SPARSE SPATIAL CODING:

A NOVEL APPROACH FOR EFFICIENT AND ACCURATE OBJECT RECOGNITION

GABRIEL LEIVAS OLIVEIRA

SPARSE SPATIAL CODING:

A NOVEL APPROACH FOR EFFICIENT AND ACCURATE OBJECT RECOGNITION

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação.

ORIENTADOR: MARIO CAMPOS

Belo Horizonte

Março de 2012

GABRIEL LEIVAS OLIVEIRA

SPARSE SPATIAL CODING:

A NOVEL APPROACH FOR EFFICIENT AND ACCURATE OBJECT RECOGNITION

Dissertation presented to the Graduate Program in Ciência da Computação of the Universidade Federal de Minas Gerais in partial fulfillment of the requirements for the degree of Master in Ciência da Computação.

ADVISOR: MARIO CAMPOS

Belo Horizonte

March 2012

Oliveira, Gabriel Leivas
D1234p Sparse Spatial Coding: A Novel Approach for Efficient and Accurate Object Recognition / Gabriel Leivas Oliveira. — Belo Horizonte, 2012
xv, 63 f. : il. ; 29cm

Dissertação (mestrado) — Universidade Federal de Minas Gerais
Orientador: Mario Campos

1. Sparse coding. 2. Object recognition. I. Título.

CDU 519.6*82.10

[Folha de Aprovação]

Quando a secretaria do Curso fornecer esta folha,
ela deve ser digitalizada e armazenada no disco em formato gráfico.

Se você estiver usando o `pdflatex`,
armazene o arquivo preferencialmente em formato PNG
(o formato JPEG é pior neste caso).

Se você estiver usando o `latex` (não o `pdflatex`),
terá que converter o arquivo gráfico para o formato EPS.

Em seguida, acrescente a opção `approval={`*nome do arquivo*`}`
ao comando `\ppgccufmg`.

We are what we repeatedly do. Excellence then, is not an act, but a habit.
Aristotle

Acknowledgments

First and foremost I would like to thank my advisor. Prof. Mario Campos is a great advisor, who was always supportive in my research endeavors and has taught me a lot about robotics, computer vision, and research in general.

I also need to thank all the people at VeRLab, Elizabeth, Samuel, Yuri, Armando, Douglas and Wolmar for the great interaction during this journey. Specially, I need to thank Erickson to introduce me to compressive sensing and for help me during this journey and Antônio Wilson for the partnership in several projects and for the close help with my dissertation.

The last, but not the least persons that I owe to thanks are my parents, my brother and my girlfriend for the unconditional support and to always give me strength to hunt my goals.

Thank all that make my masters at UFMG so enriching and unique experience!

Resumo

Até recentemente o reconhecimento de objetos, um problema clássico da Visão Computacional, vinha sendo abordado por técnicas baseadas em quantização vetorial. Entretanto, atualmente, abordagens que utilizam representação esparsa tem apresentado resultados significativamente superiores às técnicas usuais. Entretanto, uma desvantagem de métodos baseados em representação esparsa é o fato de características similares poderem ser quantizadas por conjuntos diferentes de palavras visuais.

Esta dissertação apresenta um novo método de reconhecimento de objetos denominado SSC – Sparse Spatial Coding – o qual é caracterizado pelo aprendizado do dicionário utilizando representação esparsa e codificação baseada em restrição espacial. Dessa forma, minimiza-se significativamente o problema típico encontrado em representações estritamente esparsas.

A avaliação do SSC foi realizada por meio de experimentos aplicando-o às bases Caltech 101, Caltech 256, Corel 5000 e Corel 10000, criadas especificamente para avaliação de técnicas de reconhecimento de objetos. Os resultados obtidos demonstram desempenho superior aos reportados na literatura até o momento para os métodos que utilizam um único descritor. O método também superou, para as mesmas bases, vários outros métodos que utilizam múltiplas características, e apresentou desempenho equivalente ou apenas ligeiramente inferior a outras técnicas. Finalmente, para verificarmos a generalização, o SSC foi utilizado para o reconhecimento de cenas nas bases Indoor 67, VPC e COLD tendo apresentado desempenho comparável ao de abordagens do estado da arte para as duas primeiras bases e superior na base COLD.

Palavras-chave: Visão computacional, Reconhecimento de objetos, Representação esparsa.

Abstract

Successful state-of-the-art object recognition techniques from images have been based on powerful techniques, such as sparse representation, in order to replace the also popular vector quantization approach. Recently, sparse coding, which is characterized by representing a signal in a sparse space, has raised the bar on several object recognition benchmarks. However, one serious drawback of sparse space based methods is that similar local features can be quantized into different visual words.

We present in this thesis a new object recognition approach, called Sparse Spatial Coding (SSC), which combines a sparse coding dictionary learning and a spatial constraint coding stage. Thus, we minimize the problems of pure sparse representations. Experimental evaluation was done at Caltech 101, Caltech 256, Corel 5000 and Corel 10000, that are datasets specifically designed to object recognition evaluation. The obtained results show that, to the best of our knowledge, our approach achieves accuracy beyond the best single feature method previously published on the databases. The method also outperformed, for the same bases, several methods that use multiple feature, and provide equivalent to or slightly lower results than other techniques. Finally, we verify our method generalization, applying the SSC to recognize scene in the Indoor 67 scene dataset, VPC and COLD, displaying performance comparable to state-of-the-art approaches in the first two bases and superior in COLD dataset.

Keywords: Computer Vision, Object recognition, Sparse coding.

List of Figures

List of Tables

List of Acronyms

SC	Sparse Coding
VQ	Vector Quantization
SPM	Spatial Pyramid Matching
BoF	Bag-of-Features
SPAMS	Sparse Modeling Library
SSC	Sparse Spatial Coding
PCA	Principal Component Analysis
SVM	Support Vector Machine
OMCLP	Online Multi-class LPBoost
SVD	Singular Value Decomposition
CBIR	Content-Based Image Retrieval
OCL	Orthogonal Class Learning
CRBM	Convolutional Restricted Boltzmann Machine

Contents

Chapter 1

Introduction

Recognizing objects in images has been a challenging task, and for a good number of years it has attracted the attention of a large number of researchers from several research communities such as robotics, computer vision and machine learning. Although almost all proposed techniques are based on good data representation, an inadequate representation can greatly influence the accuracy of those methods. Generally, these feature representations are designed manually or need significant prior knowledge. Therefore, to overcome this issue we present a novel coding process that automatically learns a representation from unlabeled data. Additionally, we explain how to build a sparse representation of an image, which represents an input signal, in our case data extracted from image patches, as a small combination of basis vectors, used to learn low level representations from unlabeled data.

Sparse Coding (SC) techniques are characterized by a class of algorithms that learn basis functions from unlabeled input data in order to capture high-level features. These high level features are signatures that encode an input signal as a combination of a small number of elementary signals. Frequently, those signals are selected from a dictionary. SC has been successfully used for image denoising [Elad and Aharon, 2006] and image restoration [Mairal et al., 2008b,a]. However, only recently SC has been effectively applied to replace Vector Quantization (VQ) techniques in object recognition tasks, and it is now considered the state-of-the-art with the best results for several datasets [Yang et al., 2009b; Jiang et al., 2011].

Before we fully state our problem, we will first introduce some key definitions used throughout this text, and more specifically in our methodology. These terms are: sparse and non-sparse representations, dictionary learning and pooling process.

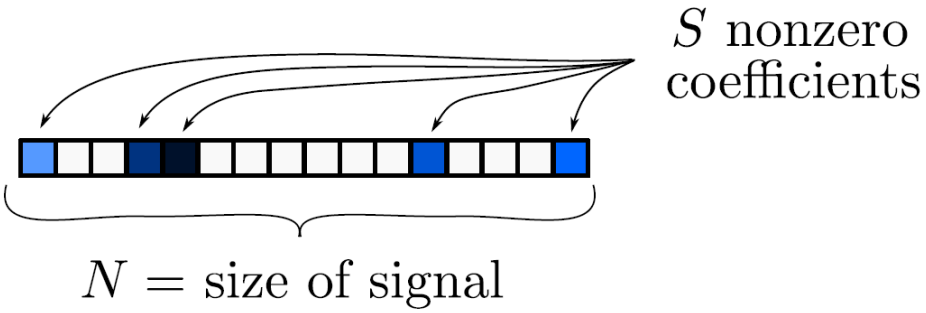


Figure 1.1: Graphical representation of a sparse vector. S represent the active coefficients for an input signal X. To be considered a sparse representation the number of activations must be a fraction of the total number of elements that could express such signal ($S \ll N$).

1.1 Sparse representations

Let $X \epsilon R^n$ be a discrete signal. X is S-sparse if it is a linear combination of S basis vectors, where $S \ll N$. Figure 1.1 exemplifies a sparse vector.

Furthermore, it is worth to explain that this kind of representation assumes that the input signal is also sparse. Therefore, similarly to Yang et al. [2008], we employ image patches as our sparse input signal to perform object recognition. Our motivation to use sparse representations was based mainly on the following observations:

- Sparse representation methods show robustness to signal recovery from noisy data;
- Sparsity has also been regarded as likely to be separable in high-dimensional sparse spaces [Ranzato et al., 2006] and therefore suitable for classification.

1.2 Non-sparse representations

Non-sparse representations can be seen as a signal composed by a set of values, where most of them are non-zero. For example, SIFT [Lowe, 2004] descriptors are composed by 128 float numbers, where the majority of them are not zero. Approaches to recognition generally concatenate SIFT descriptors to obtain an image signature that can be considered non-sparse or dense.

- dense ($l2$ normalization), generally a large number of coefficients.

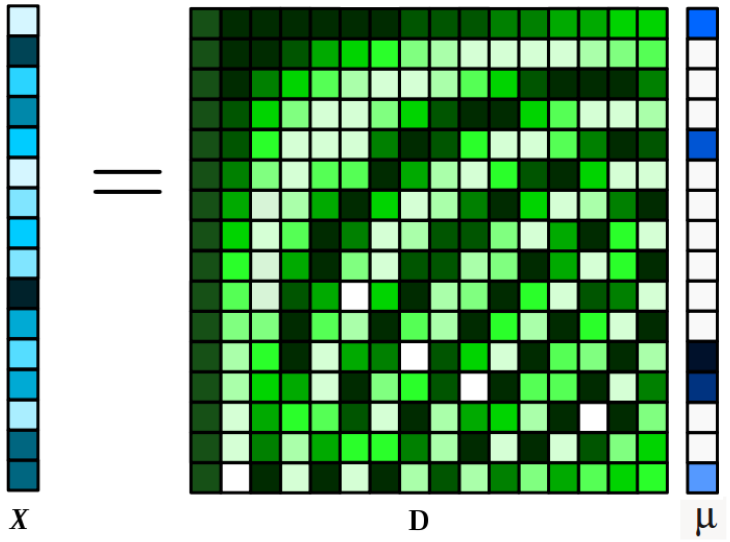


Figure 1.2: This figure shows an input signal x that is a linear combination of the dictionary D and it activation vector μ. Cells filled with blue color represent the active dictionary elements of x.

$x_2 = \text{argmin} \|x\|_2$ s.t. $Ax = y$ that represent a sample y as a linear combination of training samples A.

- sparse ($l1$ normalization), produce a sparse number of active coefficients.

  $x_1 = \text{argmin} \|x\|_2$ s.t. an approximation of $l0$ normalization.

1.2.1 Dictionary Learning

Dictionary learning algorithms receive as input tokens that, in our case, are random image patches, and learn P, which in this work will be generally considered to be $P = 1024$, basis functions.

For a set of input signals $x^{(1)}, x^{(2)}, ..., x^{(m)}$ in $R^{m \times n}$, we learn a dictionary that is a collection of bases $D_1, D_2, ..., D_k$ in $R^{m \times p}$, so that each input x can be decomposed as:

$$x = \sum_{j=1}^{m} D_j \mu_j, \tag{1.1}$$

$$s.t. \mu_{j's} \text{ are mostly zero,}$$

where μ_j is the set of basis weights for each input signal. Figure 1.2 depicts a dictionary and how it is used to represent a signal.

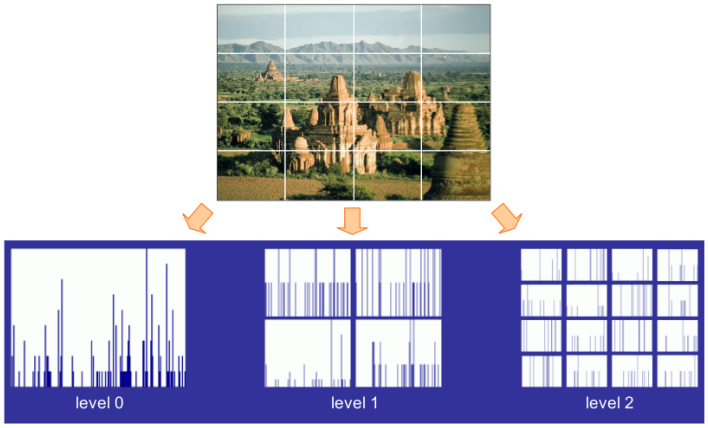


Figure 1.3: Example of multi scale pooling, called spatial pyramid matching, by [Lazebnik et al., 2006].

1.2.2 Pooling

Pooling consists of summarizing the coded features across an image to form a global image representation. The objective of such method is to achieve invariance to image transformation and robustness to noise and clutter, removing spurious data while preserving relevant information.

Several pooling functions were proposed to build image signatures, and those that attained higher success were average and max pooling. Max pooling extracts the largest response in the collection of descriptors with respect to each dictionary element. We have chosen this function, in lieu of average pooling, since the works of Liu et al. [2011]; Boureau et al. [2010b] prove that max pooling attains state-of-the-art results. In addition, we perform max pooling in a spatial pyramid image representation, Figure 1.3. This is preferable, since max pooling under different locations and spatial scales provides more robustness to local transformations.

1.3 Problem definition

The problem dealt with in this thesis is on the uncovering of the semantic category of an image. Much of the work for whole image categorization has already been successfully accomplished using Bag-of-Features (BoF) approaches. However, BoF methods represent an image as an orderless collection of local features, which obviously does not capture global features, such as shape, to distinguish objects. In order to overcome this loss of spatial information, an extension to BoF called Spatial Pyramid Matching (SPM), was proposed by Lazebnik et al. [2006]. Nowadays, SPM is

an important component of state-of-the-art object recognition techniques [Boureau et al., 2010a; Gao et al., 2010; Wang et al., 2010; Yang et al., 2009b; Coates et al., 2011].

Indeed, using SPM is preferable to improve visual object recognition, since it creates geometrical relationships between features, which combined to SC, leads to high accuracy results.

This work presents a new approach, called Sparse Spatial Coding (SSC), for object recognition which takes advantage of SPM and overcomes SC drawbacks by implementing a spatial Euclidean coding representation.

Our method is composed of three main steps:

- Training phase;
- Coding phase;
- The use of an learning approach, that could be an off-line classifier, called Orthogonal Class Learning (OCL), or an online method.

In the Training Phase the dictionary is built. Image patches are randomly extracted from the set of training images, they are normalized and then are passed on to the learning process that builds the dictionary.

The Coding Phase can be divided into two steps: i) the extraction of local descriptors, which may use descriptors like SIFT [Lowe, 2004] or SURF [Bay et al., 2006], and ii) code generation, based on the dictionary and on the quantization of each descriptor, using a spatial constraint, instead of just sparsity. Next, the codes associated with each region are pooled together to form a global image signature.

The final stage of our method sends the global features to one of the two classification methods. The first method is an off-line methodology called OCL that takes advantage of the high dimensionality of feature vectors when compared with the number of feature examples. The second possible classifier is an online classification method. We chose to use online learning motivated by requirements of tasks that need to be executed by mobile robots : i) small memory availability; ii) large amount of data, and iii) suitability for data streaming.

Online learning is well suited to several robotic tasks where, in general, the robot does not have access to the entire data domain. This is also very similar to decision making problems, where parts of the data are incrementally presented over time [Saffari et al., 2010]. This idea can be exemplified by a simple game quiz, with a student and a teacher execute n times the following steps:

1. An input sample is presented to the student.

2. The student responds to the input with a prediction.

3. The teacher reveals the true answer for the input.

4. If the prediction is correct, then the model is reinforced, if it is wrong, the student is penalized and his model is updated.

The goal of the student is to minimize the cumulative error over time by updating its internal model of the problem.

Experimental results presented later in the work show that, to the best of our knowledge, the results we obtained over several object recognition datasets, such as Caltech 101, Caltech 256, Corel 5000 and Corel 10000, showing accuracies beyond the best published results so far on the same databases. We also show that the proposed approach achieves state-of-the art performance on the COLD place recognition dataset.

In addition, high performance results were obtained on the MIT-67 indoor scene recognition dataset and VPC Visual Place Categorization dataset.

1.4 Publications

Results from the work developed in this thesis were accepted for publication in two major conferences in the field, and another one will be submitted to IROS 2012:

Conferences and Workshops

- Oliveira, G. L. ; Nascimento, E. ; Vieira, A. W. ; Campos, M. . Sparse Spatial Coding: A Novel Approach for Efficient and Accurate Object Recognition In: 2012 IEEE International Conference on Robotics and Automation , 2012, St. Paul - Minnesota - USA.

  Qualis A1

- Nascimento, E. ; Oliveira, G. L. ; Vieira, A. W. ; Campos, M. . Improving Object Detection and Recognition for Semantic Mapping with an Extended Intensity and Shape based Descriptor. In: IROS 2011 workshop - Active Semantic Perception and Object Search in the Real World (ASP-AVS-11), 2011, San Francisco. Proc. IROS Workshop ASP-AVS-11, 2011.

1.5 Contributions of the Thesis

The main contributions of this thesis are:

- A novel unsupervised feature learning method which uses SC for dictionary learning and a coding stage based on spatial constraint, called Sparse Spatial Coding (SSC);
- An object recognition technique based on an online classification method, which when combined with the previous steps, leads to state-of-the-art performance results on several benchmark datasets;
- A new off-line method called OCL, which takes advantage of the high dimensionality of features when compared to the number of feature examples;
- A deep parameter analysis of the most relevant settings, showing their effects on system accuracy and performance.

1.6 Thesis Outline

This thesis is structured as follows:

Chapter 2: We present and discuss related works on object recognition, focusing on sparse representation methods. Moreover, we give special attention to unsupervised feature learning methods that use sparse representation and additional constraints, like spatial similarity.

Chapter 3: Sparse Spatial Coding (SSC), which combines a sparse coding dictionary learning approach with a coding module which considers both sparsity and locality is carefully laid out in this chapter. We also present a novel off-line classification method, called Orthogonal Class Learning (OCL), that builds compact feature signatures to improve memory efficiency. In addition, we present an online learning algorithm that is a key part of our final object recognition approach.

Chapter 4: This chapter describes experimental results for a series of object recognition datasets namely, Caltech 101, Caltech 256, Corel 5000 and Corel 10000 and three scene/place recognition datasets, Indoor 67, VPC and COLD. Furthermore, an empirical analysis is performed on the main system parameters, showing the effects of their settings to system performance.

Chapter 5: Presents the conclusions we gathered from this work, the investigation underway and future research directions.

Chapter 2

Related Works

One of the most fundamental problems dealt with by the computer vision community is object recognition, which is concerned with identifying which type of object or set of objects are presented in an image. Solving this problem accurately, and if possible with low computational burden, directly impacts several research areas, like robotics perception and content-based image retrieval.

Seminal works that address object recognition are dated more than four decades ago [Agin, 1972; Binford, 1971]. Some limited scope applications have achieved significant success such as: handwriting digits, human face and road signs recognition tasks. In the 70's, as range sensors became popular, $3D$ data was readily available and used. In the 80's, $2D$ images were commonly used, however object data were obtained under controlled conditions, with uniform background and structured lighting to facilitate the segmentation step. The first approaches dealt with a single object class under several viewpoints, and only later multi-class methods appeared. Nonetheless, those techniques explored only a limited number of categories in controlled environments.

Object recognition methods can be divided into three main categories:

- Geometry based;
- Appearance based;
- Feature-points algorithms.

Many of the first object recognition techniques use geometrical representations based on edge contours extracted from object image. Those methods present some interesting features such as being almost unaffected by illumination changes and to variations in appearance due to different viewpoints.

Appearance based algorithms try to solve the object recognition problem by computing eigenvectors. While this kind of algorithms shows good results for object recognition tasks under significant viewpoint and illumination changes, they are affected by occlusion.

The last group of object recognition approaches are characterized by finding feature points, often present at intensity discontinuity on images. Although feature based algorithms present robustness to clutter scenes and partially occluded objects, they fail for textureless images and to small number of extracted keypoints.

2.1 Geometrical approaches

The first efforts to tackle the object recognition problem used data produced by range sensors [Agin, 1972; Binford, 1971; Bolles and Horaud, 1987; Ponce and Brady, 1987]. The main idea is that geometrical description of a $3D$ CAD object model allows the projected shape to be accurately predicted in a $2D$ image, thereby making the recognition process easier if edge or boundary information are used [Yang, 2011]. Geometrical techniques can be divided into two groups: i) alignment based approaches, which try to match an image between available models; ii) aims at to employing small image sets to compute a viewpoint, used as key for a hashing algorithm.

2.1.1 Alignment algorithms

Two stages compose the alignment based approaches. First, a correspondence step between a $3D$ model and an image, which employs lines and point sets to infer the transformation, is performed. Then, a second stage that uses edge information is executed to support the proposed location. Based on the unavailability of matches over all the available data due to exponential number of possibilities, alternative approaches, like interpretation trees [Grimson and Lozano-Prez, 1987], were explored to optimize the search process.

Lowe [1987] is one representative work of alignment techniques. First, it extracts lines from target images, then it clusters the information using co-linearity and parallelism. The unknown viewpoint is obtained from projections of groups of lines over the $3D$ model. Lowe also applies subsets of lines within the model, instead of in all domain, to achieve occlusion robustness.

Mundy and Heller [1990] propose an alignment object recognition method in which $3D$ CAD models were employed to find objects from aerial images. The pro-

cess clusters estimated poses from edge data.

Ullman and Basri [1991] address the problem of how to describe a $3D$ model as a combination of $2D$ representations, and matches are performed with this mixture model using lines and points.

2.1.2 Geometrical hashing methods

Lamdan et al. [1988]; Rigoutsos and Hummel [1995] describe hashing methods for recognition, which use text-based hashing as foundation and objects are modeled as a set of interest points from the edge. Those points are made invariant to affine transformations using three points from the set. In the learning step, all three point sets are used, and the remaining points for each set are stored in a hash table. Objects are recognized by extracting interest points from a set of images and using the results to index a hash table. This produces a number of answers for each object model. The class that is "closer" as far as similarity is concerned corresponds to the model that produces the strongest response to the input, in our case, an image. Redundant points also provide robustness to occlusion, but unfortunately at the cost of increased false positive rate in noise and/or clutter points. Rigoutsos and Hummel [1995] overcome this limitation with a probabilistic voting scheme.

The major strength of the aforementioned methods is the low computational requirements, because each object needs only to be searched in hash tables. Hence, lookup time is constant. Another positive aspect of all geometrical methods is their ability to recognize objects in an affine or projective invariant way. Similarly to alignment algorithms, geometric hashing methods can provide invariance to affine/projective transformations running at fast rates.

However, geometrical methods have as the main disadvantage to assume that contours will be reliably found, which is not true with images from real scenes, due to changes in lighting, clutter and occlusion. Finally, these methods accomplish the object recognition task in a controlled experimental setup, and do not perform well in real world situations.

2.2 Appearance based methods

Appearance based recognition methods are the first methods to example based recognition under ideal conditions, *e.g.* no occlusion and controlled light conditions.

The *eigenfaces* work of [Pentlan, 1986] uses Principal Component Analysis (PCA) in the pixel level, to recognize faces. Another work which uses PCA for

object recognition tasks is [Murase and Nayar, 1995]. In spite of the fact that previous object recognition works rely on shape, the aforementioned works use appearance based features. PCA gives a compact object representation by using as parameters pose and lightning. In order to build a final representation of an object, a vast quantity of images under different poses and illuminations need to be acquired. These images are compressed and form a low dimensional space called eigenspace, in which an object is represented as a manifold [Murase and Nayar, 1995]. Recognizing objects in this approach is accomplished by checking if a given object, transformed to the eigenspace, lies in one of the manifolds.

Zhou and Chellappa [2003] also exploit eigenspaces to compressed the training data and use particle filters with inter-frame appearance based modeling to track and to recognize objects from diverse poses and illumination conditions.

Bischof and Leonardis [2000] employ the Random Sample Consensus (RANSAC) technique to provide robustness to occlusion. The method randomly selects a subset of target pixels and finds the best eigenvector coefficients that fit those pixels. Each interaction discards the worst fit pixels, which are probably noise, and continue to iterate until a robust measurement of the eigenvector, the one that best fits the image, is found. Those coefficients are then used in the recognition step.

On one hand, the key advantage of all appearance algorithms are their simplicity, the fact that they do not require prior knowledge of the object's shape and reflectance properties, and their efficiency, since recognition can be handled in real time, and those methods exhibit robustness to image noise and quantization. On the other hand, acquiring training data is an arduous task, since it is necessary to perform scene segmentation prior to starting object training, and no occlusion is allowed. Another disadvantage is related to objects with high dimensionality eigenvectors, which require non-linear optimization methods, known to be computational costly.

2.3 Feature Points Object Recognition

Feature points methods gained popularity in the late 90's, mainly due to their robustness to clutter and partial occlusion [Lowe, 1999; Rothganger et al., 2005; Belongie et al., 2002; Boiman, 2008; Lazebnik et al., 2006; Saffari et al., 2010]. Inspired by the machine learning literature and the arrival of new classifiers such as Linear SVM, Online Random Forests (ORF) and Online LPBoost, that could run within acceptable frame rate on standard computers and deal with large scale datasets, com-

puter vision scientists started to research ways to extract features from images and apply machine learning techniques to identify objects from this set of keypoints.

Since our work focuses on sparse representation for object recognition, local feature works will be broken into non-sparse and sparse representation methods; the latter includes our method.

2.3.1 Non-sparse methods

We consider as non-sparse all those approaches without sparse representation modules, such as sparse dictionary learning or sparse coding process. For example, SIFT [Lowe, 2004] descriptors are composed by 128 float numbers, where the majority is not zero. Non-sparse methods generally concatenate descriptors, for instance SIFT or SURF, to obtain an image signature, that can be considered non-sparse or dense.

Lowe [1999] proposes an algorithm to extract keypoints using difference-of-gaussian operators. For each point, a feature vector is extracted. Local orientation is estimated through a number of scales and over a neighborhood around each point and the angle is expressed based on the dominant local orientation, providing rotational invariance. An object is recognized if a new image presents the same number of features of the object template and at similar locations.

Grauman and Darell [2006] use a bag of features (BoF) algorithm for recognition. The process consists of extracting SIFT features and concatenating them using a multi-scale pyramid pooling method. A training set is compared with a test set to measure the similarity between the two sets of features.

As far as we know, Lazebnik et al. [2006] presents the first work on SPM, once BoF, which was previously applied to same problems, presents a severe weakness, of discarding spatial order of local descriptors, harshly limiting the discriminative power of representations. Lazebnik's method extracts SIFT features from an image and repeatedly subdivides it and computes the histograms of local features at increasingly fine resolutions [Lazebnik et al., 2006]. Histograms are pooled across different locations and spatial scales to provide robustness to local transformations. These pooled features are concatenated to form a spatial pyramid representation of the image. The authors tested their representation, considered as a global feature, on a scene and object recognition task. They showed that global representation can be effective, not only to identify scenes, but also to classify scenes based on objects.

Boiman [2008] addresses the nearest neighbor (NN) classifiers problem of accuracy when compared with Support Vector Machine (SVM) techniques. Boiman points out two practices that looses performance in these methods: (i) quantization

of local feature descriptors, and (ii) the use of "image to image" distance, instead of "image to class" distance. A Naive-Bayes Nearest-Neighbor (NBNN) algorithm that only uses NN distance to local feature descriptors, specially "image to class" distance with no quantization is proposed. Boiman [2008] carries out experiments with a single descriptor, in this case SIFT, and with a combination of five types of descriptors: (1) SIFT, (2) luminance descriptor [Boiman, 2008], (3) color descriptors [Boiman, 2008], (4) Shape-context descriptor [Mori et al., 2005] and (5) Self-Similarity descriptor [Shechtman and Irani, 2007a]. Therefore, beyond the simplicity and efficiency to compute, this method also presents top results on Caltech 101[1] and Caltech 256[2] datasets.

Saffari et al. [2010] proposes a new online boosting algorithm to multi-class problems, called Online Multi-class LPBoost (OMCLP). Online learning is an essential tool for learning from dynamic environments, from large scale datasets and from streaming data sources, which is a desirable capability to perform robotics tasks. The author evaluates the method on the Caltech 101 dataset and uses as features a Level2-PHOG descriptor from [Gehler and Nowozin, 2009a].

2.3.2 Sparse representation methods

An extensive body of literature exists on non-sparse object recognition. However, we now focus in methods which generate global sparse representations for images in order to recognize different categories of objects. More specifically, we will investigate a recently proposed theory called Sparse Coding (SC), which refers to a general class of techniques that automatically select a sparse set of vectors from a large pool of possible bases to encode an input signal [Yu et al., 2011]. Based on the robustness of sparse representations to noisy data and on the suitability of sparse signatures to be separable in high-dimensional sparse spaces, we choose to use sparse representation to our work.

Several approaches using SC with dictionary learning for image classification have been proposed in recent years. These approaches can be divided into two main categories:

- Supervised feature Learning;
- Unsupervised feature learning.

[1] *www.vision.caltech.edu/Image_Datasets/Caltech*101/
[2] *www.vision.caltech.edu/Image_Datasets/Caltech*256/

Supervised feature learning can be defined as feature learning techniques which use supervised dictionary learning [Boureau et al., 2010a; Jiang et al., 2011; Zhang and Li, 2010; Aharon et al., 2006; Zhang et al., 2006]. The second class of sparse representation methods, called unsupervised feature learning, rely on unsupervised dictionary learning to learn representations from low level descriptors, such as SIFT [Lowe, 2004] or SURF [Bay et al., 2006], and provide discriminative features for visual recognition [Gao et al., 2010; Wang et al., 2010; Yang et al., 2009b; Yu et al., 2011; Sohn et al., 2011], in which the present work is part of.

Three recent works which deal with SC and supervised dictionary learning are Jiang et al. [2011], Zhang and Li [2010], and Boureau et al. [2010a]. Jiang et al. [2011] propose a supervised dictionary learning technique called Label consistent KSVD (LC-KSVD). This technique associates a label (a column of the dictionary matrix) to increase the discrimination power in sparse coding during the learning process of a dictionary. This method combines dictionary learning and a single predictive linear classifier into the objective learning function.

Zhang and Li [2010] also propose an extension for the K-SVD method [Aharon et al., 2006], called discriminative K-SVD (D-KSVD). However, the method incorporates in the dictionary learning phase, a policy of building a dictionary with not only a good representation (which means a dictionary for image reconstruction), but with a high discriminative power (for recognition tasks). The proposed method incorporates categorization error into the objective function.

In Boureau et al. [2010a], the authors proposed a method for supervised dictionary learning with a deep analysis of coding and spatial pooling modules. This evaluation ushered in two discoveries: First, that sparse coding improves soft quantization, and second, that max pooling, almost in all cases, is superior to average pooling, which is unequivocally perceived when using a linear SVM.

Another research stream is related to unsupervised dictionary learning for object recognition. Some approaches, like Yang et al. [2009b]; Sohn et al. [2011], use SC alone. Recent works have also proposed additional regularization and/or constraints, such as spacial properties, like Yu et al. [2011]; Gao et al. [2010]; Wang et al. [2010] and Kavukcuoglu et al. [2009].

Yu et al. [2011] proposes an unsupervised feature learning algorithm using a two-layer SC scheme at the pixel level. The first layer encodes individual patches, followed by a second layer that is responsible to join sets of patches belonging to similar groups. Two dictionaries must be learned together, where each code in the second dictionary level represents patterns among the first dictionary layer, to produce representations that are more invariant than single layer approaches, like [Yang

et al., 2009b]. Moreover, multi-level dictionaries, whose codes model dependency patterns of patch layer, allows the encoding of more complex visual templates. Yu et al. [2011] performs tests on digit and object recognition tasks, showing superior results when compared with single-layer sparse coding.

Sohn et al. [2011] address the challenges of training Restricted Boltman Machines (RBM), providing an efficient sparse RBM approach, with almost no hyper-parameter tuning requirement. As a primary goal, the authors examine theoretical links among unsupervised learning algorithms and take advantage of these models to train more complicated methods [Sohn et al., 2011]. The methodology consists of learning a signature based on SIFT and RBM, producing state-of-the-art results.

Yang et al. [2009b] propose an extension to the SPM method of Lazebnik et al. [2006] by replacing vector quantization for a sparse coding approach. After running SPM, a max pooling technique is applied to summarize all image local features. By incorporating locality, Wang et al. [2010] aims at decreasing the reconstruction error of sparse coding algorithms based on the idea that similar patches will have similar codes given the locality.

Our approach may be classified as an unsupervised dictionary learning technique, and more specifically, it resembles the work of Wang et al. [2010] and Gao et al. [2010]. However, instead of using locality for dictionary learning and coding, our method uses sparse representation for the dictionary, given that our data for training is limited. As Coates and Andrew [2011] conclude, sparse coding achieves consistent results when a small number of examples are available. Rigamonti et al. [2011] presents an analysis of the relevance of sparse representation for image classification, also pointing out the importance of sparsity for learning feature dictionaries.

2.4 Considerations

The aforementioned studies on the geometry and appearance based object recognition are already well established categories in the literature. However, expanding feature based algorithms in sparse and non-sparse representation for object recognition is a novel taxonomy.

The first works covering object recognition have been proposed around forty years ago and use $3D$ data of objects to recognize images (2D representation), included in geometrical object recognition approaches. Followed by appearance-based methods, and recently, in late $90's$, by techniques based on feature descriptors,

propelled by developments of powerful machine learning techniques. Over the last few years, great advances in object recognition have been attained by methods employing sparse representation. Sparse representation is a widely used theoretical subject in signal processing. It became the central module of several state-of-the-art object recognition approaches. For instance, 16 papers were published in CVPR 2011 and 13 in ICCV 2011, which deal with sparse representation for object recognition.

Our object recognition approach can be classified into feature based sparse representation methods, more specifically as unsupervised feature learning. The thesis contributions are mainly i) an object recognition module and ii) a classification method based on SVD, called OCL.

Chapter 3

Methodology

Object recognition has proven to be an important tool for robotics perception. Nevertheless, almost all proposed techniques rely on having a good representation of data, since an inadequate representation can greatly influence the accuracy of those methods. Generally, these feature representations are hand-designed or require significant prior knowledge. To address this issue, we will present a novel coding process that automatically learns a good feature representation from unlabeled data. Specially, we will present a method for object recognition based on unsupervised feature learning. We also describe how to build a sparse representation of an image, which represents each input example as a small combination of basis vectors, used to learn low level representations from unlabeled data.

Figure 3.1 presents the object recognition system proposed in this thesis. First, features are extracted and descriptors are obtained. We use SIFT to extract features. Then, a second phase responsible to learn a sparse dictionary, is carried in an unsupervised way. After building a dictionary, we perform the coding process. In our case the coding process considers not only sparsity, but also spatial similarity, defined here as Sparse Spatial Coding (SSC). This codes are pooled using a max pooling method, forming a global feature. Finally, this image signature is presented to a learning method that could be our off-line OCL or the online LaRank.

3.1 Feature Extraction

Choosing the appropriate feature is a critical step in object recognition methodologies. In this work, we will follow an approach that is similar to Fei-Fei and Perona [2005], which models the extraction phase by a collection of local patches. Each patch will be used to construct a signature defining a word to build our dictionary.

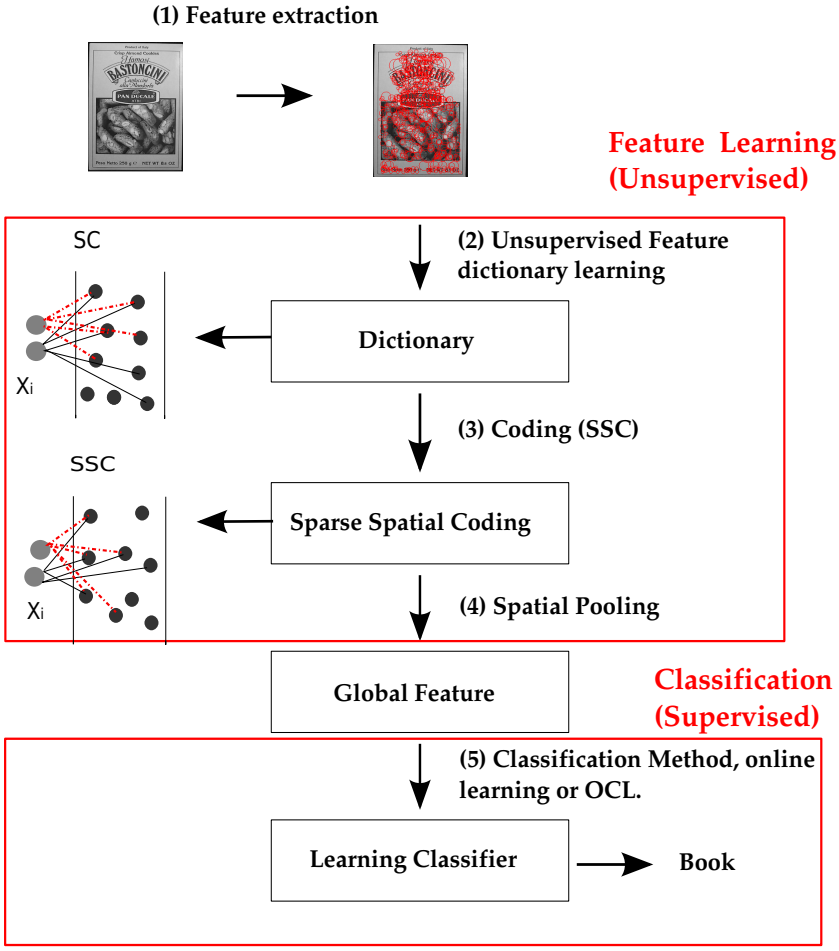


Figure 3.1: Object recognition system overview. First image descriptors are obtained, followed by the dictionary learning module, then the SSC coding process is performed, encapsulated by the feature learning module. Finally, these codes are pooled and send to a classifier, an off-line OCL, proposed in this work, or an online learning approach.

Rather than using feature detectors for recognition, we apply a dense feature extraction procedure that presents robustness to occlusion and spatial shift. This sampled grid approach consists of an equal grid space partitioning to obtain a patch of $n \times n$ size. We describe each patch by a SIFT descriptor, once several works [Lazebnik et al., 2006; Fei-Fei and Perona, 2005; Yang et al., 2009b] empirically provide an extensive set of results showing that a sliding grid with SIFT can reach to state-of-the-art recognition rates. Before presenting our pseudo-code for the feature extraction module, we briefly discuss the SIFT descriptor.

3.1.1 SIFT Descriptor

Lowe, in his landmark paper [Lowe, 2004], presents a keypoint detector as well as an algorithm to create a descriptor for each keypoint. In this text we will focus on the descriptor assembly procedure, since we did not use the detector. The standard SIFT descriptor is a vector of size 128 floats which is created in two main steps:

1. Orientation assignment and,
2. Descriptor assembly.

In the first step, local gradients are computed in a patch of size $t \times t$, by default $t = 16$. The orientation $\theta(x, y)$ of each pixel patch is computed as:

$$\theta(x, y) = \arctan\left(\frac{I(x, y+1) - I(x, y-1)}{I(x+1, y) - I(x-1, y)}\right)$$

and its magnitude $m(x, y)$

$$m(x, y) = \sqrt{[I(x+1, y) - I(x-1, y)]^2 + [I(x, y+1) - I(x, y-1)]^2},$$

where I is the image in the closest scale where the patch is located.

The patch is subdivided in t regions and the local gradients are weighted by a Gaussian window. Each region has a histogram with 8 orientation bins and these histograms are formed by taking the weighted values around the patch. The dominant direction of each region correspond to the highest peak in histograms.

The 8 bins of all t histograms are concatenated forming the 128-vector, which after normalization, represents the SIFT descriptor. The whole procedure makes the descriptor scale and rotation invariant due to the histogram based on scale and a canonical orientation, and robust to illumination changes thanks to normalization.

Algorithm 1 SIFT_descriptors = Calculate_Feature ()

Require: *Images*, *grid_space*, *patch_size*, *Max_img_dim*

1: **for** $i = 1 \rightarrow$ Image.total **do**
2: $image = read\_image(i)$;
3: **if** *image.width* or *image.length* $>$ *Max_img_dim* **then**
4: image = im_resize(Max_img_dim); {perform a bicubic interpolation.}
5: **end if**
6: $grids = obtain\_patches(image.width$ or $image.length, patch\_size, grid\_space)$;
7: $SIFT(i) = find\_sift(grids)$;
8: **end for**

Algorithm 1 is responsible for the feature extraction. The process consists of reading the whole set of images, checking for images that are beyond the specified maximum size, resizing if necessary, followed by the image division into patches. From these patches, we obtain SIFT descriptors, that feed our unsupervised dictionary learning module.

3.2 Unsupervised feature learning

In linear generative models for images, each image x is represented by a linear combination of basis functions, that in our case are columns of a dictionary (D), by blending D_i columns with weight μ_i in the aim to infer the vector μ to better reconstruct the input x using a dictionary D, is given by:

$$x = \sum D_i \mu_i, \tag{3.1}$$

this equation can be solved, obtaining the representation μ, if the number of dictionary elements is equal in size to input. So applying the inverse of the dictionary to the input, result in μ,

$$\mu = D^{-1}x. \tag{3.2}$$

Sparse code methods present an overcomplete dictionary D (dictionary elements are much greater than input dimensionality), hence there are many solutions for μ and a sparsity regularization term of μ is used to reach a single solution. Models like that have been proposed in the literature, represented as a compound function:

$$T = R\left(x, D\mu\right) + s\left(\mu\right), \tag{3.3}$$

where R measure the reconstruction accuracy of the method and s the sparsity of μ. Almost all methods agree to use as reconstruction measure the squared $l2$ of the difference between the input signal and the model reconstruction $\|x - D\mu\|_2^2$. As sparsity measure, three ways were reported in the literature, $l0$ norm $s(\mu) = \lambda \left|\mu\right|_0$, $l1$ norm, applied in this work, $s(\mu) = \lambda \left|\mu\right|_1$ and a less usual form with logarithm $s(\mu) = log\left(1 + \mu^2\right)$.

The linear regression with $L1$ norm regularization on the coefficients is a problem known as Lasso, and can be solved using tools such as those provided by the recently published Sparse Modeling Library (SPAMS) [Mairal, 2011] or with a feature-

sign search algorithm [Lee et al., 2006].

A drawback of unsupervised feature learning when compared with the supervised counterpart is that, in unsupervised learning, an empirical risk (usually a convex loss) is minimized, so that the linear model fits some training data, and we expect the learned model to generalize well on new data points. However, due to possible small numbers of training samples and/or a large number of predictors, overfitting can occur, meaning that the learned parameters fit well the training data, but have a bad generalization performance. This issue can be solved by making a priori assumptions on the solution, naturally leading to the concept of regularization.

3.2.1 Dictionary Learning

We now move to the dictionary learning phase. The problem of learning a basis set can be formulated as a matrix factorization problem. More specifically, given a training set of signals $X = \{x^1, ..., x^n\}$ in $R^{m \times n}$, in our case a set of SIFT descriptors, one looks for a matrix D in $R^{m \times p}$, where p stands for the number of bases of our dictionary, such that each signal permits a sparse decomposition in D.

$$\underset{U,D}{\operatorname{argmin}} \sum_{i=1}^{n} \left\| x^i - \mu^i D \right\|^2 + \lambda |\mu^i|, \tag{3.4}$$

where U and D are convex sets and n is the number of features. Specifically $U = \{\mu^1 \dots \mu^n\}$ is the set of basis weight of each descriptor in $U \subseteq R^n$ and λ is a sparsity regularization term. The number of samples n is generally larger than the signal dimension m, which is $m = 128$ because of SIFT and $n \geq 200000$ for our validation tests. Usually, we also have $p \ll n$, based on samples $n = 200000$ and $p = 1024$, but each signal is reconstructed using few columns from D in its representation. Note that overcomplete dictionaries with $p > m$ are permitted.

Now we will present other matrix factorization algorithms to dictionary learning.

3.2.1.1 Vector quantization - Hard Assignment

Vector quantization or clustering, can also be seen as matrix factorization problem. Given n data vectors $X = \{x^1, ..., x^n\}$, the method looks for p centroids $\{d^1, ..., d^p\}$ and a binary assignment for each vector, which can be represented by a binary vector μ^i in $\{0, 1\}^p$ such that one single entry of μ^i is equals to 1 and all the rest are zero.

Once assignments have binary values, it use the terminology clustering with hard assignment.

With these assumptions in hand, we rewrite the problem:

$$\underset{D,U\epsilon\{0,1\}^n}{\operatorname{argmin}} \sum_{i=1}^{n} \left\| x^i - \mu^i D \right\|^2 s.t. \sum_{j=1}^{p} \mu_j^i = 1, \text{for all i } \epsilon\, [1, p]\,. \tag{3.5}$$

This is the same optimization problem performed by the K-means algorithm. Moreover, the K-SVD algorithm [Aharon and Bruckstein, 2006] for dictionary learning is presented by the author as a generalization of K-means, reinforcing the link between clustering and dictionary learning. Specifically, this method can be seen as a matrix factorization problem, where the columns of μ are forced to have a sparsity of one.

3.2.1.2 Vector quantization - Soft Assignment

Another possible view for vector quantization is to model data vectors as non-negative linear combinations of centroids that sum to one. The corresponding optimization problem is

$$\underset{D,U\epsilon R^n}{\operatorname{argmin}} \sum_{i=1}^{n} \left\| x^i - \mu^i D \right\|^2 s.t. \sum_{j=1}^{p} \mu_j^i = 1, \text{for all i } \epsilon\, [1, p] \text{ and } \mu \geq 0, \tag{3.6}$$

which is more similar to dictionary learning than vector quantization.

Yang et al. [2009b] explore this model to computer vision in BoF models, using dictionary learning instead of vector quantization for building visual dictionaries for object recognition.

3.2.2 Solving Dictionary Learning

Sparse coding provides a class of algorithms that learn basis functions from unlabeled input data, capturing their high-level features. Sparse coding can be learned from overcomplete basis set, where the number of basis are greater than the input dimensionality. Sparse coding can model inhibition between bases by sparsifying their activation, with biological similarity to the virtual cortex model [Olshausen and Field, 1997, 2004].

The dictionary learning algorithm [Lee et al., 2006] consists of iteratively alternating between U (coefficients) and D (bases). U and D are not convex simul-

taneously, but convex in U when D is fixed and vice-versa. The solving approach consists of optimizing the sparsity subset with a L1-regularized least square problem and the reconstruction part with a L2-constrained least square problem. We assume L1 penalty as the sparsity function, once L1 regulation is known to produce sparse coefficients and can be robust to irrelevant features [Ng, 2004].

3.2.2.1 Solving with D fixed

When the dictionary D is fixed Eq. 3.4 can be rewritten as:

$$\underset{\mu}{\operatorname{argmin}} \sum_{i=1}^{n} \underbrace{\left\| x^i - \mu^i D \right\|_2^2}_{\text{L2 constraint reconstrution}} + \underbrace{\lambda |\mu^i|}_{\text{is a regularization parameter}}, \tag{3.7}$$

where the L2 constraint denotes reconstruction and λ is a regularization parameter, that prevents overfiting. Considering only non-zero coefficients, this reduces Eq. 3.4 to a standard unconstrained quadratic optimization problem (QP) which can be solved analytically. The algorithm tries to search for signs of coefficients μ^i, given any such guess and systematically refines the guess if it turns out to initially incorrect.

3.2.2.2 Solving with U fixed

We will present how to solve the optimization problem when U is fixed. The problem is reduced to a least squares with quadratic constraint:

$$\begin{aligned} &\underset{D}{\operatorname{argmin}} \sum_{i=1}^{n} \left\| x^i - \mu^i D_k \right\|_F^2 \\ &\text{s.t. } \|D_k\| \leq 1, 1 \leq k \leq n. \end{aligned} \tag{3.8}$$

It is solved using a Lagrange Dual, since solving the dual uses significantly fewer optimization variables than the primal [Lee et al., 2006].

Several tests were performed by extracting SIFT descriptors from random patches to train the dictionary, and then iterating Eq. 3.7 and Eq. 3.8. Finally, after the dictionary is trained, the next step is the coding phase. For that we used a spatial constraint, detailed next.

3.3 Coding Process

Sparse coding has been presented as a good alternative to VQ, once it is more effective in feature quantization. Nevertheless, some limitations are observed in pure sparse coding methods. First, sparse coding methods are sensitive to the variance of features. Another limitation is that the $L1$ regularization can select quite different bases for similar patches to favor sparsity, in this way losing relationships between codes. Thus, spatial similarity can reinforce that analogous input signals will have similar column activations, resulting in similar codes.

To improve the relationship between local features and to impart more robustness to coding process, we introduce the SSC. In addition, SSC considers spatial similarity among features instead of just sparsity. We introduce this constraint to preserve consistency in sparse coding, for similar local features. Thus, SSC codes for local features are no longer independent.

Instead of coding with a sparsity constraint, we have chosen to use the spatial Euclidean similarity, based on the works of Wang et al. [2010] and Yu and Zhang [2009], which suggest that locality produces better signal reconstruction.

In VQ each descriptor is represented by a single base. However, spatial approaches use multiple basis in order to capture possible correlations between similar descriptors.

Other feature presented by the works of Wang et al. [2010] and Yu and Zhang [2009], which led us to opt for this type of coding, is that locality gives a higher probability of selecting similar basis for similar patches. This is different from a SC approach, in which regularization can select quite diverse basis for similar patches (see Figure 3.2).

Coding with spatial sparse coding, instead of sparse coding, transforms Eq. 3.4 into:

$$\underset{\mu}{\operatorname{argmin}} \sum_{i=1}^{n} \left\| x^i - D\mu^i \right\|^2 + \lambda \left\| d^i \odot \mu^i \right\|^2 \tag{3.9}$$

$$\text{s.t. } \mu^i = 1, \forall i, i = 1, .., n,$$

where $\odot$ is the element by element multiplication and d^i is the spatial similarity member computed as

$$d^i = dist(x^i, D), \tag{3.10}$$

and $dist(x^i, D)$ is a vector of Euclidean distances between each input descriptor x^i and the basis of the dictionary.

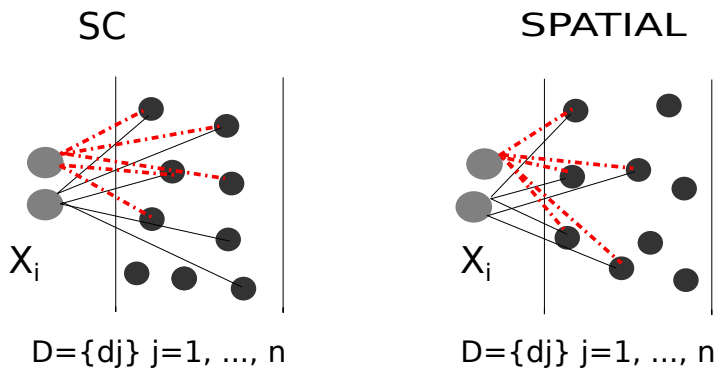


Figure 3.2: The SC shortcoming is that regularization can select different basis for similar patches, a problem that spatial constraint techniques are able to overcome. X_i represent the input features and D represents the dictionary. As it can be seen in this example, the spatial sparse coding selects the nearest basis in the dictionary.

Given these distances, we apply a KNN method that returns the N most similar basis for the given input, leading to a low computational demand to our coding process. The values of d^i are normalized by the max distance to adjust the range of possible represented numbers within the interval $(0, 1]$.

After coding each local feature, we perform a max pooling method to concatenate each code into a final image representation.

3.4 Pooling

Pooling is used to provide invariance to image transformation and robustness to noise and clutter in a way that preserves relevant information while removing spurious data.

To provide a better discriminative signature, we use high dimensional local features, which are SIFT descriptors obtained from a patch of 16×16 processed over a grid space of 6 pixels between each area. We decide to make use of dense regular grid, in opposition to interest points, based on the comparative evaluation made by [Fei-Fei and Perona, 2005], who present advantages of dense features for scene recognition tasks.

First of all, the final signature has a dimensionality defined by the function:

$$Size_{signature} = dictionary_{size} \times \sum_{i=1}^{T} (Pyramid\_scale(i))^2, \tag{3.11}$$

where T is the number of scales, in our case 3, and the pyramid scales are $[1, 2, 4]$, so that we have a dictionary size of 1024 basis. Our final signature is $1024 \times (1^2 + 2^2 + 4^2) = 21504$ vector elements per image.

Then the pooling process is applied to each scale and performs a maximization of SSC codes. Let U be the result of the application of Spatial Sparse Coding (Eq. 3.9) to a set of descriptor X, with a trained dictionary D. We build the final image signature with a function P

$$z = P(U), \tag{3.12}$$

where P is a pooling function defined on each column of U. Recall that each column of U corresponds to the response to the entire set of descriptor to a specific column of D. In our work, we choose as pooling function to maximize the SSC codes

$$z_j = max\left\{\left|u_{1j}\right|, \left|u_{2j}\right|, \left|u_{3j}\right|, ..., \left|u_{Mj}\right|\right\}, \tag{3.13}$$

where z_j is the j-th element of z, $u_{i,j}$ is the element from the column j and line i from U, and M is the number of local descriptors per area. We have chosen this function, despite average pooling once the works of Liu et al. [2011]; Boureau et al. [2010b] prove that max pooling presents state-of-the-art results. In addition, we perform max pooling in a spatial pyramid image representation. It is preferable, since max pooling under different locations and spatial scales provide more robustness to local transformations. Algorithm 2 summarizes the process.

The final representation is sent to some of our classification methods. First, we try the final signature with a proposed off-line method. We then use an online

Algorithm 2 SSC = Pooling $(X, D, Pyramid, Knn)$

$Ind = 0$;
$SSC\_codes = SSC(D, X, Knn)$; {Coding with spatial similarity}

for $Level = 1 \rightarrow$ Pyramid.levels **do**
  $Find\_local\_Feature(X, Level)$; {Find to which region of interest each local feature belongs}
  for $ROI = 1 \rightarrow$ Number of ROIs **do**
    $ind = ind + 1$;
    $B(:, ind) = max(SSC\_codes(id\_ROI))$;
  end for
end for
$SSC = B./norm\_l2(B)$;

approach that comprises the final version of the methodology.

3.5 Off-line learning method

We propose an off-line classification method based on Singular Value Decomposition (SVD), called Orthogonal Class Learning (OCL), that takes advantage of the high dimensionality of the feature vectors when compared to the number of feature examples, i.e., we have a set with t n-dimensional feature vectors where $n \gg t$. In this case, a base with only t components is used to represent new feature vectors. In addition, we obtain a new base for which new feature vectors are unit vectors, and pairwise orthogonal.

Consider, initially, we have h classes, each of which represented by k n-dimensional feature vectors so that we have $t = h \times k$ feature vectors. Let $f_1, f_2, \ldots, f_t$ denote feature vectors of all training data and let F denote the $n \times t$ matrix where columns are formed by the feature vectors, that is,

$$F = (f_1, f_2, ..., f_t). \tag{3.14}$$

Using SVD decomposition, we have that $F = USV^T$. Instead of forming new basis from columns of U as usual PCA, we use the fact that

$$V^T = (S^T S)^{-1} S^T U^T F, \tag{3.15}$$

and form a new base $A = (S^T S)^{-1} S^T U^T$ such that, in this new base, our new feature vectors are columns from V^T, being unit vectors and pairwise orthogonal. The advantage of this new representation is that, given object classes i and j and their feature vectors as matrices F_i and F_j formed with columns from F, we obtain new matrices $C_i = A.F_i$ and $C_j = A.F_j$, such that columns of C_i and C_j are pairwise orthogonal vectors. Figure 3.3 depicts the difference between PCA and OCL.

Finally, we construct our classifier based on the aforementioned observations. Given an object and its feature vector f, we obtain new feature vector $e = A \times f$ and the decision over the class S is given by

$$S = \underset{s}{\operatorname{argmax}} \|C_s^T e\|. \tag{3.16}$$

Algorithm 3 shows the training and testing process of the proposed method.

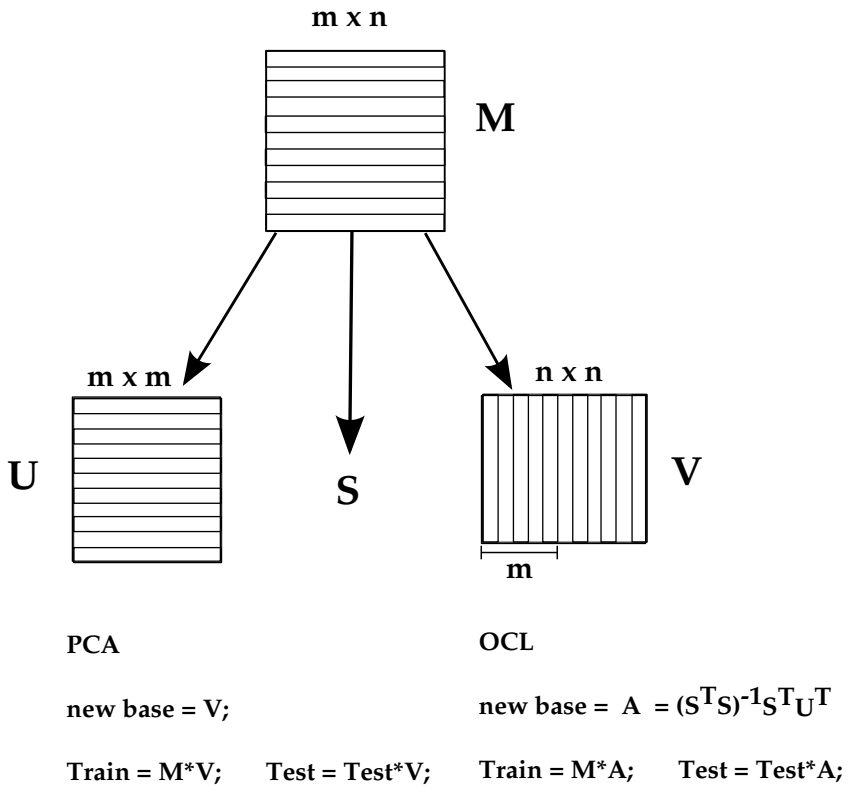


Figure 3.3: PCA vs OCL.

Algorithm 3 OCL classification(*label*)

Require: n_t, tr, ts

Training procedure
$[USV] = svd(tr)$;
$tr = V$; {V turns the new train set on new space}
$A = (S^T S)^{-1} S^T U^T$ {A turns new basis}

Test procedure
$ts = ts * A$ {pass the test set to new basis}
for $j = 1 \rightarrow$ all test examples **do**
  for $i = 1 \rightarrow$ each class cluster **do**
    $t1 = ts(j)$;
    $l = t1 * tr\left((i-1) * n_t + 1 : i * n_t, :\right)$;
    $n\left(i\right) = norm\left(l, 2\right)$;
  end for
  $[v, p] = max(n)$; {find the highest norm l2 of the classes}
  $label\left(j\right) = p$; {Prediction}
end for

3.6 Online learning method

The approach used in our final methodology was based on the Online LaRank [Bordes et al., 2008]. We have selected a LaRank multi-class solver. The LaRank algorithm is grounded in a randomized exploration, inspired by the perceptron algorithm [Bordes et al., 2007].

LaRank was selected as a solver, among other options, for the following reasons:

- Reaches equivalent accuracy values with less computational consumption when compared to other SVM solvers, like SVMstruct [Tsochantaridis et al., 2005];
- Generalizes better than perceptron-based algorithms;
- Achieves nearly optimal test error rates after a single pass over the randomly reordered training set.

The Online LaRank technique achieves the same test accuracy of batch optimization after a single epoch thanks to a reprocess step implementation over SMO-Optimization algorithm of Platt [1999].

In order to clarify how a SVM works we will give a overview of this method.

3.6.1 SVM

Support Vector Machines are based on the concept of hyperplane as boundaries. A hyperplane is used to split different class sets. Usually, a good separation is achieved by an hyperplane which has the greatest distance among the nearest data point of each class, called Maximum margin. According to Alpaydin [2010], from all possible linear decision function, the one that maximizes the margin of the training set will minimize the generalization error in a noise free dataset. Figure 3.4 exemplifies this case. Moreover, the training points that are nearest to the split function (shown by red circles) are named Support Vectors (SV).

Nevertheless, often data is not linearly distinguishable and, then, non-linear manifolds are needed to divide the data. For instance a XOR operation requires a radial function to be separable.

We choose SVM for our final learning method based on some properties:

- Overfitting can be controlled;

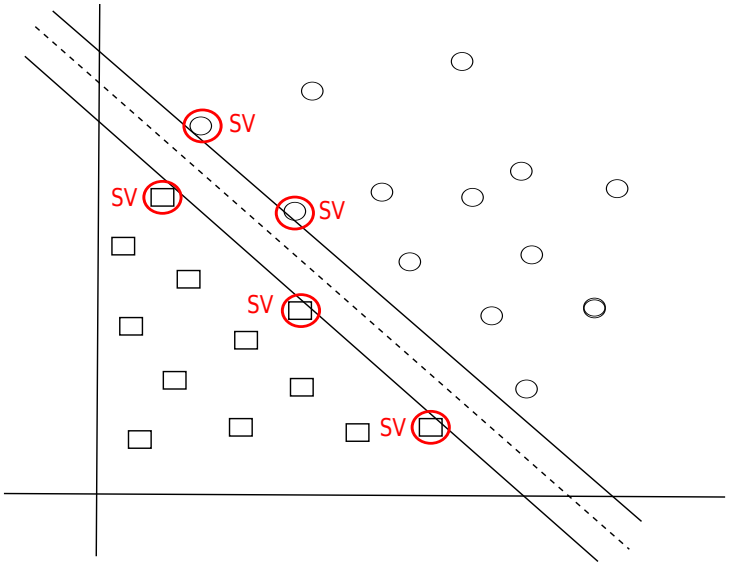


Figure 3.4: The Figure above presents an example of two classes where a hyperplane separates them in two classes (circles and squares).

- Linear SVM presents $O(n)$ in training (scale linearly with the size of the training set):
  - Efficiency to deal with extra large sets;
  - Works with high dimensional data;
  - No need for expensive computer resources.

Chapter 4

Method Validation

For evaluation purposes, we tested our method in two scenarios. First, we performed experiments using our technique with off-line classification methods, such as SVM, and with the OCL approach developed for this work. We then tested our final methodology (Sparse Spatial Coding) with an online learning algorithm, an SVM solver called OLarank [Bordes et al., 2008].

First, a parameter analysis is performed to show the effects of changing their values, also presenting when each parameter maximizes system performance. We first test our method with our off-line classification method, showing that only a sparse spatial constraint approach can lead to state-of-the-art results. We then show that the combination of sparse coding and locality with the correct online learning method can produce superior results.

4.1 Parameter Settings

One of the most critical setting for an object recognition method is the choice of a local feature to be used. In our experiments we chose SIFT [Lowe, 2004] due to its high accuracy on several object recognition tasks [Boiman, 2008; Yang et al., 2009b; Lazebnik et al., 2006]. Because of the dense grid sampling in the step for selecting regions of interest, our experiments use 6 pixels step between each region and a patch size of 16×16 pixels. During our experiments we tested the system with smaller step sizes, such as 4 and 2, as discussed in section 4.1.1.3. Our best results were with 4 pixels step; however, to make a fair comparison with the literature, which uses 6 pixels space. We report results with 6 pixels between patches and subsequently with 4. We also resize the images to 300×300 pixels.

We trained, by default, all the dictionaries for the tests with 1024 basis and

20000 random sample patches. The main parameter setting for the dictionary training is the sparsity/regularization which we empirically set to $\lambda = 0.30$ and the number of neighbors for the coding process, see section 4.1.1.2, obtaining the best classification results with $K = 5$ neighbors for the KNN.

We follow the same evaluation methodology of the compared works: all reported results are the average of 10 runs, with random selection of training and testing sets.

4.1.1 System Parameters Analysis

In order to analyze the behavior of the main parameters of the system (dictionary size, number of neighbours to coding, grid space and grid size), we perform tests to find which set of values maximizes the system accuracy and when these parameters saturate.

4.1.1.1 Dictionary Size

An investigation on the effects of dictionary sizes, as far as accuracy is concerned, was performed. On one hand, a small dictionary could not provide the required discriminative power, on the other hand, large dictionaries create antagonistic histograms for images of the same class, which will not match. Three sizes were tested, 1024, 2048 and 4096 basis. As it can be seen in Figure 4.1, our method presents a performance enhancement with larger dictionary sizes, but this performance boost starts to decrease when sizes reaches 4096 basis. The accuracy gain from 1024 to 2048 is 2.11%, but from 2048 to 4096 is just 0.74%. These results show that a policy of building even larger dictionaries becomes asymptotic both accuracy and memory efficiency.

4.1.1.2 Number of Neighbors

A parameter with a direct impact in the trade off between accuracy and performance is the number of neighbors K used for SSC. We report results with 3, 5, 10, 30 and 100 neighbors, respectively. We use the Caltech 101 dataset and present the results with 5, 10, 15, 20, 25 and 30 images per class for training.

Figure 4.2 presents our results for the difference among the lowest accuracy, in this case with 100 neighbors, and all options tested. As can be seen, usually smaller number of neighbors lead to better performance, in term of classification accuracy. Additionally, a smaller number of neighbors demand less processor and memory

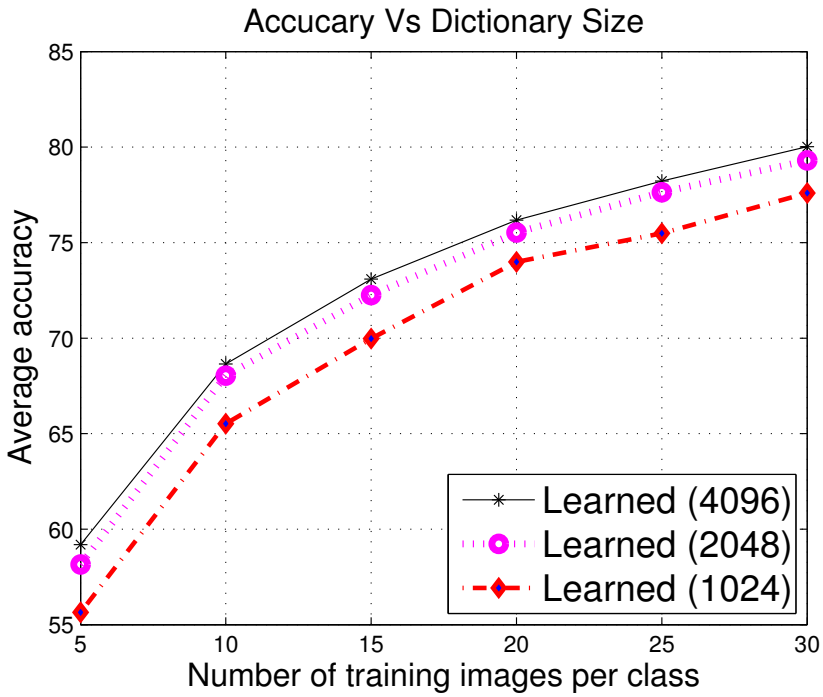


Figure 4.1: Performance of different sizes of dictionaries (Caltech 101). We can see that with the increase of dictionary size occur an improvement, but this performance boost has a limit, when sizes reach to 4096 basis.

resources. Nevertheless, with less than 5 neighbors we note that system accuracy starts to diminish. Another characteristic observed during our tests is that for large values of K, in our case 100, the accuracy also decreases, because of spurious signals added to the representation.

4.1.1.3 Grid Space

We also studied the effects of different grid spaces. Figure 4.3 presents the performance with 2, 4, and 6 pixels of space between each patch. As it can be seen, usually a small space among patches results in better classification accuracy. Differently from the number of neighbors, for which the lowest tested value presented a dip, grid spaces between 2 and 4 impose almost no difference; and with 2 pixels space, our system obtains twice the number of patches. So 4 pixels space present the best trade-off between accuracy and system performance.

4.1.1.4 Grid Size

In our experiments we observe that a way to increase our system's accuracy is possible by using a larger patch. Nevertheless, this approach has a limit, as observed

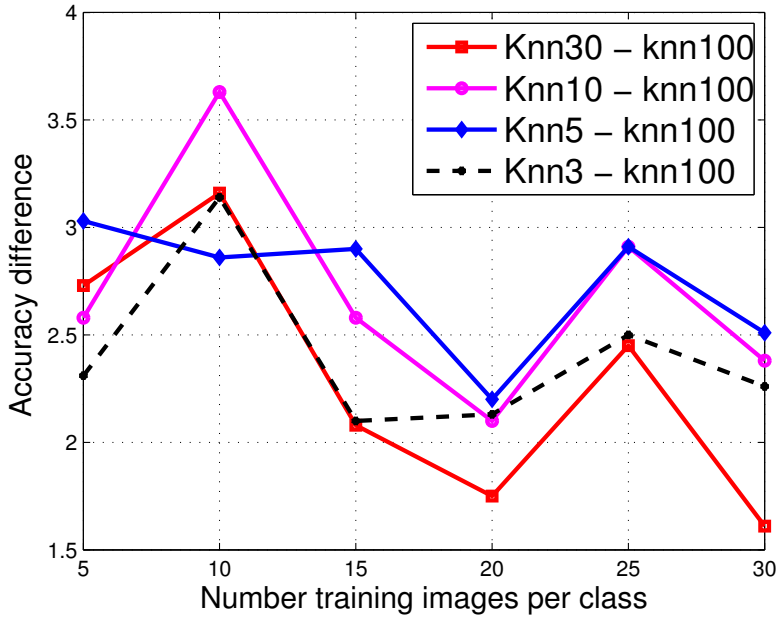


Figure 4.2: This Figure depicts the accuracy delta between the lowest value, 100 neighbours and the four other possibilities, 3, 5, 10 and 30 neighbours at Caltech 101. We can observe that our highest performance difference is obtained with 5 neighbours.

in Figure 4.4. Our results show that changing this parameter imparts some gain to system accuracy, but smaller than by changing grid space and dictionary size parameters.

4.1.2 Parameter Analysis conclusions

To provide additional analysis about the SSC method, we further evaluated its accuracy gain with respect to dictionary size, number of neighbors, grid space, and patch size. The work reports results using the Caltech 101 dataset.

We notice that among the analyzed parameters, dictionary size, number of neighbors, and grid space, present the highest gain, in term of accuracy. Furthermore, patch size presents the lowest improvement over the tested variables. Table 4.1 shows the obtains results.

As shown in Table 4.1, the learned dictionary of 4096 basis presents an average gain of 2.85% when compared with a dictionary of 1024 basis. A response like that ratifies the policy that, for our method, building bigger dictionaries can improve recognition rates. However, the SSC performance reaches stability when the dictionary size goes up to 4096.

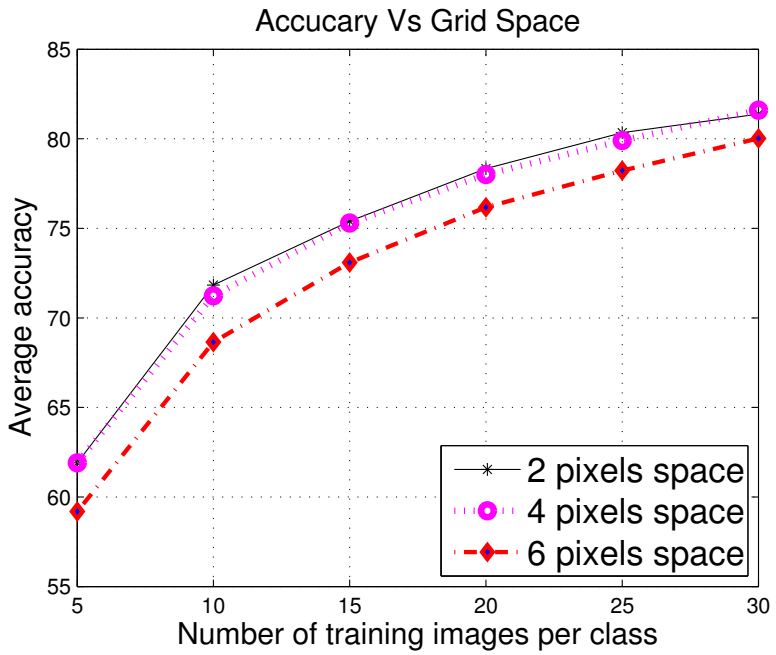


Figure 4.3: Performance under different grid spaces (Caltech 101). Our tests identify that with 2 and 4 pixels between patches our approach reaches stability, with the highest obtained recognition rate at Caltech 101. On the other side, with a grid space of 6 pixels, our system recognition accuracy start to decline, in term of accuracy. It is relevant to mention that for these tests our dictionary size was 4096.

Table 4.1: System variables gain. First, is the analyzed parameter, followed by the difference between the greatest and the lowest recognition rate obtained.

Variable	5
Dictionary $4096 - 1024$	2.85%
Number of neighbors $5 - 100$	2.71%
Grid space $4 - 6$	2.30%
Patch size $16 - 32$	0.41%

We additionally present the accuracy gain of 2.71%, in respect to the number of neighbours to our coding module. This finding give us a clear perception that small number of neighbours are the preferable choice, in terms of accuracy and computational demand. As can be seen in Table 4.1 our highest value was achieved with 5 neighbors and our lowest with 100.

Another analyzed parameter is grid space. Our results show that unlike other works Wang et al. [2010]; Yang et al. [2009b]; Lazebnik et al. [2006] that use 6 pixels between each patch, the best recognition performance were with 4 pixels, 2.30% higher than with 6. However, such gain is followed by a penalty to system compu-

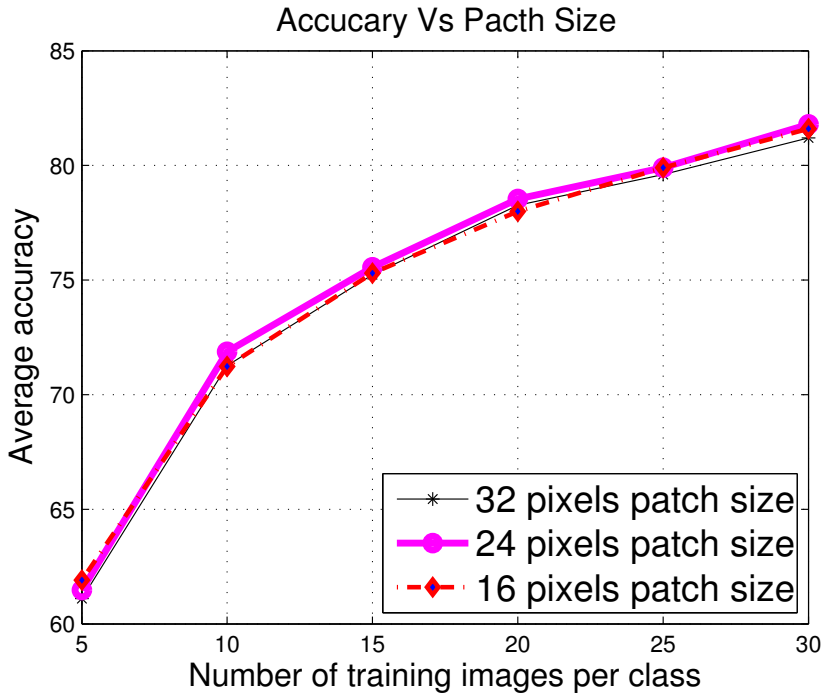


Figure 4.4: The figure shows the system response for several grid sizes, with a dictionary size of 4096. It is clear that such parameter has a smaller impact on system accuracy, when compared to grid space or dictionary size parameters. However, during our tests, we realize that a patch size between 16 and 24 pixels present the best results. Additionally, we note that above 24 pixels in size, our system accuracy starts to decline.

tational demand, due to a larger number of patches.

We also investigate the effects of patch size in our technique. Among the other parameters, patch size has presented the lowest gain (0.41%). We believe that one possible reason is the pooling over multiple patch scales, so max pooling over sparse spatial codes can capture the noticeable properties of local regions that are irrelevant to the size of local patches.

4.2 Evaluation of Offline Methods

We first test how the SSC method works in off-line standard classification method such as the Linear SVM and Random Forests. Additionally, we also present the obtained results with a new classification algorithm, called OCL. The dataset used was the Caltech 101, which consists of 101 classes with broad shape variation. For evaluation purposes, we compare our results with the following methods: LLC [Wang et al., 2010], ScSPM [Yang et al., 2009b], and NBNN [Boiman, 2008]. Table

4.2 presents the results. As it can be seen, our technique presents better performance, specially when combined with the OCL technique, which outperforms the best results reported in the literature.

Table 4.2: Off-line methodologies. We see that applying the training and coding steps with our OCL approach outperform the previous reported methods.

N. train	5	10	15	20	25	30
NBNN Boiman [2008]	-	-	65.0±1.1	-	-	70.4
ScSPMYang et al. [2009b]	-	-	67.0±0.4	-	-	73.2±0.5
LCCWang et al. [2010]	51.1	59.7	65.4±0.4	67.7	70.1	73.4
Ours(RF)	33.7±1.4	41.5±1.1	46.1±0.9	49.8±0.7	52.0±0.8	54.2±0.6
Ours(SVM)	46.3±0.6	56.5±0.4	62.0±0.4	65.3±0.9	67.6±0.5	71.1±0.5
Ours(OCL)	**56.7±0.9**	**65.2±1.0**	**69.0±0.7**	**71.7±0.7**	**73.6±0.5**	**75.7±0.5**

In addition, we also verify if it was necessary to use a vector with a number of components equal to the number of training samples. Figure 4.5 present a test with our learning approach at Caltech 101 dataset with 30 images per class for training. In this experiment we tested how our method behaves with the number of components constituting the feature signature ranging from 100 to 3000 components, 3060 is the number of training images. Thus, we can clearly see that above 2000 components our method present a limited increase in accuracy, showing that from a feature vector of 21504 elements, which is the size of our signature in a dictionary of 1024 basis, see Section 3.4. Our Method reduce this vector to 2000 elements, less than 10% of the original size, with a slight accuracy drop (less than 0.1%).

4.3 Online Learning Evaluation

Another test is related to the comparison of our methodology against other online learning classifiers. We conducted tests with SSC and with three other online methods: ORF (Online Random Forest) [Saffari et al., 2009], OMCGB (Multi-Class Gradient Boost), and OMCLPB (Online Multi-Class LPBoost) [Saffari et al., 2010]. Table 4.3 reports the average results of 10 runs with randomly chosen samples for training and for testing from the Caltech 101 dataset. Each algorithm run for 10 epochs.

To complete the experimental tests, we analyze the behavior of each online classifier over 10 epochs. Figure 4.6 shows how each method benefits from revisiting the training set. We can see that both our method and OMCLP reach stability with only 3 epochs, nevertheless our technique presents an accuracy 21% better when compared with the second best algorithm. It is relevant to note that after just one

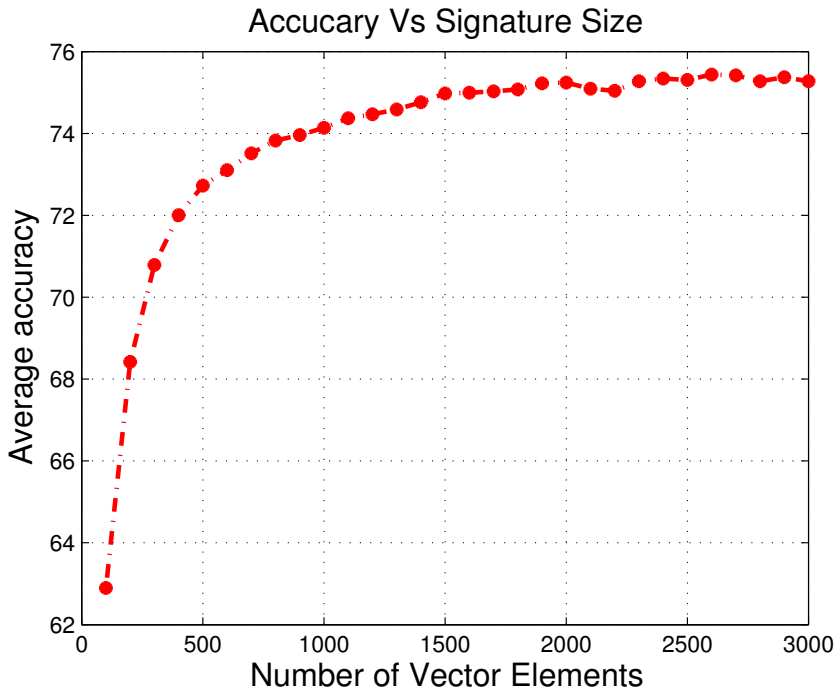


Figure 4.5: This figure show accuracy results with the number of components of the signature ranging from 100 to the total number of training examples 3060 at Caltech 101. It is clear that above 2000 components our method reach to stability.

Table 4.3: Online Learning Results. The results show a clear advantage of our method over other online learning techniques by a margin exceeding 21%.

training images	Ours	ORF	OMCGB	OMCLPB
5	**55.6±1.0**	43.2±1.2	43.6±1.5	43.9±1.6
10	**65.5±0.7**	47.7±0.9	47.8±1.5	48.4±0.7
15	**70.0±0.8**	49.8±0.8	50.0±0.7	51.1±0.9
20	**74.0±2.1**	41.9±3.8	52.4±0.8	53.2±0.9
25	**75.5±0.6**	53.8±0.4	53.2±0.5	54.8±0.5
30	**77.6±0.4**	55.5±0.5	55.1±0.8	56.5±0.8

epoch the algorithm already outperforms the best results of previously published works.

4.4 Caltech 101

The Caltech 101 contains 9144 images divided into 102 classes, 101 object classes and one background class, with broad shape variation, see Figure 4.7. The per-class number of images range from 31 to 800. To make the comparison as fair as

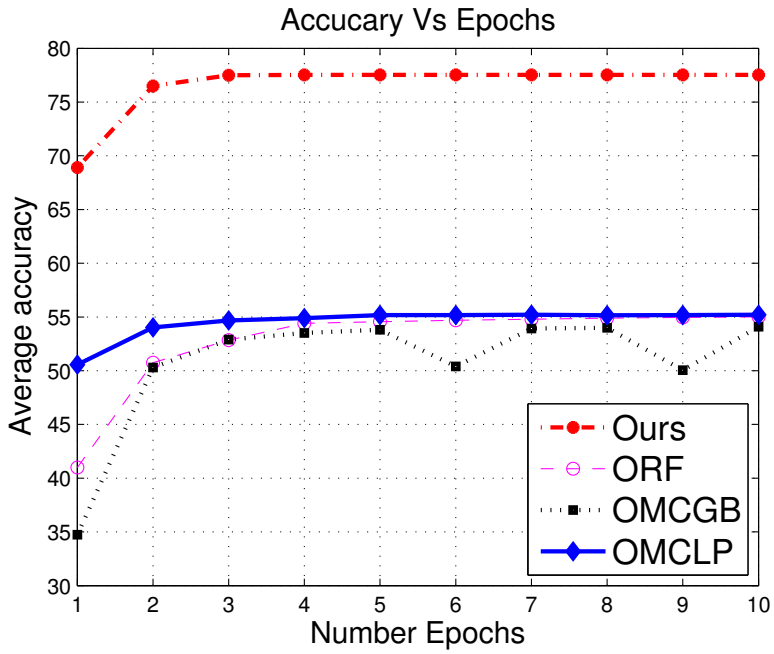


Figure 4.6: Accuracies obtained under 10 epochs, average of 10 runs, on Caltech 101 with 30 images per category for training. Our method and OMCLP need 3 epochs to stabilize, but our method reaches to state-of-the-art results with a single epoch.

(a) chair (b) camera (c) headphone

(d) laptop (e) revolver (f) umbrella

Figure 4.7: Caltech 101 dataset class samples, for example chair, camera and headphone in the first row and laptop, revolver and umbrella below them.

possible, we follow the same steps of Lazebnik et al. [2006]. We run ten times with different randomly selected training and test images, and the average recognition rate is recorded for each run.

Table 4.4: Recognition results on Caltech 101. The results can be directly compared with the literature, since all the works use the same methodology to perform the experiments. As shown in boldface, our method has superior recognition rates when compared with all the single feature approaches found in the literature. Furthermore, we report results with a dictionary of 4096 basis with 6 and 4 grid space, shown at Ours$^{\mathbf{4096\text{-}6}}$ and Ours$^{\mathbf{4096\text{-}4}}$ lines. For all the cases, our work largely outperforms the best among the current published techniques.

Number of training samples	5	10	15	20	25	30
Malik [Zhang et al., 2006]	46.6	55.8	59.1	62.0	-	66.2
KSPM [Lazebnik et al., 2006]	-	-	56.4	-	-	64.4 $\pm$ 0.80
NBNN [Boiman, 2008]	-	-	65.0 $\pm$1.1	-	-	70.4
ML+CORR [Jain et al., 2008]	-	-	61.0	-	-	64.1 $\pm$ 1.1
Boureau [Boureau et al., 2010a]	-	-	-	-	-	75.7 $\pm$ 1.1
Coates [Coates and Andrew, 2011]	-	-	-	-	-	72.6 $\pm$ 0.9
SRC [Wright et al., 2009]	48.8	60.1	64.9	67.7	69.2	70.7
K-SVD [Aharon et al., 2006]	49.8	59.8	65.2	68.7	71.0	73.2
D-KSVD [Zhang and Li, 2010]	49.6	59.5	65.1	68.6	71.1	73.0
ScSPM [Yang et al., 2009b]	-	-	67.00	-	-	73.20
LCC [Wang et al., 2010]	51.15	59.77	65.43	67.74	70.16	73.44
LC-KSVD [Jiang et al., 2011]	49.6	63.1	67.7	70.5	72.3	73.6
Ours	**55.6$\pm$1.0**	**65.5$\pm$0.7**	**70.0$\pm$0.8**	**74.0$\pm$2.1**	**75.5$\pm$0.6**	**77.6$\pm$0.4**
Ours$^{\mathbf{4096\text{-}6}}$	**59.2$\pm$1.1**	**68.6$\pm$0.6**	**73.1$\pm$0.7**	**76.2$\pm$0.5**	**78.2$\pm$0.4**	**80.0$\pm$0.3**
Ours$^{\mathbf{4096\text{-}4}}$	**61.9$\pm$1.2**	**71.2$\pm$0.5**	**75.3$\pm$0.7**	**78.0$\pm$0.5**	**79.9$\pm$0.4**	**81.6$\pm$0.5**

Table 4.4 shows the results and the comparison with other recent proposed methods, indicating the superior responses obtained with our approach. This results confirms the hypothesis of Coates and Andrew [2011], in which, datasets with a low number of available training examples, such as 30 images in 80, show that sparse representation are far superior than soft-thresholded ones.

Our results, shown in Table 4.4, were based on a dictionary of 1024 basis; however, during the experiments we have tested different dictionary sizes, for instance 1024, 2048 and 4096. Our highest scores were obtained with a dictionary size of 4096 and with a grid space of 4 pixels. As it can be readily seen from Table 4.4, the results represent a significant improvement over recognition rates, once our work, using a single feature approach, reached 81.6% of accuracy at Caltech 101. Related to the improvement of our work, Table 4.5 present the gain over the compared methods. It can be seen that we outperform the previous techniques by a margin ranging from 10.8%, Caltech 101 with 5 images per class for train, to 5.9 to 30 images per class.

Table 4.5: Recognition gain over previous reported methods.

Number of training samples	5	10	15	20	25	30
Ours	**10.8%**	**8.1%**	**7.6%**	**7.5%**	**7.6%**	**5.9%**

Additionally, we also compare our results with multiple feature approaches, see Table 4.6. It is a unfair comparison with our technique, because such approaches

like OBSCURE [Orabona et al., 2012], GS [Yang et al., 2009a], LP-Beta [Gehler and Nowozin, 2009b], Holistic [Li et al., 2010] and Todorovic [Todorovic and Ahuja, 2008], employ several sources of information and the SSC method just use appearance.

Table 4.6: Recognition results on Caltech 101 (Multiple Features)

Number of training samples	5	10	15	20	25	30
OBSCURE [Orabona et al., 2012]	50.1±0.8	63.2±0.7	68.8±0.6	72.9±0.8	75.2±0.9	77.8±0.7
Todorovic [Todorovic and Ahuja, 2008]	-	-	72.0	-	-	83.0
Holistic [Li et al., 2010]	60.9	-	74.7	-	-	81.9
LP-Beta [Gehler and Nowozin, 2009b]	59.5±0.7	69.2±0.4	74.6±1.0	77.6±0.3	79.6±0.4	82.1±0.3
GS [Yang et al., 2009a]	-	65.1	73.2	**80.1**	**82.7**	**84.3**
Ours$^{4096\text{-}4}$	**61.9±1.2**	**71.2±0.5**	**75.3±0.7**	78.0±0.5	79.9±0.4	81.6±0.5

Table 4.6 presents the results of our method against several multi-feature approaches. Specially, with GS [Yang et al., 2009a] and LP-Beta [Gehler and Nowozin, 2009b] that constitute the state-of-the-art for multiple feature at Caltech 101. Both use 5 different features to reach to these results, for instance GS extract dense color SIFT, dense SIFT [Lazebnik et al., 2006], Self-Similarity (SS) [Shechtman and Irani, 2007b], Pyramid histogram of oriented gradients (PHOG) [Bosch et al., 2007] and Garbor features.

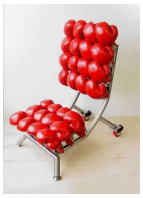
We can observe that multiple cues approaches present a clear superiority for 20, 25 and 30 images per class for training. Although, for 5, 10 and 15 we outperform the top score multi-feature techniques. The results again confirm the assumption of Coates and Andrew [2011], that for a low number of available samples, sparse representation are far superior than other techniques.

4.5 Caltech 256

Caltech 256 is an extension of Caltech 101 with 29780 images with 257 categories, including background, see Figure 4.8. This dataset presents additional challenges when compared with Caltech 101, once intra-class variance is larger and object locations are quite different.

Tests were performed with 15, 30, 45 and 60 images for training and the remaining images were used for testing. Each category contains at least 80 images. Table 4.7 lists our results and those reported in the literature for a dictionary of 1024 basis. It can be easily seen that as far as accuracy is concerned, our method outperforms the existing techniques.

To extend the analysis we also tested our method with a dictionary of 4096 basis and 4 pixels space, in this way it is fare to make a direct comparison with



(a) box glove

(b) ipod

(c) baseball bat

(d) box glove

(e) ipod

(f) baseball bat

Figure 4.8: Caltech 256 dataset. These three pairs of classes (box glove, ipod and baseball bat) illustrate the high intra-class variance of Caltech 256.

Table 4.7: Average accuracy on the Caltech 256 dataset. Our method clearly presents superior accuracy results when compared with several other high performance methods, specially when compared with a method which applies spatial constraint to the coding phase (LScSPM).

N. training	15	30	45	60
ScSPM [Yang et al., 2009b]	27.3±0.5	34.0±0.3	37.4±0.5	40.1±0.9
LScSPM [Gao et al., 2010]	30.0±0.1	35.7±0.1	38.5±0.3	40.4±0.3
Ours	**30.6±0.3**	**37.0±0.3**	**40.7±0.1**	**43.5±0.3**

IFK [Perronnin et al., 2010], LLC [Wang et al., 2010] and Convolutional Restricted Boltzmann Machine (CRBM) [Sohn et al., 2011] (see Table 4.8).

Table 4.8: Average accuracy on the Caltech 256 dataset with a dictionary of 4096 basis. Our method leads the performance in most of the cases in this dataset. Although, with 30 images per class for training we lost, by a small margin, to CBRN [Sohn et al., 2011].

N. training	15	30	45	60
LLC [Wang et al., 2010]	34.36	41.19	45.31	47.68
IFK [Perronnin et al., 2010]	34.7±0.2	40.8±0.1	45.0±0.2	47.9±0.4
CRBM [Sohn et al., 2011]	**35.09±0.24**	**42.05±0.27**	45.69 ± 0.31	47.94 ± 0.42
Ours	**35.07±0.3**	41.81±0.29	**45.93 ±0.14**	**48.97 ±0.40**

4.6 Corel Datasets

Corel 1000, 5000 and 10000 datasets were originally created for Content-Based Image Retrieval (CBIR). However, we believe that they are of particular interest for our tests, since they have a large number of images and they are based on natural images including those from outdoor scenes. The same procedure used in the Caltech 101 experiments was applied to these tests. We chose to perform experiments using 50 images for training and 50 for test.

In Table 4.9 we present the results on Corel datasets, with our approach compared against SMK, LCC, ScSPM and LScSPM. We highlight the greater recognition rate of our technique. Furthermore, this table demonstrates the superiority of the technique at Corel 5000 and 10000, even when compared with a method which applies spatial constraint to the coding phase as Wang et al. [2010]. One can observe that our method attains similar results to state-of-the-art on the Corel 1000 dataset.

Table 4.9: Results in Corel datasets. For Corel 1000 our work reach to the same accuracy of Gao et al. [2010]; however, for Corel 5000 and 10000 the proposed approach outperform the other evaluated methods.

Methods	Corel 1000	Corel 5000	Corel 10000
SMK [Lu and Ip, 2009]	77.9	-	-
LCC [Wang et al., 2010]	-	76.5±0.7	67.7±0.5
ScSPM [Yang et al., 2009b]	86.2±1.0	77.1±0.5	68.4±0.3
LScSPM [Gao et al., 2010]	**88.4±0.8**	-	-
Ours	**88.4 ± 0.8**	**78.2±0.6**	**69.3±0.4**

4.7 MIT 67 Indoor

We also compare our method with the challenging scene dataset MIT 67. This dataset constitutes the largest publicly available benchmark base to scene recognition, with 67 classes and 15620 images. The dataset presents large in-class variability and few distinctive attributes when compared with Scene-15 [Lazebnik et al., 2006], see Figure 4.9. The accuracy metric is the same of other experiments. However we follow the same experimental setup of Quattoni and Torralba [2009], which uses 80 images per class for training and 20 images per class for testing.

Figure 4.10 compares our results with other works reported in literature, such as GIST [Quattoni and Torralba, 2009], MM-scene [Zhu et al., 2010], CENTRIST [Wu and Rehg, 2011], Object Bank [jia Li et al., 2010] and GG [Nakayama et al., 2010].

(a) airport

(b) corridor

(c) bookstore

(d) greenhouse

Figure 4.9: MIT 67 Indoor examples of image classes with high in-class variability and few distintive attributes (corridor class).

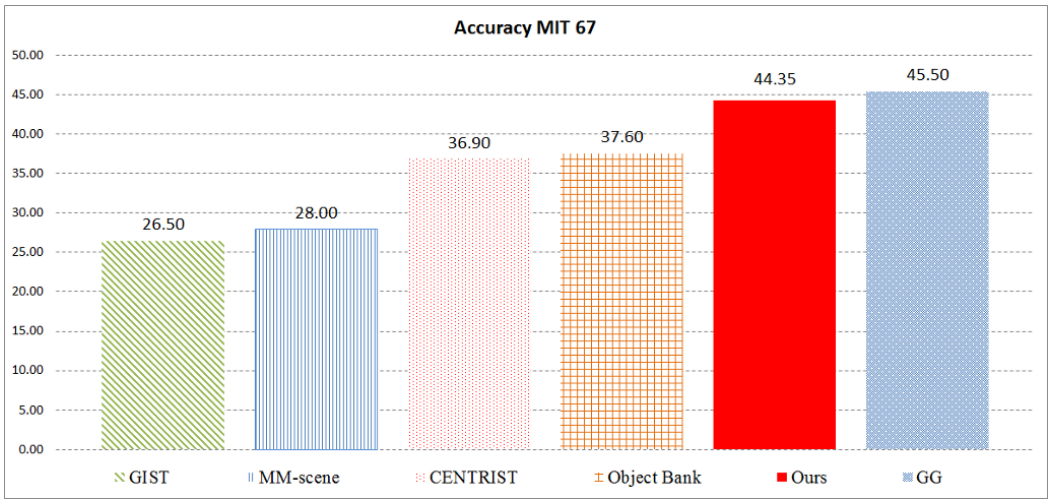


Figure 4.10: Average classification rates for MIT 67 indoor scene dataset, with the exception of our result and Nakayama et al. [2010], all the other methods present the accuracy of a single run. Our method reaches high performance results, although inferior to GG, which presents 45.5% of recognition with a standard deviation of 1.1, against our method which presents 44.35% of accuracy with a standard deviation of 0.90.

Differently from the compared works, we do not apply any annotation to the images, to show the superior results obtained by our method. One can see that our method, using a single feature to recognition, presents performance that is superior

to algorithms specially developed for this purpose: 44.35% against 36.9% obtained by Wu and Rehg [2011]. However, inferior to the highest reported result in literature of 45.5 by Nakayama et al. [2010].

4.8 Statistical Analysis

Additionally to the established way to compare the previous results, we propose to employ a quantitative method to provide statistical reliability to our comparisons, since the simple mean comparison is not accurate enough to draw precise conclusions about which method has better performance. Based on the provided values by the literature (mean and standard deviation), we choose a confidence interval as metric to compare our results with the state-of-the-art works on these datasets. We are the first work to propose such kind of comparison metric to Caltech 101, Caltech 256, Corel 5000, Corel 10000 and MIT Indoor 67 datasets, to the best of our knowledge.

To compute the confidence interval, we approximate the samples to a normal distribution and based on the low number of samples, in our case $n = 10$, we employ a t-student distribution. From this assumptions and with two means (X_1, X_2) and standard deviations (S_1 S_2), we calculate the confidence interval:

$$X_1 - X_2 + t_{[1-\alpha;V]}S \tag{4.1}$$

where $\alpha = 0.05$ is the significance level and V is the degree of freedom:

$$V = \frac{\left(\frac{S_1^2}{n} + \frac{S_2^2}{n}\right)^2}{\frac{1}{n+1}\left(\frac{S_1^2}{n}\right)^2 + \frac{1}{n+1}\left(\frac{S_2^2}{n}\right)^2} - 2. \tag{4.2}$$

With the confidence interval we check if the interval includes zero. If the interval includes zero there is no difference between the two systems. Although, if zero is not part of the interval, then the systems are statistically contrasting.

The first comparison was between our previous results on [Oliveira et al., 2012] and our new values obtained from the parameter tunning. Table 4.10 shows in bold which results are statistically superior on Caltech 101 with single feature approaches. As can be seen the tuned system outperform the previous results in all instance, with a confidence of 95%.

The second comparison was between SSC and the state-of-the-art approach [Gehler and Nowozin, 2009b] which employ multiple features to recognition on Cal-

Table 4.10: Statistical analysis Caltech 101 single feature

Number of training samples	5	10	15	20	25	30
SSC [Oliveira et al., 2012]	59.2±1.1	68.6±0.6	73.1±0.7	76.2±0.5	78.2±0.4	80.0±0.3
SSC (tunned)	**61.9±1.2**	**71.2±0.5**	**75.3±0.7**	**78.0±0.5**	**79.9±0.4**	**81.6±0.5**

tech 101. We excluded [Yang et al., 2009a] from our comparison based on the number of samples to compute the mean value, that is 5 instead of 10 (default value) and on the lack of standard deviation values, making impossible the computation of the confidence intervals. Table 4.11 presents in bold when the recognition rate is statistically meaningful. From this experiment, we can observe that for 5 and 10 images per class for training, SSC outperform the best multiple feature so far. However, for 15, 20 and 25 our method present recognition rates that are statistically similar to the state-of-the-art and for 30 our method presents inferior performance.

Table 4.11: Statistical analysis Caltech 101 multiple feature

Number of training samples	5	10	15	20	25	30
LP-Beta [Gehler and Nowozin, 2009b]	59.5±0.7	69.2±0.4	74.6±1.0	77.6±0.3	79.6±0.4	**82.1±0.3**
SSC	**61.9±1.2**	**71.2±0.5**	75.3±0.7	78.0±0.5	79.9±0.4	81.6±0.5

We also evaluate our method with the CRBM method [Sohn et al., 2011] on Caltech 256. Table 4.12 depicts the obtained results, showing that our method is at least statistically similar to the state-of-the-art, for 15 and 30 images per class for training. Nevertheless, for 45 and 60 our approach obtain results that are statistically differentiable from CRBM with a significance level of 95%.

Table 4.12: Statistical analysis Caltech 256

N. training	15	30	45	60
CRBM [Sohn et al., 2011]	35.09±0.24	42.05±0.27	45.69 ± 0.31	47.94 ± 0.42
SSC	35.07±0.3	41.81±0.29	**45.93 ±0.14**	**48.97 ±0.40**

Table 4.13: Statistical analysis in Corel datasets

Methods	Corel 5000	Corel 10000
LCC [Wang et al., 2010]	76.5±0.7	67.7±0.5
ScSPM [Yang et al., 2009b]	77.1±0.5	68.4±0.3
Ours	**78.2±0.6**	**69.3±0.4**

Table 4.13 presents the quantitative analysis of the Corel datasets. In bold are shown results with proved dominance. In Corel 5000 and Corel 10000 our results outperform other sparse coding methods with 95% of confidence.

(a) sunny (b) cloudy (c) night

Figure 4.11: Lighting conditions of COLD dataset.

To further evaluate the results from MIT Indoor 67 dataset, we also quantitatively compare our result with the best from literature [Nakayama et al., 2010]. This test ratifies the first experiments that our result is inferior to the state-of-the-art. See numerical values at Appendix A.

4.9 COLD Dataset

Place recognition experiments were carried out on the COLD dataset [Pronobis and Caputo, 2009]. This dataset was built to evaluate vision-based place recognition methods for mobile platforms in realistic settings and to test robustness over different kinds of variations. The dataset is divided into three labs (Ljubljana, Freiburg and Saarbrücken). Each laboratory has two parts, called A and B, with two explored paths, standard and extended. Each path has three different lighting conditions (night, sunny and cloudy) acquired in several times, see Figure 4.11. The acquisition rate was 5Hz. The dataset presents omnidirectional and perspective images, but all the performed experiments were done with perspective images.

The experiments on the COLD dataset aim to recognize a room, previously seen during training, when imaged under different illumination settings and/or different time. To directly compare with literature, we follow the same experimental methodology of Ullah et al. [2008]. For each experiment, training is done in one sequence acquired in the same laboratory, and testing is performed on sequences acquired under various conditions. Results were averaged for all permutations of the training and testing. This dataset was selected based on its capability to verify robustness to changes in illumination and pose.

From the available labs, we selected the Ljubljana based on its challenging environment, confirmed by the low recognition rates [Guilleaume et al., 2011]. Figure 4.12 presents our results on this dataset.

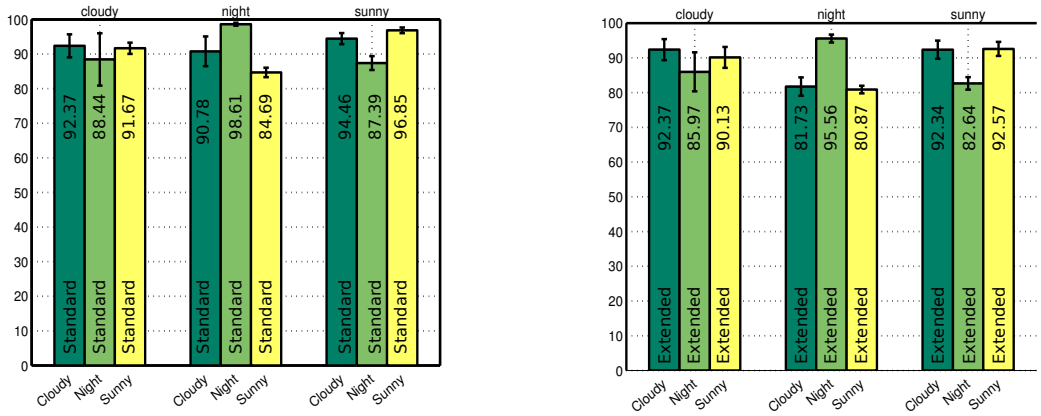


Figure 4.12: Average results on COLD-Ljubljana dataset (a) standard sequence and (b) extended sequence. The illumination condition used for training is shown on top of each group. The bottom axis represent the testing illumination conditions. The vertical axes is the average classification rate.

We can observe that our method achieves state-of-the-art results when trained with equal training and testing illumination conditions (see Table 4.14). This test was performed to observe how our method tackles other kinds of variability, like human activity or changes in view points due to robot motion. Figure 4.12 also

Table 4.14: COLD recognition rates for equal illumination conditions. For each test, the best results are in bold. The results obtained with our method outperform previous ones by 5%.

	Ours	Guilleaume et al. [2011]	Ullah et al. [2008]
Ljubljana - Standard	**95.94**%	90.7%	90.4%
Ljubljana - Extented	**93.50**%	87.7%	85.8%

shows that the recognition rates for the standard part of the dataset are generally superior to the extended part. This can be explained by the characteristic that the extended sequences contain a larger number of classes, making the problem harder.

Another important result is related to the robustness of our method to different lighting conditions. In all experiments, our method largely outperforms those reported in the literature, as detailed in Table 4.15.

Table 4.15: Comparison of the results obtained with our method and with the best reported recognition rates found in the literature on the COLD Ljubljana dataset, where [1] is [Ullah et al., 2008] and [2] is [Guilleaume et al., 2011]. Each result and the respective work are identified by the superscript of the related reference.

Ljub-Std	Train					
	cloudy		night		sunny	
	Prev. Best	Ours	Prev. Best	Ours	Prev. Best	Ours
Cloudy	85.88[1]	**92.37**	84.69[1]	**90.78**	86.12[2]	**94.46**
Night	83.46[1]	**88.44**	96.64[2]	**98.61**	82.85[1]	**87.39**
Sunny	88.12[2]	**91.67**	81[1]	**84.69**	95.61[2]	**96.85**
Ljub-Ext	Train					
	cloudy		night		sunny	
	Prev. Best	Ours	Prev. Best	Ours	Prev. Best	Ours
Cloudy	80.8[2]	**92.37**	73.37[1]	**81.73**	79.82[1]	**92.34**
Night	77.77[1]	**85.97**	91.86[1]	**95.56**	74.95[1]	**82.64**
Sunny	85.02[2]	**90.13**	71.61[1]	**80.87**	92.25[2]	**92.57**

4.10 VPC

The Visual Place Recognition (VPC) dataset consist of images taken from 6 different homes where each of them has 1 to 3 floors. Images were obtained using a high definition camcorder with a resolution of 1280×720. During the capture procedure white balance and auto-focus are enabled. The dataset was built by a rolling tripod to mimic a robot. The data set is significantly challenging because many frames do not capture characteristic data of the room, containing only walls or just some closed views, which is similar to a robot moving around. The dataset contains 11 classes, however only 5 of them appear in all houses, bedroom, bathroom, kitchen, living room and dining room.

Our experimental methodology follows [Wu et al., 2009] where we trained our system with data from 5 houses and test it on the remaining one. The frames that not belong to the bedroom, bathroom, kitchen, living room and dining room classes are omitted.

Table 4.16 presents the obtained results by SSC and by Wu et al. [2009] method. We can observe that we lost in 4 of 5 categories, superior just in Bed class, obtaining an average result of 43.21 against 45.62. One possible reason to Wu's higher recognition rate can be the use of a Bayesian filter. This filter integrate information from many frames, maintain a belief and using a bayesian filter for updating category beliefs [Wu et al., 2009]. Thus, the categorization accuracy without the filter reach to

41.87 and with it to 45.62.

Table 4.16: Recognition rates from VPC dataset dataset, excepting by the Bed class, we lost in all other classes. Our overall average to the 5 categories is 43.21 against 45.62 from Wu et al. [2009].

	Bed		Bath		Kitchen		Living		Dining		Average	
	VPC	Ours	VPC	Ours	VPC	Ours	VPC	Ours	VPC	Ours	VPC	Ours
Home 1	**75.76**	74.51	**80.04**	71.66	12.03	**12.45**	**43.90**	6.1	**11.15**	1.82	**44.58**	33.30
Home 2	**67.20**	63.24	32.14	**63.62**	**64.37**	44.74	2.04	**15.45**	13.78	**30.39**	35.89	**43.48**
Home 3	80.07	**86.88**	**95.32**	85.67	26.14	**40.88**	3.26	**8.09**	0.00	0.00	40.96	**44.30**
Home 4	49.77	**57.09**	63.92	**75.12**	69.04	**93.27**	30.50	**33.71**	**36.41**	7.39	49.93	**53.31**
Home 5	81.47	**92.13**	**86.41**	81.26	**45.05**	33.13	**21.30**	16.45	0.30	**1.35**	**46.91**	44.88
Home 6	35.17	**52.55**	**90.81**	61.8	**72.77**	45.74	22.54	**22.88**	**56.00**	16.89	**55.46**	39.97
Average	64.89	**71.06**	**74.77**	73.18	**48.24**	45.03	**20.59**	17.12	**19.61**	9.64	**45.62**	43.21

Additionally, we perform a statistical test, called Wilcoxon signed-rank test [Wilcoxon, 1945], to compare our method and the best reported result from the literature [Wu et al., 2009]. We choose such kind of statistical hypothesis test, based on the low number of available sample (below 10) and because we have access to the set of values for each instance, different from Caltech 101 and Caltech 256 where the literature reports just the mean and standard deviation. Hence, it was calculated the wilcoxon test to the whole VPC dataset, showing that through the high value of the computed p-values, the test gives no evidence against the null hypothesis (H_0 : SSC = VPC), showing no statistical difference in the means, see Appendix B.

Chapter 5

CONCLUSION

This thesis presented a novel methodology for object recognition, called SSC, that uses sparse coding dictionary learning with a spatial Euclidean coding phase. Furthermore, an encouraging result of this work is that it builds image representation work with online learning algorithms, which present some desired properties, like low memory consumption, and large amounts of data can be processed, which makes it suitable for data streaming.

Experimental evaluation was performed on the Caltech 101, Caltech 256, Corel 5000 and Corel 10000 datasets that were specifically designed for the evaluation of object recognition algorithms. The obtained results show that, to the best of our knowledge, our approach achieves accuracy beyond the best single feature method previously published on those databases. The method also outperformed, for the same bases, several methods that use multiple feature, and provide equivalent to or slightly lower results than other techniques. Finally, we verify our method generalization, applying the SSC to recognize scenes in the Indoor 67 scene dataset and VPC, displaying performance comparable to state-of-the-art approaches to this type of application.

Related to the system drawbacks, we can point out the high computational requirements of our approach, due to the feature extraction process, SIFT descriptors, and our dictionary learning module, since this problem is NP-HARD [Ophir et al., 2011]. Another constraint lies in the fact that if several objects are presented in an image, SSC will recognize only one class, probably of the least occluded object.

Future works will include exploration of sparse supervised dictionary learning methods, which could lead to better accuracy, and faster methods to dictionary learning. Other types of constraints and/or additional regularization will be investigated and other datasets, with data collected by robots, will be experimented.

We also intend to apply this method to semantic mapping applications, since the developed object recognition method can be responsible for extract semantic knowledge from images. Additionally, we aim to build a semantic mapping dataset, with intensity and depth information, from RGB-D sensors. Such kind of dataset for semantic mapping benchmark is not available, to the best of our knowledge, and based on the work of Lai et al. [2011], we believe that intensity and depth would significantly boost our object recognition technique.

Bibliography

Agin, G. (1972). *Representation and description of curved objects*. PhD thesis, Stanford University.

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.

Aharon, M. Elad, M. and Bruckstein, A. (2006). K-svd: design of dictionaries for sparse representation. In *IEEE Trans. Image Processing*.

Alpaydin, E. (2010). *Introduction to machine Learning*. MIT press, Massachusetts.

Bay, H., Tuytelaars, T., and Gool, L. V. (2006). Surf: Speeded up robust features. *European Conference on Computer Vision*, pages 404--417.

Belongie, S., Malik, J., and Puzicha, J. (2002). Shape matching and object recognition using shape contexts. *TPAMI*, 24(4):509--522.

Binford, T. (1971). Visual perception by computer. In *IEEE Conference on Systems and Control*.

Bischof, H. and Leonardis, A. (2000). Robust recognition using eigenimages. *Computer Vision and Image Understanding (CVIU)*, 78(1):99--118.

Boiman, O. (2008). In defense of nearest-neighbor based image classification. In *CVPR*.

Bolles, R. C. and Horaud, R. (1987). A three-dimensional part orientation system. *Three Dimensional Vision*, pages 399–450.

Bordes, A., Bottou, L., Gallinari, P., and Weston, J. (2007). Solving multiclass support vector machines with larank. In *ICML*, pages 89--96.

Bordes, A., Usunier, N., and Bottou, L. (2008). Sequence labelling svms trained in one pass. In *Machine Learning and Knowledge Discovery in Databases: ECML PKDD 2008*, pages 146–161.

Bosch, A., Zisserman, A., and Munoz, X. (2007). Representing shape with a spatial pyramid kernel. In *Proceedings of the ACM International Conference on Image and Video Retrieval*.

Boureau, Y., Bach, F., LeCun, Y., and Ponce, J. (2010a). Learning mid-level features for recognition. In *CVPR*.

Boureau, Y., Ponce, J., and Lecun, Y. (2010b). A theorical analysis of feature pooling in visual recognition. In *ICML*.

Coates, A. and Andrew, N. (2011). The importance of encoding versus training with sparse coding and vector quantization. In *ICML*.

Coates, A., Lee, H., and Andrew, N. (2011). An analysis of single-layer networks in unsupervised feature learning. In *AISTATS*.

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.

Fei-Fei, L. and Perona, P. (2005). A bayesian hierarchical model for learning natural scene categories. In *CVPR*.

Gao, S., hung Tsang, I. W., tien Chia, L., and Zhao, P. (2010). Local features are not lonely? laplacian sparse coding for image classification. In *CVPR*.

Gehler, P. and Nowozin, S. (2009a). On feature combination for multiclass object classification. In *ICCV*.

Gehler, P. V. and Nowozin, S. (2009b). On feature combination for multiclass object classification. In *ICCV*.

Grauman, K. and Darell, T. (2006). Unsupervised learning of categories from sets of partially matching image features. In *CVPR*.

Grimson, W. and Lozano-Prez, T. (1987). Localizing overlapping parts by searching the interpretation tree. *TPAMI*, 9(4):469 –482.

Guilleaume, H., Dubois, M., Frenoux, E., and Tarroux, P. (2011). Temporal bag-of-words a generative model for visual place recognition using temporal integration. *VISAPP*.

Jain, P., Kullis, B., and Grauman, K. (2008). Fast image search for learned metrics. In *CVPR*.

jia Li, L., Su, H., Xing, E. P., and Fei-fei, L. (2010). Object bank: A high-level image representation for scene classification and semantic feature sparsification. In *NIPS*.

Jiang, Z., Lin, Z., and Davis, L. S. (2011). Learning a discriminative dictionary for sparse coding via label consistent k-svd. In *CVPR*.

Kavukcuoglu, K., Ranzato, M., Fergus, R., and LeCun, Y. (2009). Learning invariant features through topographic filter maps. In *CVPR*.

Lai, K., Bo, L., Ren, X., and Fox, D. (2011). Sparse distance learning for object recognition combining rgb and depth information. In *ICRA*.

Lamdan, Y., Schwartz, J. T., and Wolfson, H. J. (1988). Object recognition by affine invariant matching. In *CVPR*.

Lazebnik, S., Schmid, C., and Ponce, J. (2006). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, pages 2169--2178.

Lee, H., Battle, A., Raina, R., and Ng, A. Y. (2006). Efficient sparse coding algorithms. In *NIPS*, pages 801--808. NIPS.

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*, pages 1712 –1719.

Liu, L., Wang, L., and Liu, X. (2011). In defense of soft-assignment coding. In *ICCV*.

Lowe, D. (1999). Object recognition from local scale-invariant features. In *ICCV*.

Lowe, D. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, pages 91--110.

Lowe, D. G. (1987). The viewpoint consistency constraint. *International Journal of Computer Vision (IJCV)*, 1(1):57--72.

Lu, Z. and Ip, H. H. (2009). Image categorization by learning with context and consistency. In *CVPR*.

Mairal, J. (2011). Sparse modeling software. http://www.di.ens.fr/willow/SPAMS/.

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., Sapiro, G., and Elad, M. (2008b). Learning multiscale sparse representations for image and video restoration. Technical Report 7.

Mori, G., Belongie, S., and Malik, J. (2005). Efficient shape matching using shape contexts. *TPAMI*, 27(11).

Mundy, J. L. and Heller, A. J. (1990). The evolution and testing of a model-based object recognition system. In *CVPR*.

Murase, H. and Nayar, S. (1995). Visual learning and recognition of 3-d objects from appearance. *IJCV*, 14:5--24.

Nakayama, H., Harada, T., and Kuniyoshi, Y. (2010). Global gaussian approach for scene categorization using information geometry. In *CVPR*, pages 2336–2343.

Ng, A. Y. (2004). Feature selection, l1 vs. l2 regularization, and rotational invariance. In *ICML*.

Oliveira, G. L., Nascimento, E., Vieira, A. W., and Campos, M. (2012). Sparse spatial coding: A novel approach for efficient and accurate object recognition. In *ICRA*.

Olshausen, B. A. and Field, D. J. (1997). Sparse coding with an overcomplete basis set: A strategy employed by v1. In *Vision Research*, volume 37, pages 3311--3325.

Olshausen, B. A. and Field, D. J. (2004). Sparse coding of sensory inputs. current opinion in neurobiology. volume 12, pages 481--487.

Ophir, B., Lustig, M., and Elad, M. (2011). Multi-scale dictionary learning using wavelets. *Selected Topics in Signal Processing, IEEE Journal of*, 5(5):1014 –1024.

Orabona, F., Jie, L., and Caputo, B. (2012). Multi kernel learning with online-batch optimization. In *Journal of Machine Learning Research*, pages 165–191.

Pentlan, A. (1986). Perceptual organization and the representation of natural form. *Artificial Intelligence*, 28(2):293--331.

Perronnin, F., Snchez, J., and Mensink, T. (2010). Improving the fisher kernel for large-scale image classification. In *ECCV*.

Platt, J. C. (1999). *Fast training of support vector machines using sequential minimal optimization*, pages 185--208. MIT Press, Cambridge, MA, USA.

Ponce, J. and Brady, J. (1987). Towards a surface primal sketch. *Three Dimensional machine Vision*, pages 195–240.

Pronobis, A. and Caputo, B. (2009). COLD: COsy Localization Database.

Quattoni, A. and Torralba, A. (2009). Recognizing indoor scenes. In *CVPR*.

Ranzato, M., Poultney, C., Chopra, S., and LeCun, Y. (2006). Efficient learning of sparse representations with an energy-based model. In *NIPS*.

Rigamonti, R., Brown, M., and Lepetit, V. (2011). Are sparse representation really relevant for image classification. In *CVPR*.

Rigoutsos, I. and Hummel, R. (1995). A bayesian approach to model matching with geometric hashing. *Computer Vision and Image Understanding (CVIU)*, 62:11--26.

Rothganger, F., Lazebnik, S., Schmid, C., and Ponce, J. (2005). Object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints. In *International Journal of Computer Vision*.

Saffari, A., Godec, M., Pock, T., Leistner, C., and Bischof, H. (2010). Online multi-class lpboost. In *CVPR*.

Saffari, A., Leistner, C., Santner, J., Godec, M., and Bischof, H. (2009). On-line random forests. In *3rd IEEE ICCV Workshop on On-line Learning for Computer Vision*.

Shechtman, E. and Irani, M. (2007a). Matching local self-similarities across images and videos. In *CVPR*.

Shechtman, E. and Irani, M. (2007b). Matching local self-similarities across images and videos. In *CVPR*.

Sohn, K., Jung, D. Y., Lee, H., and Hero, A. O. (2011). Efficient learning of sparse, distributed, convolutional feature representations for object recognition. In *ICCV*.

Todorovic, S. and Ahuja, N. (2008). Learning subcategory relevances for category recognition. In *CVPR*.

Tsochantaridis, I., Joachims, T., Hofmann, T., and Altun, Y. (2005). Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6:1453--1484.

Ullah, M., Pronobis, A., Caputo, B., Luo, J., Jensfelt, R., and Christensen, H. (2008). Towards robust place recognition for robot localization. In *ICRA*.

Ullman, S. and Basri, R. (1991). Recognition by linear combinations of models. *TPAMI*, 13(10).

Wang, J., Yang, J., Yu, K., Lv, F., Huang, T., and Gong, Y. (2010). Locality-constrained linear coding for image classification. In *CVPR*.

Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80--83.

Wright, J., Yang, A., Ganesh, A., Sastry, S., and Ma, Y. (2009). Robust face recognition via sparse representation. *TPAMI*, 31(2):210 –227.

Wu, J., Christensen, H., and Rehg, J. (2009). Visual place categorization: Problem, dataset, and algorithm. In *IROS*.

Wu, J. and Rehg, J. M. (2011). Centrist: A visual descriptor for scene categorization. *TPAMI*, 33(8):1489 –1501.

Yang, J., Li, Y., Tian, Y., Duan, L., and Gao, W. (2009a). Group-sensitive multiple kernel learning for object categorization. In *ICCV*, pages 436–443.

Yang, J., Wright, J., Huang, T., and Ma, Y. (2008). Image super-resolution as sparse representation of raw image patches. In *CVPR*.

Yang, J., Yu, K., Gong, Y., and Huang, T. (2009b). Linear spatial pyramid matching using sparse coding for image classification. In *CVPR*.

Yang, M.-H. (2011). Object recognition. http://faculty.ucmerced.edu/mhyang.

Yu, K., Lin, Y., and Lafferty, J. (2011). Learning image representation from the pixel level via hierarchical sparse coding. In *CVPR*.

Yu, K. and Zhang, T. Gong, Y. (2009). Nonlinear learning using local coordinate coding. In *NIPS*.

Zhang, H., Berg, A. C., Maire, M., and Malik, J. (2006). Svm-knn: Discriminative nearest neighbor classification for visual category recognition. In *CVPR*, pages 2126--2136.

Zhang, Q. and Li, B. (2010). Discriminative k-svd for dictionary learning in face recognition. In *CVPR*, pages 2691 –2698.

Zhou, S. and Chellappa, R.and Moghaddam, B. (2003). Adaptive visual tracking and recognition using particle filters. In *International Conference on Multimedia and Expo*.

Zhu, J., Li, L.-J., Fei-Fei, L., and Xing, E. P. (2010). Large margin learning of upstream scene understanding models. In *NIPS*.

Chapter 6

Attachments

Appendix A

Confidence Interval Values

A.1 Confidence Intervals Caltech 101

A.1.1 Caltech 101 single feature

Comparison Between SSC (tunned) and the SSC [Oliveira et al., 2012].

Table A.1: Confidence Intervals Caltech 101 single feature

N. of training samples	5	10	15	20	25	30
Confidence Intervals	[-3.77; -1.63]	[-3.117; -2.083]	[-2.77; -1.63]	[-2.33; -1.26]	[-1.89; -1.06]	[-2.0147; -1.18]

A.1.2 Caltech 101 multiple feature

Comparison Between SSC and the LP-Beta [Gehler and Nowozin, 2009b], see Table A.2.

Table A.2: Confidence Intervals Caltech 101 multiple feature

N. of training samples	5	10	15	20	25	30
Confidence Intervals	[1.4564; 3.36]	[1.58; 2.44]	[-0.1341; 1.53]	[-0.03; 0.83]	[-0.10; 0.70]	[-0.10; -0.8966]

A.2 Confidence Interval Caltech 256

Comparison Between SSC and the CRBM [Sohn et al., 2011], see Table A.3.

Table A.3: Confidence Intervals Caltech 256

N. of training samples	15	30	45	60
Confidence Intervals	[-0.23; 0.27]	[-0.02; 0.50]	[-0.0076; -0.4724]	[-1.3825; -0.6475]

A.3 Confidence Interval Corel Datasets

Comparison Between SSC and the ScSPM [Yang et al., 2009b], see Table A.4.

Table A.4: Confidence Intervals Caltech 256

Methods	Corel 5000	Corel 10000
Confidence Intervals	[-1.61; -0.59]	[0.57; 1.23]

A.4 Confidence Interval MIT-67 Indoor Datasets

Comparison Between SSC and the GG [Nakayama et al., 2010], see Table A.5.

Table A.5: Confidence Intervals MIT-67 Indoor

Confidence Interval	[0.21; 2.087]

Appendix B

VPC dataset P-values

Table B.1: VPC P-Values

	Bed		Bath		Kitchen		Living		Dining		P-Value
	VPC	SSC	VPC	SSC	VPC	SSC	VPC	SSC	VPC	SSC	
Home1	75.76	74.51	80.04	71.66	12.03	12.45	43.90	6.1	11.15	1.82	P-value=0.125
Home2	67.20	63.24	32.14	63.62	64.37	44.74	2.04	15.45	13.78	30.39	P-value=0.625
Home3	80.07	86.88	95.32	85.67	26.14	40.88	3.26	8.09	0.0	0.0	P-value=0.625
Home4	49.77	57.09	63.92	75.12	69.04	92.27	30.50	33.71	36.41	7.39	P-value=0.625
Home5	81.47	92.13	86.41	81.26	45.05	33.13	21.30	16.45	0.30	1.35	P-value=0.625
Home6	35.17	52.55	90.81	61.8	72.77	45.74	22.54	22.88	56.0	16.89	P-value=0.3125
P-Value	P-value=0.1563		P-value=1.00		P-value=0.8438		P-value=1.00		P-value=0.4185		P-value=0.4648