator, 2003-2004 Alkim Publishing House, Ankara/Turkey
Worked at the English to Turkish translation project of the book of ”Beginning Visual C, Wrox, 2002”

Trainee, 08.2001-09.2001
Turkish Telecommunication Company, Ankara / Turkey

Trainee, 07.2000-08.2000
SIEMENS Information Communication Network Dept. Ankara / Turkey

Projects

2012, Güngör C., Kurt M., **Aslan S.** ,Ergun S., Yeni Teknolojiler İle Bir İnsanın Yüz Modelini Ve Modele Ait Vücut Hareketlerini Yakalama, 09/UBE/001, Scientific Research Project

2009, Güngör C., **Aslan S.** , Kurt M., Abnormal Region Determination at Radiologic Images Obtained From The Same Tissue at Different Times and Abnormal Region Measurement, 07/UBE/003, Scientific Research Project

2009 Cinsdikici M., **Aslan S.**, Kurt M., Ege Üniversitesi Kampüs Otomobil Giriş-çıkış denetimi İçin Plaka Tanıma Sistemi, 07/UBE/002, Scientific Research Project.

2007, Tunalı T., Kardaş G., Boztok G., **Aslan S.**, Kablosuz Yerel Ağlarda Hassas Bant Genişliği Ölçüm Tekniklerinin Geliştirilmesi, 05/UBE/001 Scientific Research Project

Publications

Aslan S., Yamac M., and Sankur B.,, 2016. DCT-based multiscale binary descriptor robust to complex brightness changes. in European Signal Processing Conference (EUSIPCO 2016), 2016 24th. p. 1573–1577.

Aslan S., Akgül C.B., Sankur B., and Tunalı E.T, 2016. A novel study on developing a model-driven visual dictionary. in IEEE Signal Processing and Communications Applications Conference, (SIU 2016), 24th.

Aslan S., Akgül C.B., Sankur B., and Tunalı E.T, 2015. SymPaD: Symbolic Patch Descriptor. in International Conference on Computer Vision Theory and Applications, (VISAPP 2015/VISIGRAPP), Berlin, 2015 10th p.266–271.

Aslan S., Akgül C.B., and Sankur B, 2014. Symbolic feature detection for image understanding. in IST/SPIE Electronic Imaging, International Society for Optics and Photonics, San Francisco, p. 902406–902406–138.

Aslan S., Tunalı E.T, 2013. Joint compressive video coding and analysis With Hidden Markov model based weighted reconstruction. in IEEE Signal Processing and Communications Applications Conference, 2013 21st.

Aslan S., Tunalı E.T, 2013. A Comparative Study of Face Feature Metrics for a Dynamic and Self-Organised Multimedia Indexing Tool. *International Journal of Natural Engineering Sciences*, 7(3).

Aslan S., Tunalı E.T, 2012. A Comparative Study Of Compressed Sensing Video Encoding GOP Patterns For Stereo Distributed Video Coding. in IEEE Signal Processing and Communications Applications.

Karaoglan, B.; Aslan, S.; Karayer, E., 2012. The Zipfian distributions on term x document matrix. in 1st International Conference on Analysis and Applied Mathematics, AIP Conf. Proc. 1470, pp. 247 – 250.

Aslan, S.; Uzer, Y.; Isik, O.; Altun, M.; Cinsdikici, M., 2008. A Simple and Improvable Method for Face Region Extraction. in 23rd International Symposium on Computer and Information Sciences, 2008. ISCIS 2008.

Aslan S.; Tunalı T.; Cinsdikici M., 2007. A Comparative Study of Feature Metrics for Classification of Human Passport Photos. in IEEE Signal Processing and Communications Applications.

APPENDIX

Appendix 1 Basic Image Features (BIFs)

Basic Image Features (BIFs) proposed by Griffin et al. (Griffin, 2007; Griffin and Lillholm, 2007; Griffin et al., 2009; Crosier and Griffin, 2010; Lillholm and Griffin, 2008) is the most current model-driven dictionary construction method in the literature. Densely visited image pixels are categorized according to seven type of image symmetries, that are *flat*, *ramp*, *dark* / *light bar*, *dark* / *light circular blob*, and *saddle* in this technique. This set of image symmetries, i.e., local structure categories that form the visual dictionary, is named as *Basic Image Features*.

The typical initial stage of any computer vision task is representing the characteristics of local image regions by computing some measurements on them. Griffin et al. used a filterbank of filters to compute such measurements.

A Gaussian kernel of scale $\sigma \in \mathbb{R}^+$ in and is defined as in Eq. A.1 and Eq. A.2 respectively.

$$G_\sigma := \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}} \tag{A.1}$$

$$G_\sigma(x, y) := G_\sigma(x) G_\sigma(y) \tag{A.2}$$

According to Scale Space approach (ref), derivative of a rescaled image by σ can be computed by convolving that image by the derivative of Gaussian of scale σ, i.e., $I' = G'_\sigma \otimes I$, where derivatives of a Gaussian kernel are defined as in Eq. A3 and Eq. A4. The filterbank used in the computation of BIFs method consists of six filters of one 0^{th} order (G^{00}_σ), two 1^{st} order (G^{10}_σ, G^{01}_σ) and three 2^{nd} order (G^{20}_σ, G^{11}_σ, G^{02}_σ). This filterbank is illustrated in Figure A.1.

$$G^{(u)}_\sigma(x) := \frac{d^u}{dx^u} G_\sigma(u) \tag{A.3}$$

$$G^{(u,v)}_\sigma(x, y) := G^{(u)}_\sigma(x) G^{(v)}_\sigma(y),\ u, v \in Z^+ \tag{A.4}$$

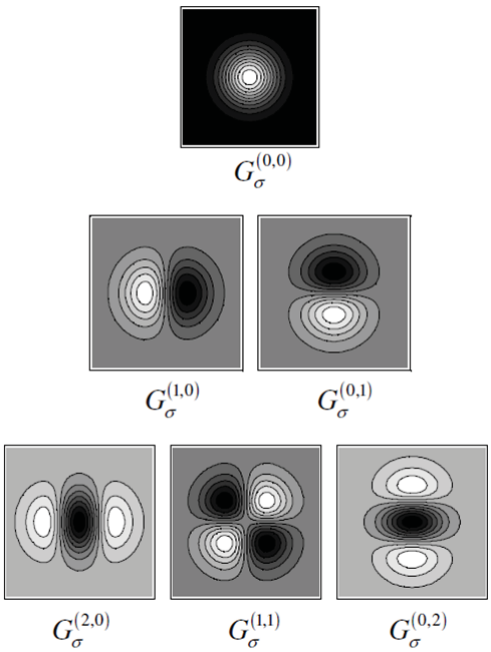


Figure A.1. The filterbank used in the BIFs technique consists of one 0^{th} order (G_σ^{00}), two 1^{st} order (G_σ^{10}, G_σ^{01}) and three 2^{nd} order (G_σ^{20}, G_σ^{11}, G_σ^{02}) Gaussian derivative filters. Image credits to (Griffin, 2007).

The vector of measurements computed by these six filters at a single image location is named as a *jet*, i.e., denoted by $j := (c_{00}\ c_{10}\ c_{01}\ c_{20}\ c_{11}\ c_{02})^T$ in (Griffin, 2007; Griffin and Lillholm, 2007; Griffin et al., 2009; Crosier and Griffin, 2010; Lillholm and Griffin, 2008) and each c_{uv} component in this vector is formalized by an inner product as in Eq. A5. Each jet vector is actually a point in the *jet space*.

$$c_{uv} := (-1)^{u+v}\langle G_\sigma^{(u,v)}|I\rangle := (-1)^{u+v}\int_{x,y\in\Re} G_\sigma^{(u,v)}(u,v)I(x,y) \tag{A.5}$$

Different changes in the imaging setup yields to changes in the scene with different importance. For example adding an object to the scene would change the intrinsic structure of the image, yet rotating the scene, or increasing the intensity levels of pixels by a constant would not affect our perception about the scene, thus these are named as extrinsic changes, and their reflections on the scene are named as extrinsic information. The shape models in the BIFs technique are defined by a parametric mapping from the initial jet space to a partitioned *orbifold*. Intrinsic aspect of the 6D jet vectors corresponds to particular locations on this orbifold, which is named 2^{nd} *order local-image-structure-solid* (Griffin, 2007; Griffin and Lillholm,

2007; Griffin et al., 2009). Thus, by defining the 2^{nd} order local-image-structure-solid, the intrinsic information of the scene could be separated from the extrinsic information which were caused by a group of transformations. The group of transformations Griffin et al. considered was including translation, rotation, addition of a constant intensity, and multiplication of intensities by a positive constant. Finally, 2^{nd} order local-image-structure-solid is partitioned into seven regions, each correspond to different image symmetries. The definitions of the shape models, i.e., the parametric mapping for partitioning the orbifold, are made by the algorithm, namely Algorithm 1 in (Crosier and Griffin, 2010), including the following steps:

1. Compute responses c_{ij} of an input image to the filterbank that is showed in Figure A.1 and apply scale normalization by $s_{ij} = \sigma^{i+j} c_{ij}$,
2. Compute λ and γ as $\gamma = s_{20} + s_{02}$, $\gamma = \sqrt{(s_{20} + s_{02})^2 + 4s_{11}^2}$,
3. Label the pixel of the input image according to the largest of $\{\varepsilon s_{00}, 2\sqrt{s_{10}^2 + s_{01}^2}, \pm\lambda, 2^{\frac{1}{2}(\gamma \pm \lambda)}\gamma\}$. ε controls the tolerance to Flat labelling, e.g., $\varepsilon = 0$ corresponds to discarding labelling with Flat.

The patch stereotypes of these image symmetries, i.e., BIFs, are illustrated in Figure A.2. In this figure, BIFs of an example image is computed in two scales of DtG filters. The pixel classes in terms of seven feature categories are demonstrated by color labels on the two different scales of the texture images at the rightmost side.

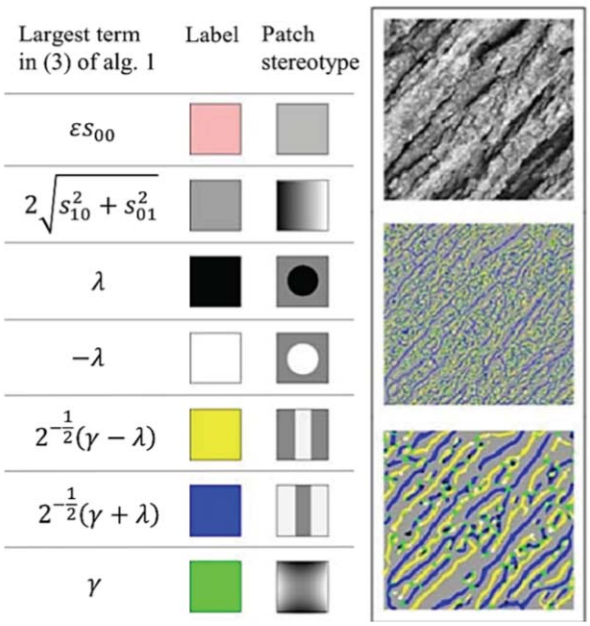


Figure A.2. Seven BIFs defined by Algorithm 1 in (Crosier and Griffin, 2010) with corresponding shape stereotypes and an example image labelled by BIFs computed at scales $\sigma = 1$ and $\sigma = 4$ both with $\varepsilon = 0$. Image credits to (Crosier and Griffin, 2010).

Since the BoW-style descriptors by seven feature types yields to a quite coarse description, (Crosier and Griffin, 2010) proposed to use *BIF-column* representation which considers co-occurences of BIFs in multiscale. A BIF-column corresponds to a stack of BIFs at the same pixel position of the image, that were computed at different scales. An example for the computation of BIF-column representation is presented in Figure A.3. Briefly, (i) BIFs are computed at four scales, i.e., $\sigma = 1$, $\sigma = 2$, $\sigma = 4$, and $\sigma = 8$, on the texture image at the lefthand side. Flat feature is discarded by setting $\varepsilon = 0$. The four scales of BIFs are illustrated in the center of the figure. (ii) Each stack of four BIF labels at the same image position is used to accomplish a hard voting-like encoding on the final histogram descriptor which has $6^4 = 1296$ bins (Flat feature is discarded, so 6 features take part in the final representation).

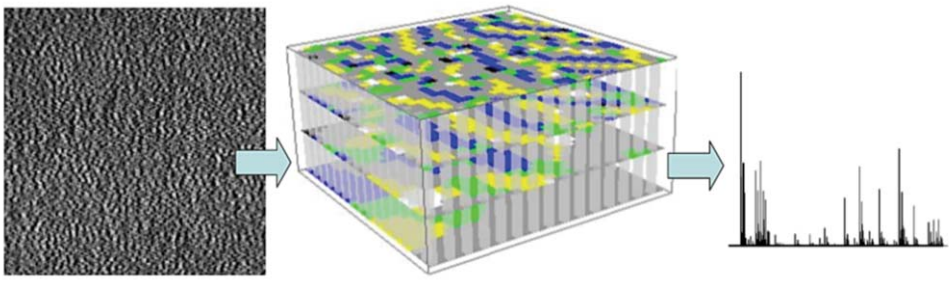


Figure A.3. BIF-columns. Left: an example input image, Centre: Seven BIFs computed at four scales, Right: BIF-columns representation. Image credits to (Griffin et al., 2009).

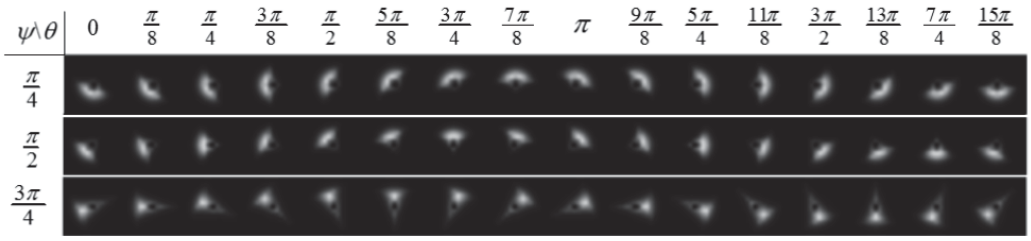
For the object categorization problem, (Lillholm and Griffin, 2008) proposed to use oBIFs (oriented BIFs). The 7 feature types are increased to 23 by quantizing orientations of ramp shape into 8 levels and dark / light line and saddle shapes into 4 levels, i.e., with angular step size of $\pi/4$. They also used oBIF-columns in object categorization problem in (Lillholm and Griffin, 2008).

Appendix 2 Duplicates of Cross shape result from quantization of rotation and compounding angle parameters in the SymPaD scheme

Row 1	θ = 0 ψ = π/4	θ = 0 ψ = π/2	θ = 0 ψ = 3π/4	θ = π/8 ψ = π/4	θ = π/8 ψ = π/2	θ = π/8 ψ = 3π/4
Row 2: Duplicates of shapes in Row 1	θ = π/4 ψ = 3π/4	θ = π/2 ψ = π/2	θ = 3π/4 ψ = π/4	θ = 3π/8 ψ = 3π/4	θ = 5π/8 ψ = π/2	θ = 7π/8 ψ = 3/4
Row 3	θ = π/4 ψ = π/4	θ = π/4 ψ = π/2	θ = 3π/8 ψ = π/4	θ = 3π/8 ψ = π/2	θ = π/2 ψ = 3π/4	θ = 5π/4 ψ = π/4
Row 4: Duplicates of shapes in Row 3	θ = π/2 ψ = 3π/4	θ = 3π/4 ψ = π/2	θ = 5π/8 ψ = 3π/4	θ = 7π/8 ψ = π/2	θ = 3π/4 ψ = 3π/4	θ = 7π/8 ψ = 3π/4

In this table, we demonstrate the duplicates of Cross shape patterns that result from the orientation quantization used in SymPaD scheme, where the rotation angle range of the bi-directional shapes, e.g. valley, is split into $q_\theta = 8$ quanta and compounding angle range into $q_\psi = 3$ quanta. θ, denotes the rotation angle and ψ denotes the compounding angle between first and second parents. The valley shapes indicated in yellow color on the cross shape figures in this table are generated by the first parent, i.e., $F_2(x, y, u = 0; \theta, \alpha)$, the valley shapes in its original (black) color are generated by the second parent, i.e., $F_2(x, y, u = 0; \theta + \psi, \alpha)$. This scheme of quantization results with vis-à-vis duplications as illustrated in the 1^{st} to 2^{nd} and 3^{rd} to 4^{th} rows of this table. To preclude such duplication of shapes, we used three quanta $q_\theta = 3$ of the rotation angle for the Cross shape. Note that one could also prefer to use different compounding angle slices to preclude such duplications, yet we preferred this solution for our case.

Appendix 3 Unintended appearances encountered in Curve shape generation when quantization in the SymPaD scheme was used



In this figure, we present Curve shape patterns generated by quantization scheme used in SymPaD, where the rotation angle range of the uni-directional shapes, e.g. curve, is split into $q_\theta = 16$ quanta and compounding angle range into $q_\psi = 3$ quanta. We observe that patterns that are not in Curve appearance are generated when the compounding angle of $\frac{3\pi}{4}$ was used. These are unintended appearances, since they look more like a shifted mesa/basin, so cause some duplications. Thus, we applied $q_\psi = 2$ quantization, in the values of $\psi = \{\frac{\pi}{4}, \frac{\pi}{2}\}$, to the compounding angle parameter for the Curve shape generation.

Appendix 4 Algorithm of ranking shape patterns regarding to Mutual Information (MI) scores

---

Input:

w_{ij}: Occurrence probability score of shape pattern i in training image j. $i = 1, \ldots, D$ and $j = 1, \ldots, J$.

Y_j: Category label of training image j, $j = 1, \ldots, J$, $Y_j \in 1, 2, \ldots, K$

R: Number of shape patterns to be selected with respect to the highest MI score.

Output:

indices of R shape patterns having highest s_i scores.

---

function RANKING_ATOMS(w_{ij}, Y_j, R)

$X_{ij} \leftarrow$ MEDIAN_QUANTIZATION(w_{ij})

for each shape pattern d **do**

compute $s_i = \text{MI}(X_i, Y) = \Sigma_{x \in \{0,1\}} \Sigma_{y \in \{1,2,\ldots,C\}} \text{Pr}[X_i = x, Y = y] \log \frac{\text{Pr}[X_i = x, Y = y]}{\text{Pr}[X_i = x]\text{Pr}[Y = y]}$

end for

sort s_i in descending order.

return indices of R atoms having highest s_i scores.

---

function MEDIAN_QUANTIZATION(w_{ij})

for each shape pattern i **do**

find median $\overline{w_i}$ of the occurrence probability scores $w_{ij} : j = 1, \ldots, J$

for each image j **do**

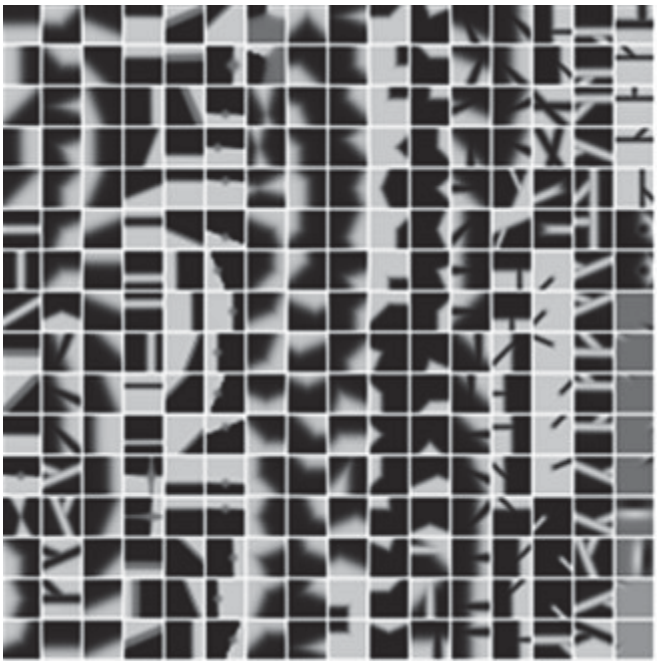
$$\text{compute quantized } X_{ij)} = \begin{cases} 1, & \text{if } w_{ij} > \overline{w_i} \\ 0, & \text{otherwise} \end{cases}$$

end for

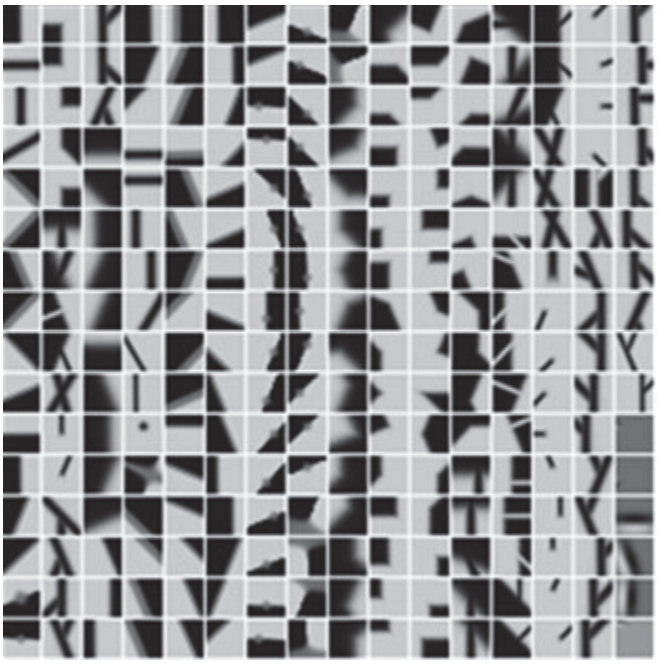
end for

return X_{ij}

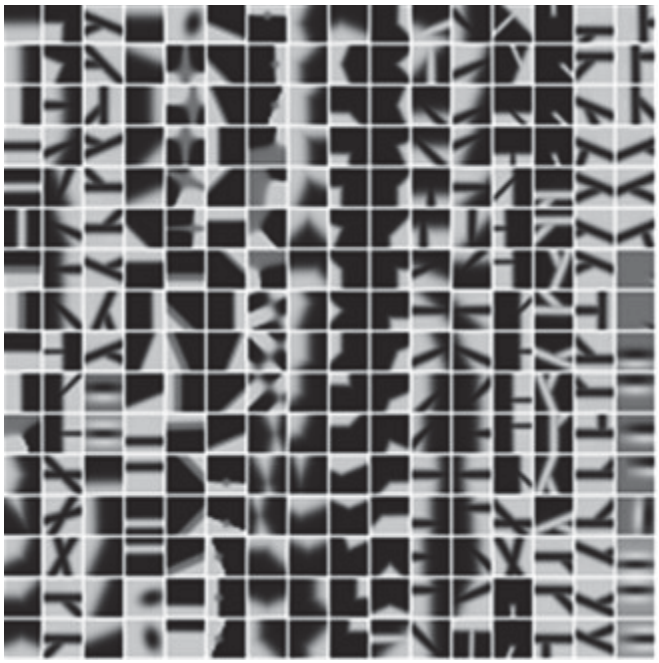
Appendix 5 Illustration of SymPaD pruned with size R = 256 according to four datasets



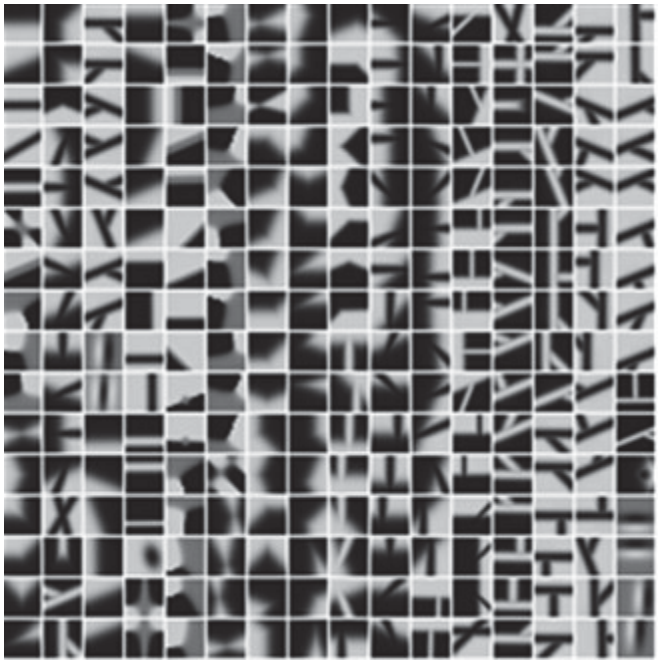
Dictionary pruned with size R = 256 according to COIL-100.



Dictionary pruned with size R = 256 according to Caltech-101

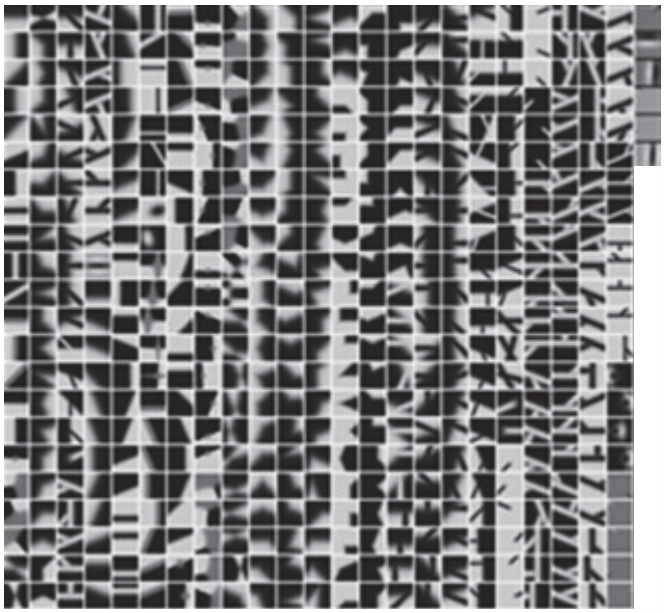


Dictionary pruned with size R = 256 according to ALOI-View.

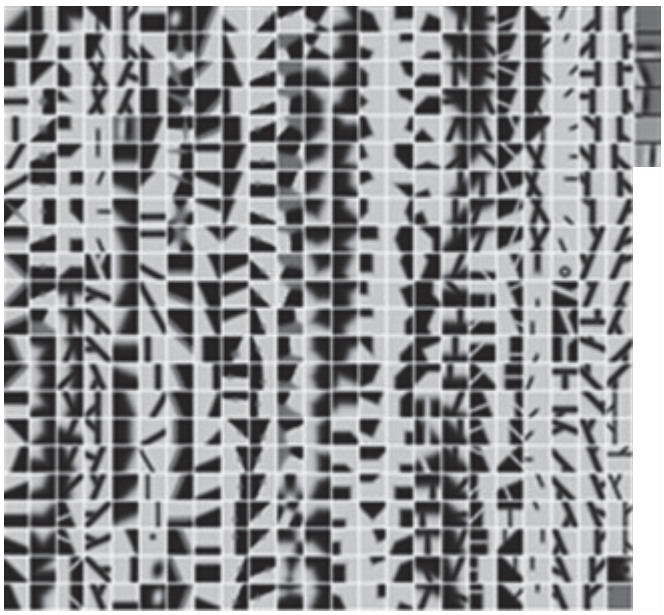


Dictionary pruned with size R = 256 according to ZuBuD

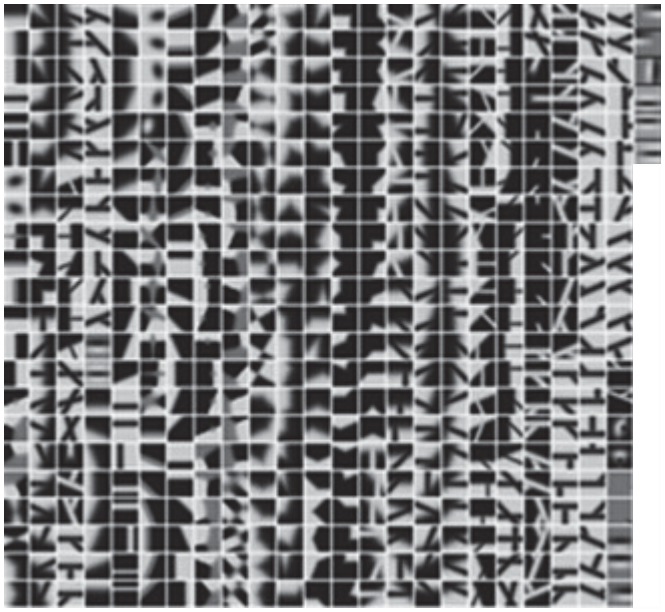
Appendix 6 Illustration of SymPaD pruned with size R = 512 according to four datasets



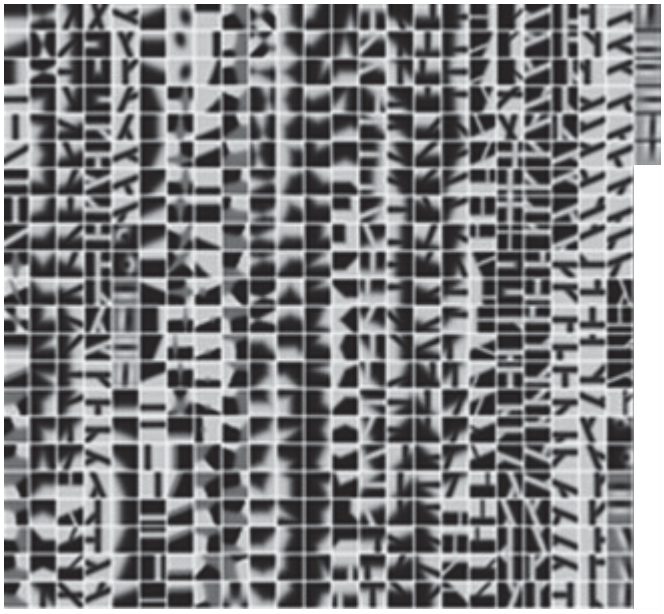
Dictionary pruned with size R = 512 according to COIL-100.



Dictionary pruned with size R = 512 according to Caltech-101

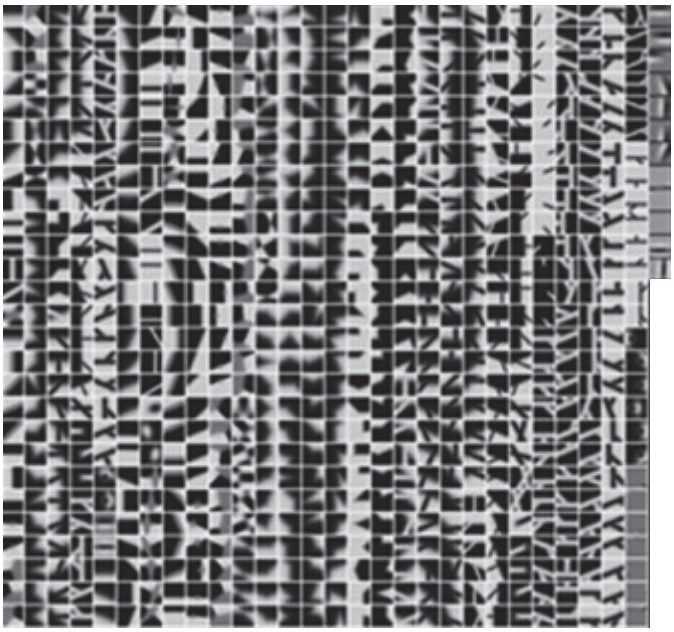


Dictionary pruned with size R = 512 according to ALOI-View.

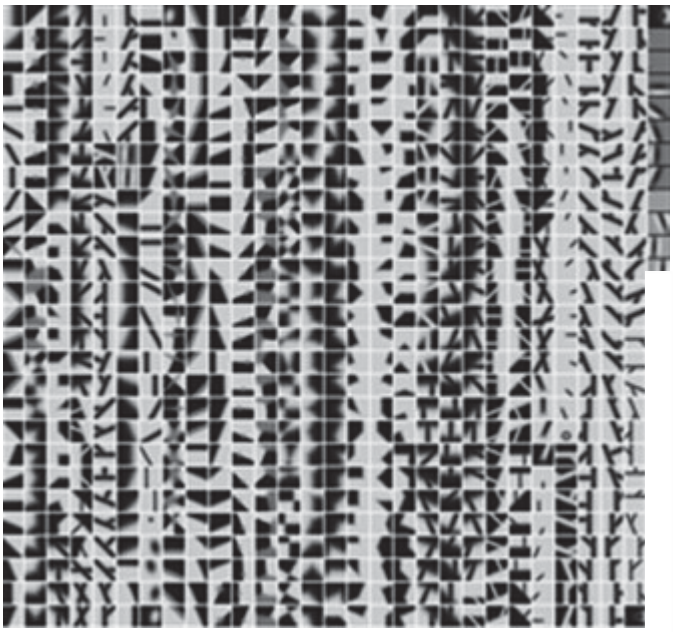


Dictionary pruned with size R = 512 according to ZuBuD

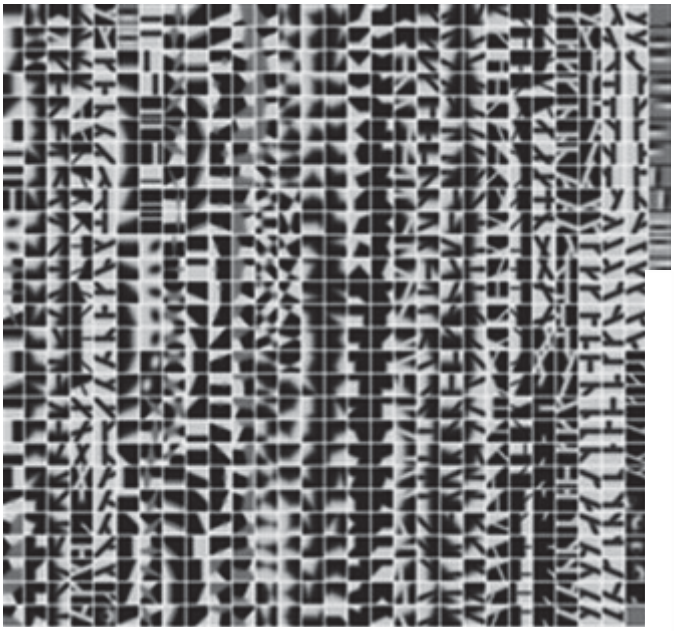
Appendix 7 Illustration of SymPaD pruned with size R = 768 according to four datasets



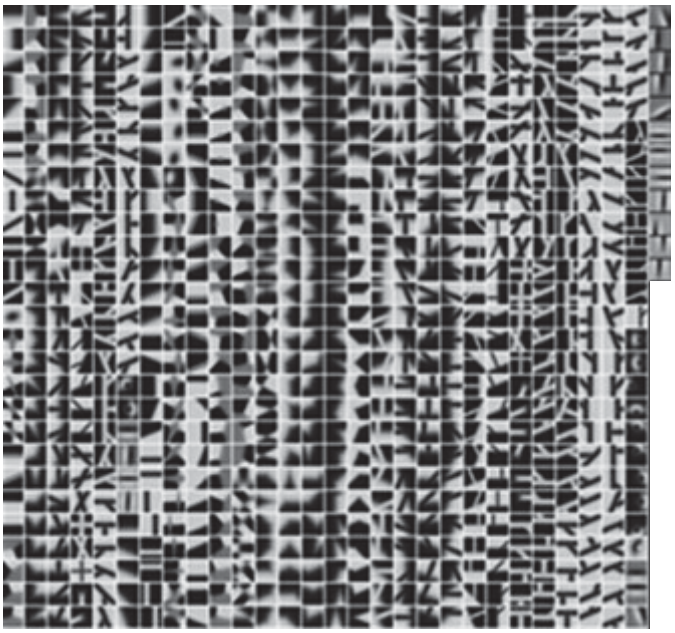
Dictionary pruned with size R = 768 according to COIL-100.



Dictionary pruned with size R = 768 according to Caltech-101

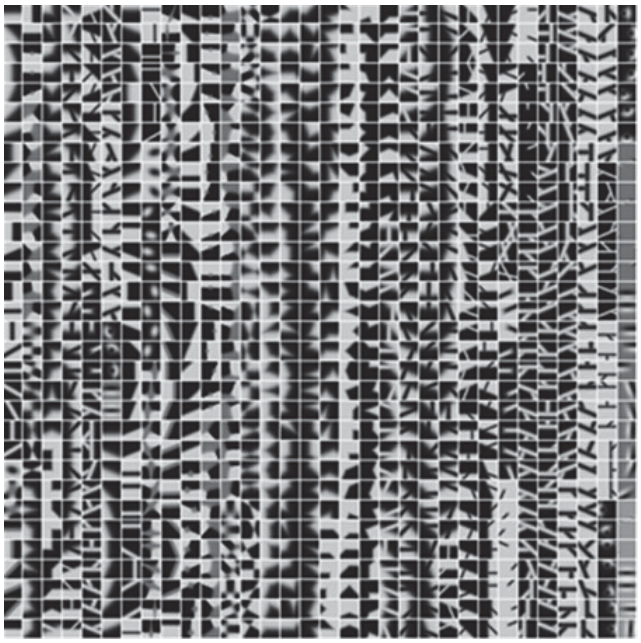


Dictionary pruned with size R = 768 according to ALOI-View.

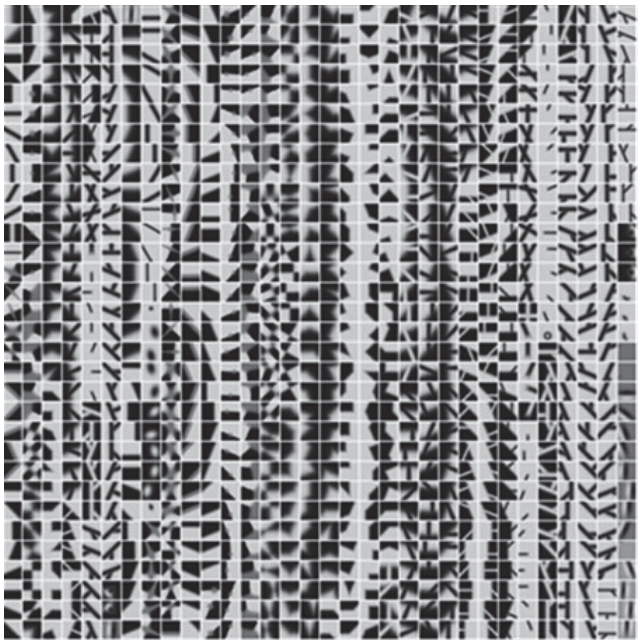


Dictionary pruned with size R = 768 according to ZuBuD

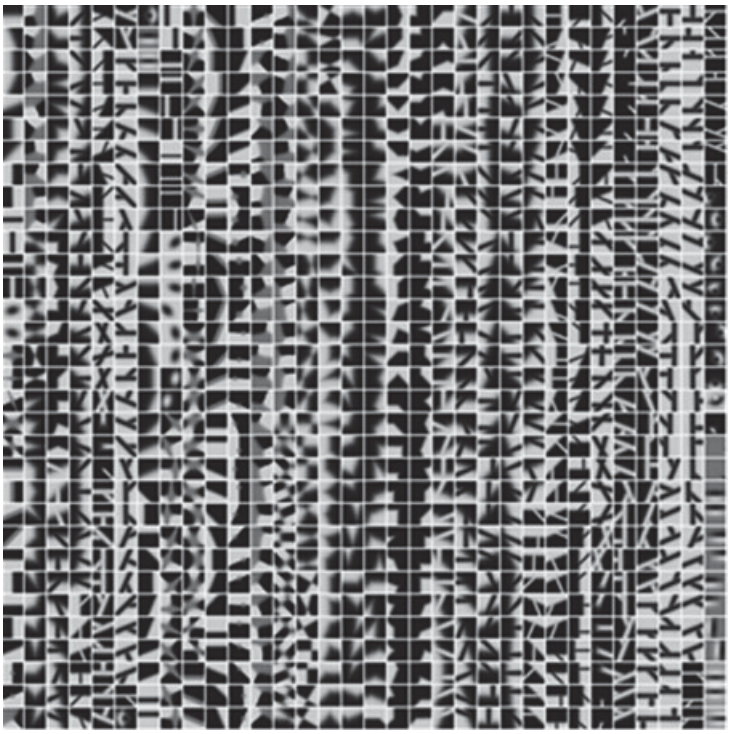
Appendix 8 Illustration of SymPaD pruned with size R = 1024 according to four datasets



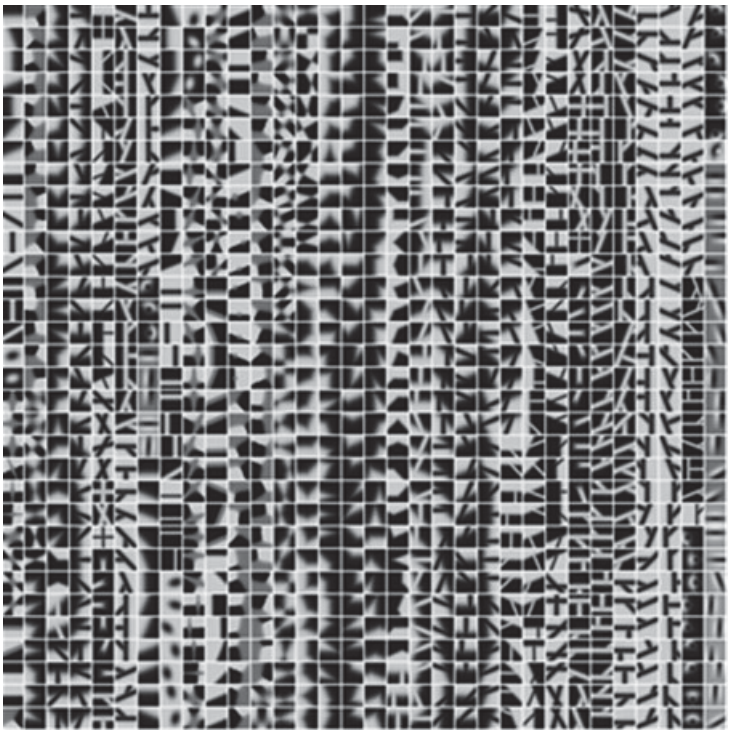
Dictionary pruned with size R = 1024 according to COIL-100.



Dictionary pruned with size R = 1024 according to Caltech-101

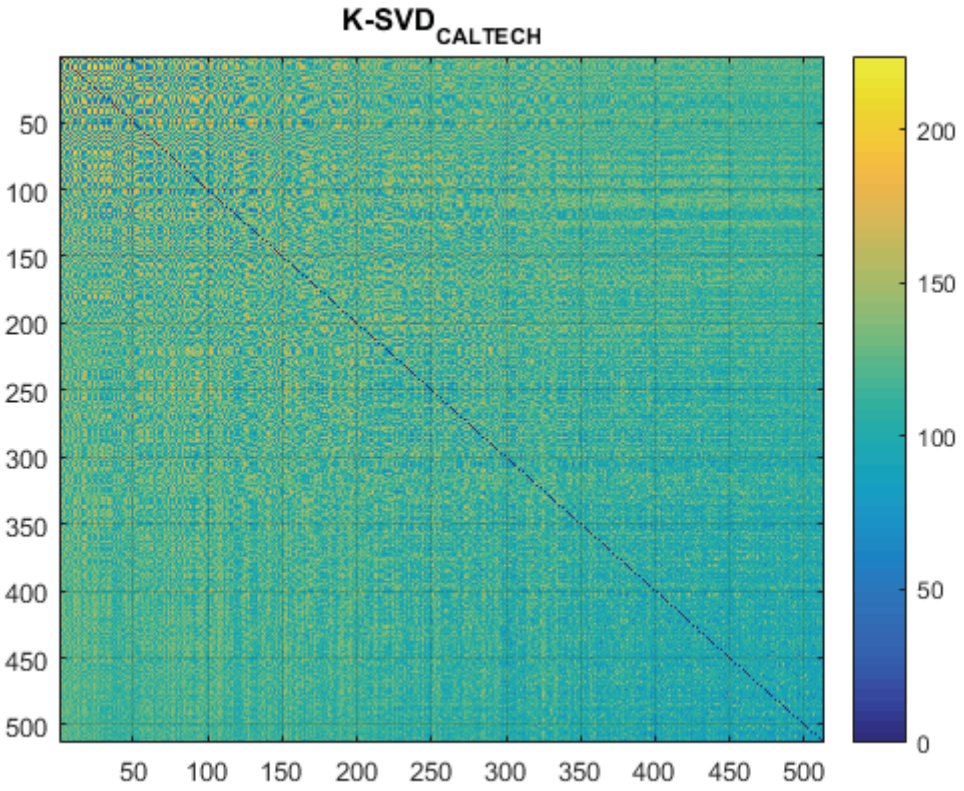


Dictionary pruned with size R = 1024 according to ALOI-View.



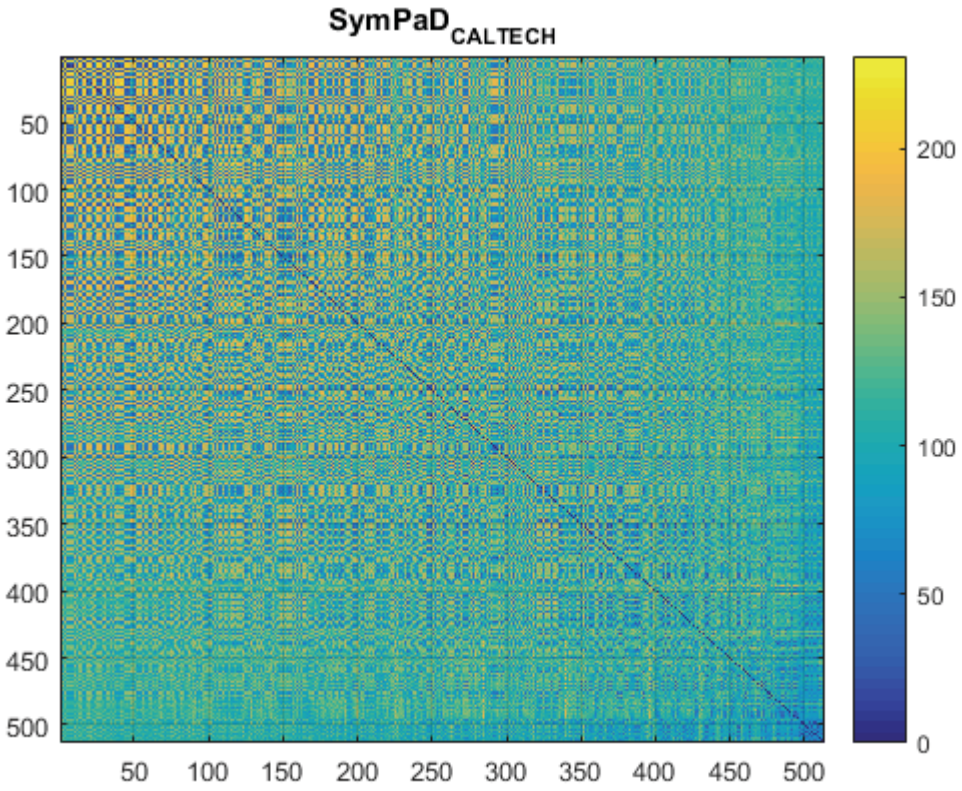
Dictionary pruned with size R = 1024 according to ZuBuD

Appendix 9 Redundancy in shape dictionary learned from Caltech-101 dataset by K-SVD



Hamming distance between BRIEF features of each shape atom to all other atoms is presented. More yellow colors indicates less redundancy.

Appendix 10 Redundancy in the SymPaD dictionary pruned according to Caltech-101 dataset



Hamming distance between BRIEF features of each shape atom to all other atoms is presented. More yellow colors indicates less redundancy.