A bag of features approach for human attribute analysis on face images

Rafael Will Macedo de Araujo

Thesis submitted
to the
Institute of Mathematics and Statistics
of the
University of São Paulo
for the
obtention of the title
of
Doctor of Science

Program: Computer Science

Advisor: Prof. Dr. Roberto Hirata Junior

During the development of this work
the author received financial support from CAPES

São Paulo, September, 2019

A bag of features approach for human attribute analysis on face images

This version of the thesis contains the corrections and changes suggested by the Committee Members during the public defense of the original version of this work, which occurred on Sep 06th, 2019. A copy of the original version is available at the Institute of Mathematics and Statistics (IME) of the University of São Paulo (USP).

Committee Members:

- Prof. Dr. Roberto Hirata Junior (advisor) - IME-USP
- Prof. Dr. Paulo André Vechiatto de Miranda - IME-USP
- Prof. Dr. Aparecido Nilceu Marana - UNESP
- Prof. Dr. David Menotti Gomes - UFPR
- Prof. Dr. Alain Ndimby Heritsimba Rakotomamonjy - Université de Rouen

Acknowledgments

First and foremost, I would like to thank my advisor and friend, professor Dr. Roberto Hirata Jr, for his support and friendship throughout the last 5 years. Roberto has always been willing to guide and help me in anything I needed, has trusted in my work and pushed me forward. His dedication and stimulating example taught me everything about what a good researcher should be.

I am also especially thankful for the time spent at the LITIS lab (*Université de Rouen, France*) under the supervision of professor Dr. Alain Rakotomamonjy, who received me so well and gave me all the support needed for my stay in France. I express my deepest gratitude for the opportunity to work with him, and for all the knowledge he shared with me about dictionary learning and sparse coding.

To the members of the defense committee, professors Aparecido Nilceu Marana, David Menotti Gomes, Alain Rakotomamonjy and Paulo André Vechiatto de Miranda, for accepting our invitation and for their precious suggestions to improve this work.

I am grateful to all the professors and employees at IME-USP, which collaborated with my education and professional growth, and contributed directly or indirectly with the production of this work. Especially, I would like to thank secretaries Katia and Lucileide from the Computer Science Graduation program, for their quick answers to all my questions and support with many bureaucratic matters, and to the administrators and collaborators of the Vision Network, which computers were very important to run all the experiments presented here. I am also grateful to the Office of the Graduate Studies (PRPG-USP) for their support with our solicitation for a better period to do my thesis defense.

To the Brazilian Federal Agency for Support and Evaluation of Graduate Education (CAPES), for funding my PhD studies and also for making my studies abroad possible.

To my longtime friends Thiago Fonsêca, Valdir Falcão and Marcos Cleison, I thank you all for the words of motivation and advices. To all my friends and colleagues at IME-USP, especially Milson Monteiro, Robson Feitosa, Rafael Roque and Yuri David, for the pleasure of their company and many laughter moments, as well as very important discussions in our study groups. I also thank my Brazilian colleague Fabio Spanhol (also a PhD candidate at UFPR), whom I shared an office while in Rouen, for our endless discussions about machine learning topics and tips about the life in France. Finally, a special thanks to Mariana Mazzei for being my angel in the last days of this work, when it was mostly needed.

I am very grateful to my parents, Naum de Araujo and Wâner Will. Without their

support it would not be possible to move to São Paulo, nor I would have the stability and strength to fulfill my dreams. Since my childhood they showed me the importance of good education and they did everything possible to provide me with the best, even when I could not realize it.

Lastly, I would like to thank God for giving me good health and wisdom, which were necessary to complete this work. I also thank him for the opportunity to have all these wonderful people in my life.

My sincere gratitude to everyone!

Resumo

Pesquisadores de visão computacional são constantemente desafiados com perguntas motivadas por aplicações reais. Uma dessas questões é se um programa de computador poderia distinguir grupos de pessoas com base em sua ascendência geográfica, usando apenas imagens frontais de seus rostos. Os avanços nesta área de pesquisa nos últimos dez anos mostram que a resposta a essa pergunta é afirmativa. Vários artigos abordam esse problema aplicando métodos como Padrões Binários Locais (LBP), valores de pixels brutos, Análise de Componentes Principais ou Independentes (PCA/ICA), filtros de Gabor, Características Biologicamente Inspiradas (BIF) e, mais recentemente, Redes Neurais Convolucionais (CNN).

Neste trabalho propomos combinar o modelo "bag-of-words" visual com novas técnicas de aprendizagem por dicionário e uma nova abordagem de estrutura espacial para características da imagem. Um extenso conjunto de experimentos foi realizado usando dois dos maiores bancos de dados de imagens faciais disponíveis (MORPH-II e FERET), alcançando resultados muito competitivos para reconhecimento de gênero e etnia, ao passo que utiliza um conjunto consideravelmente pequeno de imagens para treinamento.

Palavras-chave: processamento de imagens de faces; classificação de gênero e etnia; modelo bag-of-words visual; aprendizagem por dicionário.

Abstract

Computer Vision researchers are constantly challenged with questions that are motivated by real applications. One of these questions is whether a computer program could distinguish groups of people based on their geographical ancestry, using only frontal images of their faces. The advances in this research area in the last ten years show that the answer to that question is affirmative. Several papers address this problem by applying methods such as Local Binary Patterns (LBP), raw pixel values, Principal or Independent Component Analysis (PCA/ICA), Gabor filters, Biologically Inspired Features (BIF), and more recently, Convolution Neural Networks (CNN).

In this work we propose to combine the Bag-of-Visual-Words model with new dictionary learning techniques and a new spatial structure approach for image features. An extensive set of experiments has been performed using two of the largest face image databases available (MORPH-II and FERET), reaching very competitive results for gender and ethnicity recognition, while using a considerable small set of images for training.

Keywords: face image processing; gender and ethnicity classification; bag-of-visual-words model; dictionary learning.

Contents

List of Abbreviations

BOSSA	Bag of Statistical Sampling Analysis
BoW	Bag-of-Words
BoVW	Bag-of-Visual-Words
CNN	Convolutional Neural Network
CoI	Centers of Incidence
DL	Dictionary Learning
ICA	Independent Component Analysis
K-SVD	K-Singular Value Decomposition
LC-KSVD	Label Consistent K-Singular Value Decomposition
PCA	Principal Component Analysis
PIWAH	Pair of Identical visual Words Angle Histogram
RBF	Radial Basis Function
RDL	Robust Dictionary Learning
SIFT	Scale-Invariant Feature Transform
SPM	Spatial Pyramid Match
SURF	Speeded Up Robust Features
SVM	Support Vector Machine

List of Figures

List of Tables

Chapter 1

Introduction

Adult human beings can usually distinguish certain groups of people based on their geographical ancestry. This poses a fair question to the Computer Vision researchers: could a computer program do the same based on frontal images of human faces? Many papers name this problem as "race" or "ethnicity" discrimination/classification. None of these words correctly applies to the problem and they are also subject of controversial discussions. In this work, we are going to follow Fu *et al.* (2014) and adopt "ethnicity classification" to name this problem.

Gender is another possible subject of study in the realm of classification of face images. Gender, ethnicity and face recognition have a widely range of applications: from CBIR systems and automatic image annotation, to social networks and social privacy, targeted advertising, law enforcement applications, etc. In a nutshell, a solution to these problems can be useful to mitigate the complexity of several applications related to human identification.

The advances in this research topic in the last ten years show that it is possible to classify gender and ethnicity based solely on face images. Although there are several papers that approach this problem (recent surveys can be found in Fu *et al.* (2014), Ng *et al.* (2015) and Ng *et al.* (2012)), one of the key challenges about developing a method for ethnicity and gender categorization is the scarcity of databases with built-in labels that by themselves are very difficult to assign, specially for ethnicity. Furthermore, most of the existing databases offer only a few thousand images, many from the same subject in distinct poses or acquisition dates.

Despite the problem about creating and labeling a new database, which is arduous and challenging by itself, the majority of the papers (about 40) reviewed in the surveys use databases with less than 4,000 images, usually with less unique subjects than that. For this reason, although many of them report great accuracy ($\geq 90\%$), their results could be biased and the proposed methods could underperform drastically if tested in bigger and more diverse databases.

1.1 Objectives

The main motivation of this work is to improve the Bag-of-Visual-Words (BoVW) model for the problem of ethnicity and gender classification from faces, by proposing new ideas and methods for its clustering and pooling phases. We used the MORPH-II dataset, one of the largest available with labels for both ethnicity and gender (Ricanek Jr and Tesafaye, 2006). We designed and experimented with domain adaptation using two other well known databases in the literature for this problem: FERET (Phillips *et al.*, 2000) and AR (Martinez , 1998).

We are specially interested in improving the quality of the generated codewords (the "visual vocabulary"), as well as to suggest new ideas to improve the quality of existing pooling methods. Finally, we are also interested in some properties of the kind of datasets we are dealing with (faces), since they are stable in the sense that most features (such as mouth, nose, eyes, ears, etc) will share similar relative spatial positions along all images in the set.

Furthermore, we hope the methods proposed here will be useful not only for gender or ethnicity classification, but also to a wide range of problems with similar needs and/or properties. Our Centers of Incidence approach (see Chapter 4) can be used to improve accuracy within existing pooling strategies, and our pooling formalism (also in Chapter 4) should work fine in any dataset with the property of quasi-invariability of the spatial position of features (details in Section 4.4). Similarly, our Robust Dictionary Learning scheme (see Chapter 5) can be applied in many applications which require clustering, sparse coding or even outlier detection techniques.

1.1.1 Contributions

The major contributions achieved by this work are the following:

- A new way to divide the spatial subregions when using approaches like the spatial pyramids (Lazebnik *et al.*, 2006). We introduce the concept of "Centers of Incidence" (CoIs), based on the relative spatial distribution of features in the training set. This method can directly benefit the existing pooling methods.

- A side effect of proposing the CoIs approach led to the development of our own pooling method, which takes advantage in the property of the stability of the relative spatial position of features in face images.

- A new framework for Robust Dictionary Learning (RDL), based on the composition of two concave functions to diminish the effects of outliers in the training set.

- A heuristic for our RDL method algorithm initialization, which employs some under-complete dictionaries and helps it to better detect outliers, and consequently delivers a better set of codewords.

- Experiments with ethnicity and gender classification using the proposed approaches with a large ($> 40,000$ images) test set, while achieving competitive results even when using a relatively small training set.

1.2 Publications

During the development of this work, the following papers were published and are currently planned:

- The paper Araujo *et al.* (2018) describes our Robust Dictionary Learning framework (see Section 5.4), which takes advantage of a composition of two concave functions to generate robust dictionaries while suppressing the interference of outliers in the training set. It also proposes a heuristic initialization which can further increase the identification of outliers through the use of undercomplete dictionaries. This study was presented as a lecture at the *2018 IEEE International Conference on Acoustics, Speech and Signal Processing* (ICASSP), from 15 to 20 of April 2018, in Calgary, Canada.

- A paper with a more detailed study of our RDL approach along with its mini-batch version (see Section 5.4.3) and new experiments is being prepared to be published.

- A paper on the new pooling formalism is presented in Chapter 4. As in Chapter 6, where we plan to show how existing pooling methods can benefit from our idea of Centers of Incidence (see Section 4.2), as well as the superiority of our new pooling method presented here in Section 4.4, is being prepared to be published.

1.3 Organization of this work

This work is organized as follows: Chapter 2 introduces the main researches in the area of ethnicity and gender classification. Chapter 3 reviews the bag-of-words architecture along with other theoretical foundations. Chapter 4 introduces new ideas on how to subdivide image features into subregions and presents a new pooling strategy that takes the spatial distribution of features into account. Chapter 5 reviews some important dictionary learning techniques and proposes a new robust dictionary learning algorithm to attenuate the importance of outliers in the input data. Chapter 6 presents experiments using the proposed methods. Finally, Chapter 7 contains our final considerations and the conclusion of this work.

Chapter 2

Literature Review

This chapter makes a brief introduction to the main works in the area of ethnicity and gender classification from human faces.

2.1 Related work on gender classification

The problem of gender classification has been recently surveyed by Ng *et al.* (2015). The authors reviewed about 40 papers and report that most of them use databases of less than 4,000 images: some works are based on the use of smaller databases, such as FERET (Phillips *et al.*, 2000), while others select a fraction of larger databases to approximate the number of females and males, due to the unbalance between classes. Concerning methods, most of them uses Local Binary Patterns (LBP), pixel values, Principal Component Analysis (PCA) or Independent Component Analysis (ICA), Gabor filter, all associated with Support Vector Machine (SVM) classifiers.

From the papers reported, the study proposed by Ramón-Balmaseda *et al.* (2012) do experiments with a very large database known as MORPH-II (Ricanek Jr and Tesafaye , 2006) database. However, they only consider a subset of 8,488 females and 9,326 males and not the entire dataset. They use Local Binary Patterns (LBP) (Ojala *et al.*, 1996, 2002), which is a well known texture descriptor, computed for many subregions that are combined later, and then used to feed a Support Vector Machine (SVM) classifier. The authors report they achieved accuracies ranging from 94% to 97%, but they use around 14,000 images for their training set, leaving only 3,560 images for the test set, a small and disproportional amount compared to the set used for training the classifier. They also perform two cross-database experiments (i.e., given databases A and B, one trains with database A and tests with database B) using the Labeled Faces in the Wild (LFW) database (Huang *et al.*, 2007) and the Image of Groups databases (Gallagher and Chen, 2009) as the test set, achieving accuracies ranging from 50% to 76% in both databases.

Another study that uses MORPH-II is the one proposed by Chu *et al.* (2010), using subspace learning techniques. The number of images used for training and for testing is not

exactly clear, but the authors report they use the same number of male and females subjects for training and test. They achieve an accuracy near 88% when using 400 training subjects. None the methods proposed by Ramón-Balmaseda *et al.* (2012) and by Chu *et al.* (2010) were verified for a potential classification of ethnicity.

More recently, Borgi *et al.* (2014) reported accuracies around 92% for the AR database (Martinez , 1998), but he uses half the images (50) for training, leaving only the other 50 images for test. They also use the FEI database (Thomaz and Giraldi, 2010) for gender experiments, reaching about 94% accuracy, but to achieve this they had to use 1,800 images for training and only 1,000 for test. The DeepGender method proposed by Juefei-Xu *et al.* (2016) uses around 89,000 images from 5 different datasets for training, achieving accuracies around 79% to 98% using the AR dataset for test, depending on the subset.

The work of Duan *et al.* (2018) proposes age and gender classification on the MORPH-II dataset using more advanced deep learning techniques. They report an accuracy slightly better than 87% for gender classification.

2.2 Related work on ethnicity classification

The problem of ethnicity classification from face images has also been recently surveyed by Fu *et al.* (2014). The authors reviewed about 60 papers on some of the different recognition methods and databases used to train and test the classifiers. This survey states that most of the about 20 databases reviewed are not specific to ethnicity and gender classification but they can be used to this task because they are already labeled. One important information about the databases reviewed is that most of them have less than 10,000 images. Besides that, there are several images per subject so the amount of distinct persons is even less than that.

Fu *et al.* (2014) also review several algorithms published to classify gender and ethnicity. Local Binary Patterns, or Kernel Class-dependent Feature Analysis (KCFA), or shape, or skin color and or Haar Wavelets, combined with Support Vector Machines are the most commonly used methods to classify ethnicity. Forty two papers have been reviewed in the survey and only five of them report less than 90% of accuracy. The best accuracy reported is 99.5% for a method that uses a combination of Harr Wavelets, KNN, Kernel KNN and Multidimensional Scaling (MDS), although Fu *et al.* (2014) states that these accuracies mean the best results for all possible ethnic groups.

Only one paper surveyed (Guo and Mu, 2010) uses the MORPH-II database, and it applies techniques such as Biologically Inspired Features (BIF) combined with Manifold Learning, Principal Component Analysis (PCA), Orthogonal Locality Preserving Projections and SVM to classify ethnicity. However, they only consider ethnic groups with the same gender, because their objective is to evaluate the influence of gender (and also age) for ethnicity classification. Moreover, they use a smaller subset of the database, with 5,140

females and 15,920 males. In their experimental design, each gender is divided into two groups used as training and test sets, meaning they use a considerable amount of training images. The results achieve accuracies ranging from 97% to 99%.

The work of Wang *et al.* (2016b) also made experiments using the MORPH-II dataset and compared their results with Guo and Mu (2010). Thus, they randomly chose 10,530 images from Africans and another 10,530 from Caucasians, and used them in a ten fold scheme, similarly to what Guo and Mu (2010) did. They reported an overall accuracy above 99%, using a training set with around 19,000 images.

More recently, Wang *et al.* (2017) proposed a Deep Multi-Task Learning (DMTL) network to perform age, gender and race classification at the same time. The model is pre-trained with the ImageNet 2012 dataset (Krizhevsky *et al.*, 2012), and they use the entire MORPH-II dataset in a five fold scheme to fine-tune the initial model and then classification. They reported around 96% and 98% for ethnicity and gender, respectively.

Chapter 3

Bag-of-Visual-Words

This chapter presents an introduction to the bag-of-words formalism and its applicability in the area of computer vision, along with other correlated methods.

3.1 The bag-of-words model for text classification

Inspired by the work by Harris (1954) on text analysis using the frequency of the words' occurrences in a text, Bag-of-Words (BoW) has been proposed as a text ranking and classification method. The mathematical definition of a "Bag" is a set that admits duplicated elements. Therefore, the histogram of frequencies of the elements presented in a bag can be used as a vector of features to represent an instance of the bag.

The BoW architecture was also exploited by computer vision researchers with good successes. First, low-level features, such as those detailed in Section 3.2, are extracted from an image. Then, using unsupervised learning algorithms, these low-level features are clustered, generating a codebook (an analogy to a word dictionary). Finally, a pooling operation takes place to associate each low-level feature to one or more clusters. This final operation generates a histogram of frequencies of codewords, similar to the one generated by the traditional BoW model, which is a mid-level representation of the image. In the following sections we detail and explore this architecture.

3.2 Feature descriptors

Local image descriptors plays an important role in several computer vision tasks nowadays. Among the several types of local descriptors, SIFT and SURF are the most used and cited ones. SIFT is a method introduced by Lowe (Lowe, 1999) for identifying image features that are invariant to translation, rotation and scale. The method was adopted by Csurka (Csurka *et al.*, 2004) for its good stability against perturbations (noise), robustness to moderate perspective transformations and illumination variations, leading to a richer and more discriminative representation. The method also enables the use of Euclidean metric

over the feature space, which is simpler and faster to compute in contrast to Mahalanobis distance (Mahalanobis, 1936), for example.

The main idea of the method is to obtain image descriptors from the difference between successive Gaussian-blurred images in distinct scales. These descriptors are computed from keypoints, which are obtained based on the analysis of the orientations of the gradients at each point of the image. This analysis stage determines potential points and eliminate unstable ones. The surviving points will form the set of SIFT keypoints from an image.

The other method that is largely used to obtain image descriptors is SURF (Bay *et al.*, 2006). The method is based on SIFT but it is designed to be faster. In its standard version, SURF detects points of interest by an approximation of the Hessian matrix and employs a squared-shaped filter to approximate the Laplacian of Gaussian (LoG). This allows the use of a summed area table (also known as integral image) for filtering, which is very efficient and can be done easily in parallel. SURF descriptors are based on the sum of the Haar Wavelets response around a point of interest.

Local descriptors are computed over keypoints, normally from random points or over a evenly sampled grid of them. Both methods give good results but points evenly sampled gives better accuracy (Fei-Fei and Perona, 2005b).

3.3 A study about visual words

It is possible to apply the BoW model to digital images, through some adaptations. Firstly, there is no textual vocabulary that could be utilized. Thus, a local feature descriptor such as SIFT or SURF is employed to extract a set of points of interest which are capable to discriminate distinct aspects in an image.

Figure 3.1 shows faces from 4 subjects: a man and a woman from Caucasian origin (Figures 3.1a and 3.1b), a man and a woman from Hispanic origin (Figures 3.1c and 3.1d). Each face contains 3 fixed points, described as the following:

- Points $\{P_{1a}, P_{1b}, P_{1c}, P_{1d}\}$: central region of the right [1] eye.
- Points $\{P_{2a}, P_{2b}, P_{2c}, P_{2d}\}$: right lower corner [1] of the nose (near the nostril).
- Points $\{P_{3a}, P_{3b}, P_{3c}, P_{3d}\}$: central region from the mouth.

These points are selected not only because they represent important features of human faces (such as mouth, nose, eye), but also because they are in regions with drastic gradient changes. Thus, they are ideal for categorization with feature descriptors like the ones described in Section 3.2.

After fixing the points in Figure 3.1, one can compute their respective SIFT descriptors. Each descriptor is a 128-dimensional vector, and the respective values for the points above

[1] With respect to the subject.

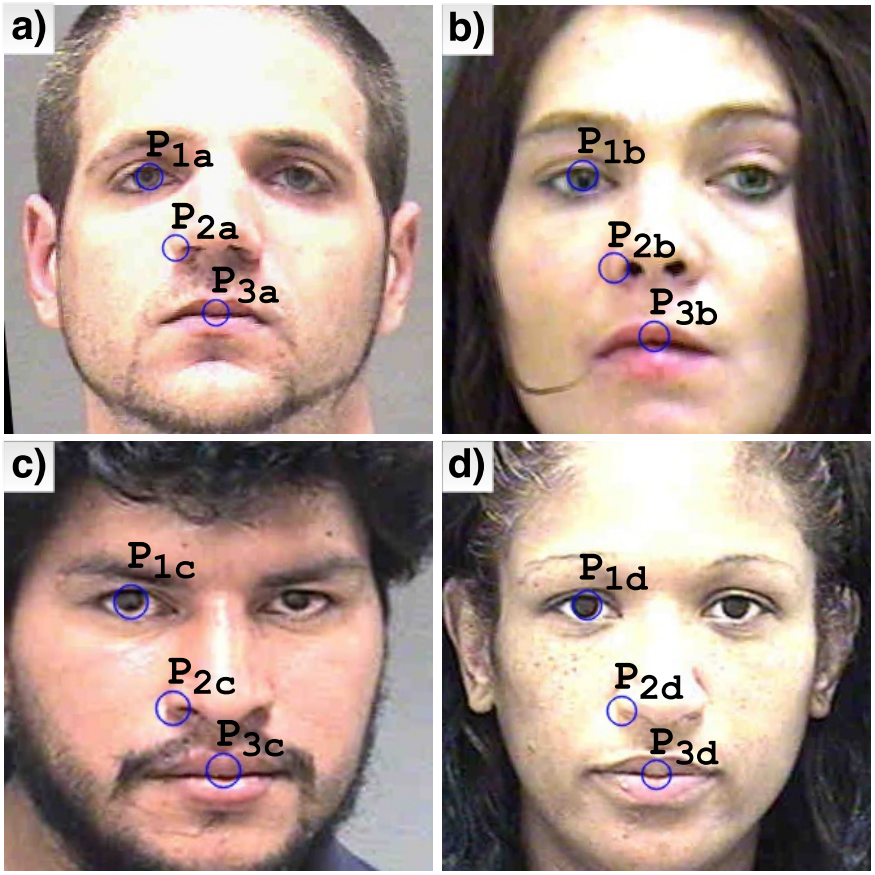


Figure 3.1: *Examples of visual words in face images extracted from MORPH-II database.*

are presented in Tables A.1 and A.2 (see Appendix A). Table 3.1 shows all possible distances between each point represented by its SIFT descriptors (Fig. 3.1). The 4-nearest neighbors from each point are highlighted.

	P_{1a}	P_{2a}	P_{3a}	P_{1b}	P_{2b}	P_{3b}	P_{1c}	P_{2c}	P_{3c}	P_{1d}	P_{2d}	P_{3d}
P_{1a}	0.0	532.4	364.0	196.9	481.3	399.2	329.0	519.8	351.1	227.8	517.3	358.1
P_{2a}	532.4	0.0	466.2	515.2	186.6	456.8	479.1	273.5	468.2	512.9	267.9	473.7
P_{3a}	364.0	466.2	0.0	390.4	439.3	190.0	368.0	400.1	241.8	377.3	436.9	236.2
P_{1b}	196.9	515.2	390.4	0.0	478.0	408.2	311.6	507.8	374.8	206.5	497.0	386.4
P_{2b}	481.3	186.6	439.3	478.0	0.0	428.0	436.5	275.9	443.5	455.9	285.0	444.0
P_{3b}	399.2	456.8	190.0	408.2	428.0	0.0	373.3	404.4	302.2	398.1	446.1	289.2
P_{1c}	329.0	479.1	368.0	311.6	436.5	373.3	0.0	474.9	358.1	289.9	492.4	356.9
P_{2c}	519.8	273.5	400.1	507.8	275.9	404.4	474.9	0.0	424.5	492.3	196.0	410.7
P_{3c}	351.1	468.2	241.8	374.8	443.5	302.2	358.1	424.5	0.0	348.4	438.2	132.9
P_{1d}	227.8	512.9	377.3	206.5	455.9	398.1	289.9	492.3	348.4	0.0	491.1	345.6
P_{2d}	517.3	267.9	436.9	497.0	285.0	446.1	492.4	196.0	438.2	491.1	0.0	425.3
P_{3d}	358.1	473.7	236.2	386.4	444.0	289.2	356.9	410.7	132.9	345.6	425.3	0.0

Table 3.1: *Distance matrix between the SIFT descriptors from the points in Figure 3.1*

From the data in Table 3.1, one can see that given any point, its nearest points are the ones from other images in similar regions. One possible explanation is that these regions share similar gradient directions. Furthermore, also examining Table 3.1, people from the

Point	Nearest centroid
P_{1a}	c_1
P_{2a}	c_2
P_{3a}	c_3
P_{1b}	c_1
P_{2b}	c_2
P_{3b}	c_3
P_{1c}	c_1
P_{2c}	c_2
P_{3c}	c_3
P_{1d}	c_1
P_{2d}	c_2
P_{3d}	c_3

Table 3.2: *Nearest centroids to the SIFT descriptors from the points in Figure 3.1. Centroid coordinates are available in Table A.3.*

same ethnic group are closer, compared to people from different ones.

If a vector quantization algorithm such as K-Means is applied to SIFT descriptors, it is possible to group the points from Figure 3.1 into clusters of similar elements. Table 3.2 shows the centroids assigned[2] to each point from Figure 3.1, considering 3 clusters.

Note that each cluster is correctly assigned to the same specific region from each of the considered faces: centroid c_1 represents the right eye, c_2 the right lower corner from the nose, and c_3 the central region from the mouth. A visual word represents a group of similar descriptors. In the example of Figure 3.1 and Table 3.2, each one of the centroids is a visual word. Consequently, the vocabulary size is adjustable, depending on the number of clusters chosen for the K-Means algorithm.

3.4 The Bag-of-Visual-Words model

Using the same ideas from the BoW model, the Bag-of-Visual-Words (BoVW) method has been proposed by Csurka *et al.* (2004) and performs considerably well for scene and object classification. This model has no predefined vocabulary as in the case of the text categorization problem. Instead, multidimensional features are extracted from the images in a training set and later grouped into "words". The rational to do that is because many of the feature vectors share some similarities with one another and then a vocabulary can be generated using unsupervised learning algorithms from the set of feature vectors.

Formally, consider a set of images $\mathcal{I} = \{\mathbf{I}_1, \ldots, \mathbf{I}_N\}$ as a training set, $\mathcal{U} = \{\mathbf{U}_1, \ldots, \mathbf{U}_N\}$ a set of descriptors, where $\mathbf{U}_i \in \mathbb{R}^{d \times M}$ corresponds to the set of descriptors extracted from the image $\mathbf{I}_i$, so that $\mathbf{U}_i = \{\mathbf{u}_{i_1}, \ldots, \mathbf{u}_{i_M}\}$, with $\mathbf{u}_{i_j} \in \mathbb{R}^d$, i.e., d is the dimension of the

[2]The closest centroid from each point.

descriptor vector and M the number of descriptors (or features) per image[3].

After the extraction of the descriptors from the images in the training set, a vector quantization method partitions all the elements $\mathbf{u}_{i_j} \in \mathcal{U}$ into K clusters, along with their respective centroids: $\mathcal{C} = \{\mathbf{c}_1, \ldots, \mathbf{c}_K\}$, with $\mathbf{c}_i \in \mathbb{R}^d$. The centroids (or codewords) are representatives of the clusters and they form a visual vocabulary of size K, also known as the codebook or dictionary. The standard K-Means algorithm is widely used for this task, since it is one of the simplest square-error partitioning methods (Csurka *et al.*, 2004). However, other algorithms can be used for this task, such as the K-SVD (Aharon *et al.*, 2006) or LC-KSVD (Jiang *et al.*, 2011), which are dictionary learning methods for sparse representations based on multiple singular value decomposition.

Let $\mathcal{H} = \{\mathbf{h}_1, \ldots, \mathbf{h}_N\}$ be a set of histograms, where $\mathbf{h}_i$ corresponds to a single histogram that encodes $\mathbf{U}_i$. Each $\mathbf{h}_i$ has a fixed length equal to K, the vocabulary size. Let $\mathbf{h}_{i_m}$ be the m-bin of histogram $\mathbf{h}_i$, with $1 \leq m \leq K$, and let f be a function that projects a descriptor $\mathbf{u}_{i_j}$ to a visual word:

$$f(\mathbf{u}_{i_j}) = \begin{cases} 1 & \text{if } m = \underset{k \in \{1,\ldots,K\}}{\arg\min} ||\mathbf{u}_{i_j} - \mathbf{c}_k||_2^2, \, \forall \mathbf{u}_{i_j} \in \mathbf{U}_i \\ 0 & \text{otherwise} \end{cases} \tag{3.1}$$

then, for each bin m of $\mathbf{h}_i$, a pooling operation is performed according to:

$$\mathbf{h}_{i_m} = \sum_{j=1}^{M} f(\mathbf{u}_{i_j}) \tag{3.2}$$

Finally, each histogram $\mathbf{h}_i$ along with its respective label (provided by the dataset) is given as input to a classifier (SVM, Naïve Bayes, etc) for training.

Once the classifier has been trained, given a new image $\mathbf{I}' \notin \mathcal{I}$, the descriptors are extracted from $\mathbf{I}'$ and encoded to a histogram $\mathbf{h}' \notin \mathcal{H}$, in the same way as performed in the training phase. Finally, $\mathbf{h}'$ is given to the classifier which returns its predicted label.

3.5 Spatial structure

BoVW is intrinsically a collection of orderless words therefore it does not take the spatial structure of a face into consideration. When dealing with face categorization problem, all the images belong to the same category, thus a single histogram of orderless local features from the whole face would not have sufficient between-class variations (Lazebnik *et al.*, 2006). One way to improve the method and use the structure information is to divide the image into regions, treat each region as a new image and compute the occurrences of words for each one of them, exemplified in Figure 3.3.

The algorithm is straightforward: after partitioning the image, treat each region as a new

[3] M is fixed for all images since we employ a dense sampling approach.

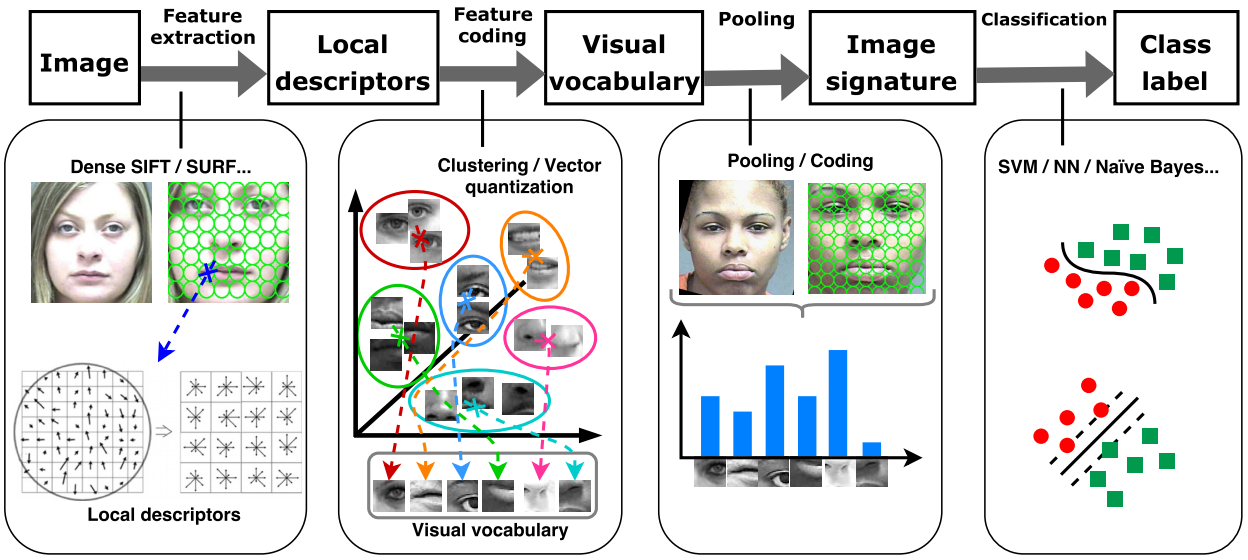


Figure 3.2: *The BoVW pipeline: features are extracted from human faces in order to create a visual vocabulary, which is used to train a classifier.*

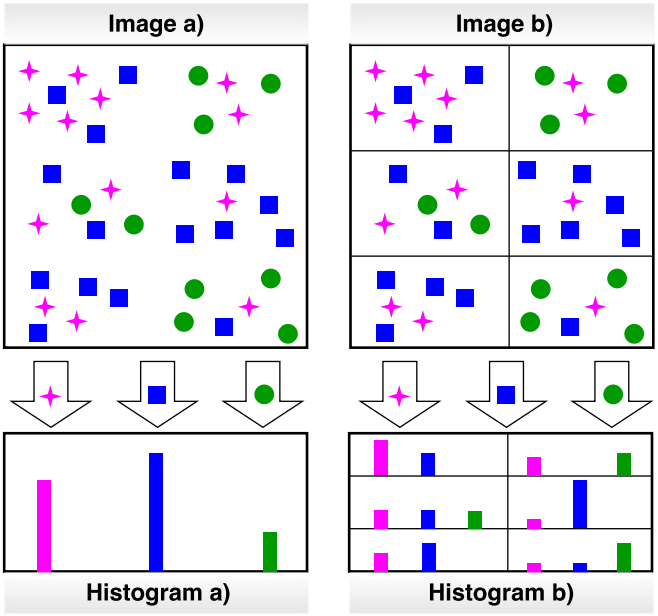


Figure 3.3: *Images a) and b) represent the same image (with equal features), while histograms a) and b) represent their feature counts, respectively. Note that Histogram b) is more discriminative than a).*

image and compute the occurrences of words for each one of them. The final histogram is achieved by concatenating every individual region histogram in the same order (e.g., top-bottom and left-right) for all images in a set. The size of that histogram depends directly on the number of regions and it is equal to $K \cdot p \cdot q$ in the standard BoVW model, where K is the size of the visual vocabulary, p and q are the number of horizontal and vertical subdivisions, respectively. Finally, equal-sized keypoints are sampled over a non-overlapping dense grid (Figure 6.5f). In addition, a new boundary is defined in the middle of the x-axis.

An extension of this idea was proposed by Grauman and Darrell (2005) and Lazebnik *et al.*

(2006). They came up with the concept of spatial pyramids, where the spatial positions of the features in an image are subdivided in various levels of resolution. Histograms in distinct levels are associated to a weight, which is inversely proportional to the cell width. This way, the more finer the resolution is, the higher is the weight associated with it.

3.6 Pooling strategies

More recently, extensions of the BoW formalism for images were proposed. In this work, we explore in more detail the ideas behind three of those novel models: PIWAH (Pair of Identical Words Angle Histogram) (Khan *et al.*, 2012), BOSSA (Bag of Statistical Sampling Analysis) (Avila *et al.*, 2011) and its improved version BossaNova (Avila *et al.*, 2013). Their main idea lies in improving the links between the codewords and local descriptors in the resulting histogram, by focusing on the pooling step.

3.6.1 PIWAH scheme

This technique aims to characterize the relative spatial distribution of the patches associated with each visual word $\mathbf{c}_k$. It assumes that for a given object category and visual word the distribution of angles between the descriptors is stable.

The construction of the proposed histogram is based on Equation 3.3. First, from the set $\mathscr{U}_{\mathbf{c}_k}$ of the descriptors assigned to the codeword $\mathbf{c}_k$, consider all the pairs of that set and build the set PIW_k composed by the corresponding position pair:

$$\text{PIW}_k = \{(P_a, P_b) | (\mathbf{u}_a, \mathbf{u}_b) \in \mathscr{U}^2_{\mathbf{c}_k}, \mathbf{u}_a \neq \mathbf{u}_b\} \tag{3.3}$$

where P_a and P_b are the spatial positions in the image from which descriptors $\mathbf{u}_a$ and $\mathbf{u}_b$ have been extracted. Thus, the cardinality of PIW_k is $\binom{\beta}{2}$, i.e. the number of all possible pairs between two distinct elements among β elements.

Next, for each pair of points in the set PIW_k, the angle θ between them and the horizontal axis is computed using the law of cosines:

$$\theta = \begin{cases} \arccos\left(\frac{\overrightarrow{P_aP_b} \cdot \vec{i}}{||\overrightarrow{P_aP_b}||}\right) & \text{if } \overrightarrow{P_aP_b} \cdot \vec{j} > 0 \\ \pi - \arccos\left(\frac{\overrightarrow{P_aP_b} \cdot \vec{i}}{||\overrightarrow{P_aP_b}||}\right) & \text{otherwise} \end{cases} \tag{3.4}$$

with $\overrightarrow{P_aP_b}$ being the vector formed by the points P_a and P_b, as well as the orthogonal unit vectors $\vec{i}$ and $\vec{j}$ which define the image plane. Finally, the histogram of all θ angles is computed, with the optimal number of bins B being chosen empirically. The authors called this the PIW angle histogram for the word $\mathbf{c}_k$, denoting it as PIWAH_k. Figure 3.4 presents three examples of PIWAH_k histogram distributions.

A global representation of an image is achieved by combining all the individual PIWAH_k histograms, according to Equation 3.5. Each sub histogram PIWAH_k is normalized to the

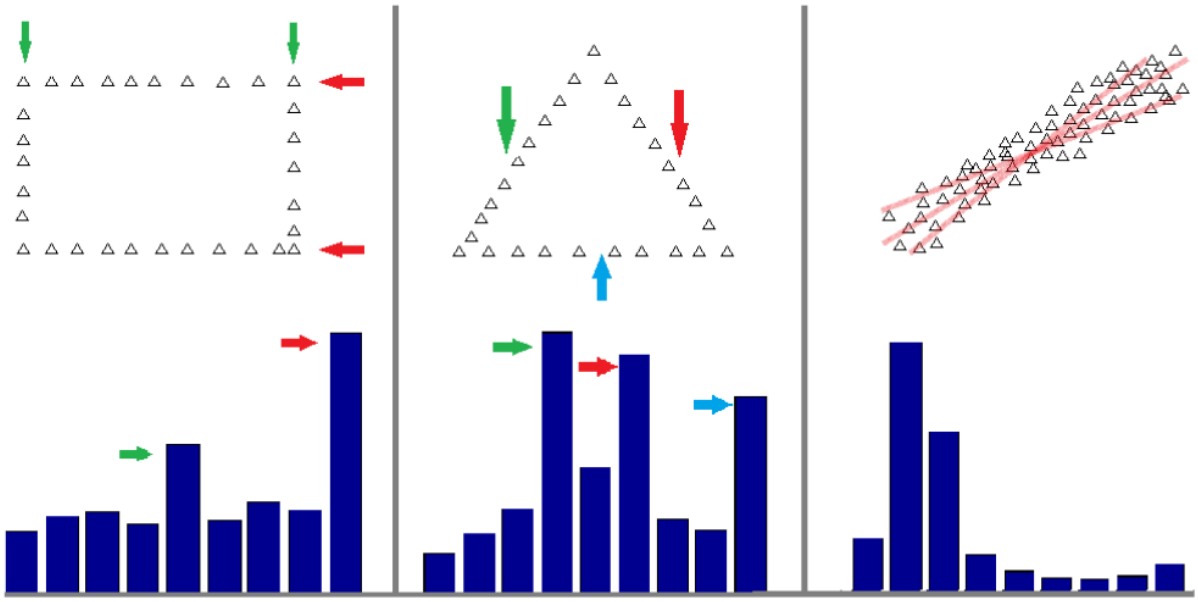


Figure 3.4: *Examples of toy distributions and their respective 9-bin PIW histograms (Khan et al., 2012).*

number of descriptors β assigned to $\mathbf{c}_k$.

$$\text{PIWAH} = (\alpha_1 \text{PIWAH}_1, \ldots, \alpha_K \text{PIWAH}_K) \tag{3.5}$$

with $\alpha_k = \frac{\beta}{||\text{PIWAH}_k||_1}$.

3.6.2 BOSSA scheme

Let $\alpha_{i,j}$ be the Euclidean distance from a centroid $\mathbf{c}_i$ to a descriptor $\mathbf{u}_j$, and let $\bar{\mathbf{c}}_i$ and σ_i be the mean and standard deviation of the distances from the descriptors of cluster to its centroid $\mathbf{c}_i$. The BOSSA histogram is computed by estimating the probability density function of each cluster, according to Equation 3.6:

$$\mathbf{z}_{i,b} = \text{card}\left(\mathbf{u}_j | \alpha_{i,j} \in \alpha_i^{max} . \left[\frac{b}{B}; \frac{b+1}{B}\right]\right) \tag{3.6}$$

where B denotes the number of bins of each histogram $\mathbf{z}_i$, $\alpha_i^{max} = \bar{\mathbf{c}}_i + (\lambda_{\max} \cdot \sigma_i)$ is the maximum distance in the $\mathbb{R}^d$ feature space to which $\mathbf{z}_i$ is computed. Essentially, B and $\lambda_{\max}$ are the two parameters of BOSSA. Figure 3.5 shows an illustration of the α_i^{max} parameter and histogram $\mathbf{z}_i$.

After computing a local histogram $\mathbf{z}_i$ for all centroids $\mathbf{c}_i$, they are normalized according to $\mathbf{z}_i = \mathbf{z}_i / ||\mathbf{z}_i||_1$ and then concatenated to build the final image representation $\mathbf{h}$. Additionally, a histogram $\mathbf{t} = \{t_1, \ldots, t_K\}$ counting the occurrence of descriptors in each cluster is incorporated to $\mathbf{z}$. The final BOSSA image representation is defined as:

$$\mathbf{h} = [[\mathbf{z}_{i,b}], t_i]^T, \text{ where } 1 \leq i \leq K \text{ and } 1 \leq b \leq B \tag{3.7}$$

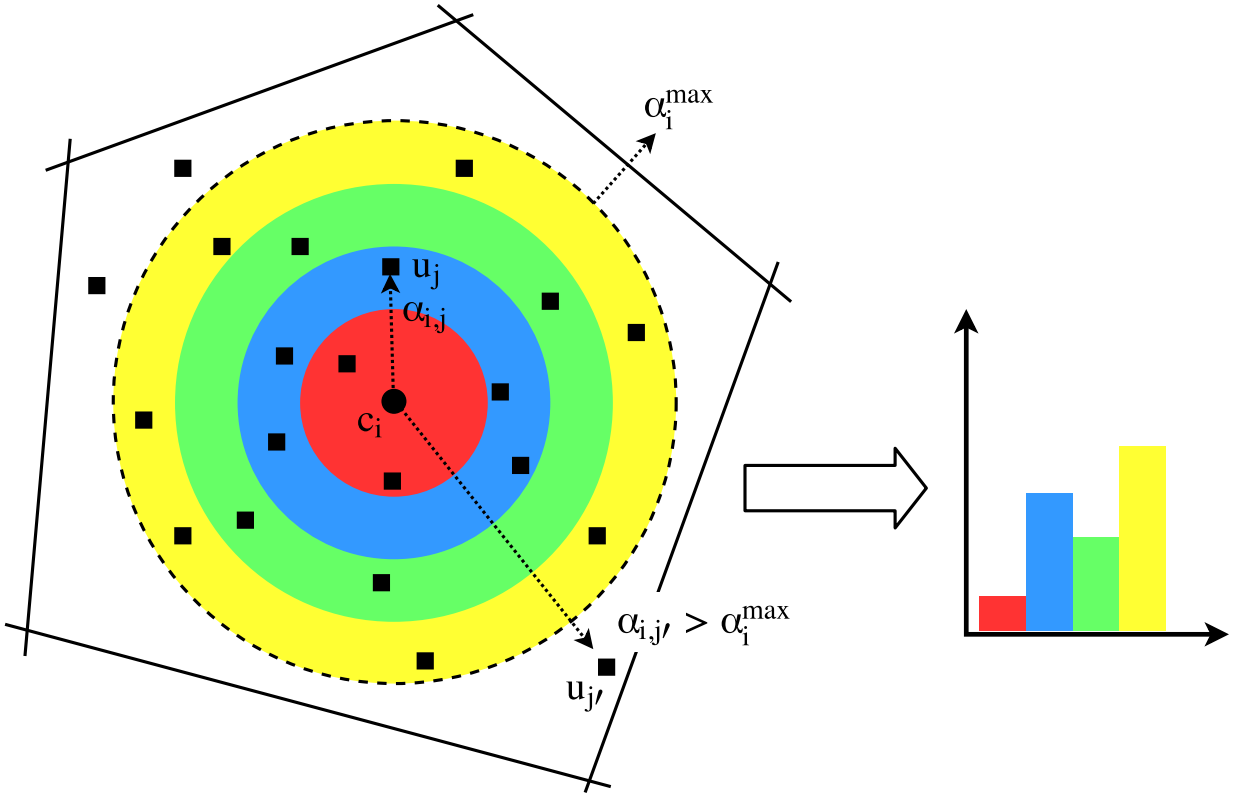


Figure 3.5: *Illustration of the BOSSA α_i^{max} parameter and histogram Z_i.*

therefore, the final BOSSA histogram $\mathbf{h}$ is of size $K \times (B+1)$.

3.6.3 BossaNova scheme

BossaNova brings four major improvements over BOSSA. The first improvement proposes the use of soft-assignment coding in the computation of each $\alpha_{i,j}$ and is based on the work of Liu *et al.* (2011). The authors argue that soft-assignment attenuates the effects of coding errors induced by the quantization of descriptor space. Considering the k-nearest descriptors to a centroid $\mathbf{c}_i$, the soft-assignment $\alpha_{i,j}$ can be computed as:

$$\alpha_{i,j} = \frac{e^{-\beta_i ||\mathbf{u}_j - \mathbf{c}_i||_2^2}}{\sum_{m=1}^{k} e^{-\beta_m ||\mathbf{u}_j - \mathbf{c}_m||_2^2}} \tag{3.8}$$

where β_i regulates the softness of the soft-assignment (the bigger, the hardest is the assignment). While Liu *et al.* (2011) uses a global β parameter, BossaNova takes advantage of the standard deviation σ_i of each cluster $\mathbf{c}_i$ to compute individual β_i values, such that $\beta_i = \sigma_i^{-2}$ (Avila *et al.*, 2013).

The second improvement was the addition of the $\alpha_i^{min} = \bar{\mathbf{c}}_i - (\lambda_{\min} \cdot \sigma_i)$ parameter. This comes from the observation that the descriptors rarely are closer than a certain threshold to a given centroid. That means some bins in each $\mathbf{z}_i$ histogram that form the BOSSA image representation are mostly zero. Equation 3.6 is rewritten as follows:

$$\mathbf{z}_{i,b} = \text{card}\left(\mathbf{u}_j | \alpha_{i,j} \in \left[\frac{b}{B}; \frac{b+1}{B}\right]\right), \text{ where } \frac{b}{B} \geq \alpha_i^{min} \text{ and } \frac{b+1}{B} \leq \alpha_i^{max} \tag{3.9}$$

The third improvement is a two step histogram normalization: primarily compute the power normalization of each $\mathbf{z}_i$ and $\mathbf{t}_i$ and then perform a ℓ_2-normalization to each of these histograms as well, according to the following set of equations:

$$\begin{aligned} \mathbf{z}_i &= \sqrt{\mathbf{z}_i} \\ \mathbf{t}_i &= \sqrt{\mathbf{t}_i} \\ \mathbf{z}_i &= \mathbf{z}_i / ||\mathbf{z}_i||_2 \\ \mathbf{t}_i &= \mathbf{t}_i / ||\mathbf{t}_i||_2 \end{aligned} \tag{3.10}$$

The forth and last improvement is to apply a weight factor s to each $\mathbf{t}_i$ value, which can be learned via cross-validation. Thus, Equation 3.7 is rewritten as follows:

$$\mathbf{h} = [[\mathbf{z}_{i,b}], s\mathbf{t}_i]^T, \text{ where } 1 \leq i \leq K \text{ and } 1 \leq b \leq B \tag{3.11}$$

Chapter 4

New Pooling Formalism

This chapter presents new ideas for the pooling phase in the Bag-of-Visual-Words model. This new formalism takes advantage on the quasi-invariability of the spatial positions of certain features in face images, and can be applied to other similar datasets.

4.1 Introduction

Most pooling strategies used in the BoVW model do not take into account the spatial structure of the objects in an image. State-of-the-art methods such as BOSSA and BossaNova (Avila *et al.*, 2011, 2013) consider the distribution of the features in relation to a codeword in the $\mathbb{R}^d$ space, but disregard the relation between features in the 2-dimensional image space. Other methods like the one proposed in Khan *et al.* (2012) try to solve this problem, but while trying to keep simplicity, it lacks important information about feature counting, and in addition suffer from inefficiencies.

Another important issue is that all these models rely on the standard SPM (Spatial Pyramid Match) (Lazebnik *et al.*, 2006) subregions division to increase their accuracy rates. This type of rigid division usually cause the final BoVW histogram to be very sparse and does not take into account the spatial distribution of the features on the images. In the following, we present some new ideas to improve recognition rates in BoVW like models.

4.2 Centers of Incidence

One of the problems of subdividing an image into r rectangular subregions of equal area regards to the spatial distribution of the features for a given codeword $\mathbf{c}_i$. As can be seen in Figure 4.1, most descriptors concentrate in specific areas of the images, leaving some regions with (almost) zero incidence of them. This causes the final histogram $\mathbf{h}$ (which represents an image after the pooling step) to be very sparse, since all the bins related to the regions with zero incidence of features will be equal to zero, which will later affects negatively the accuracy of the classifier.

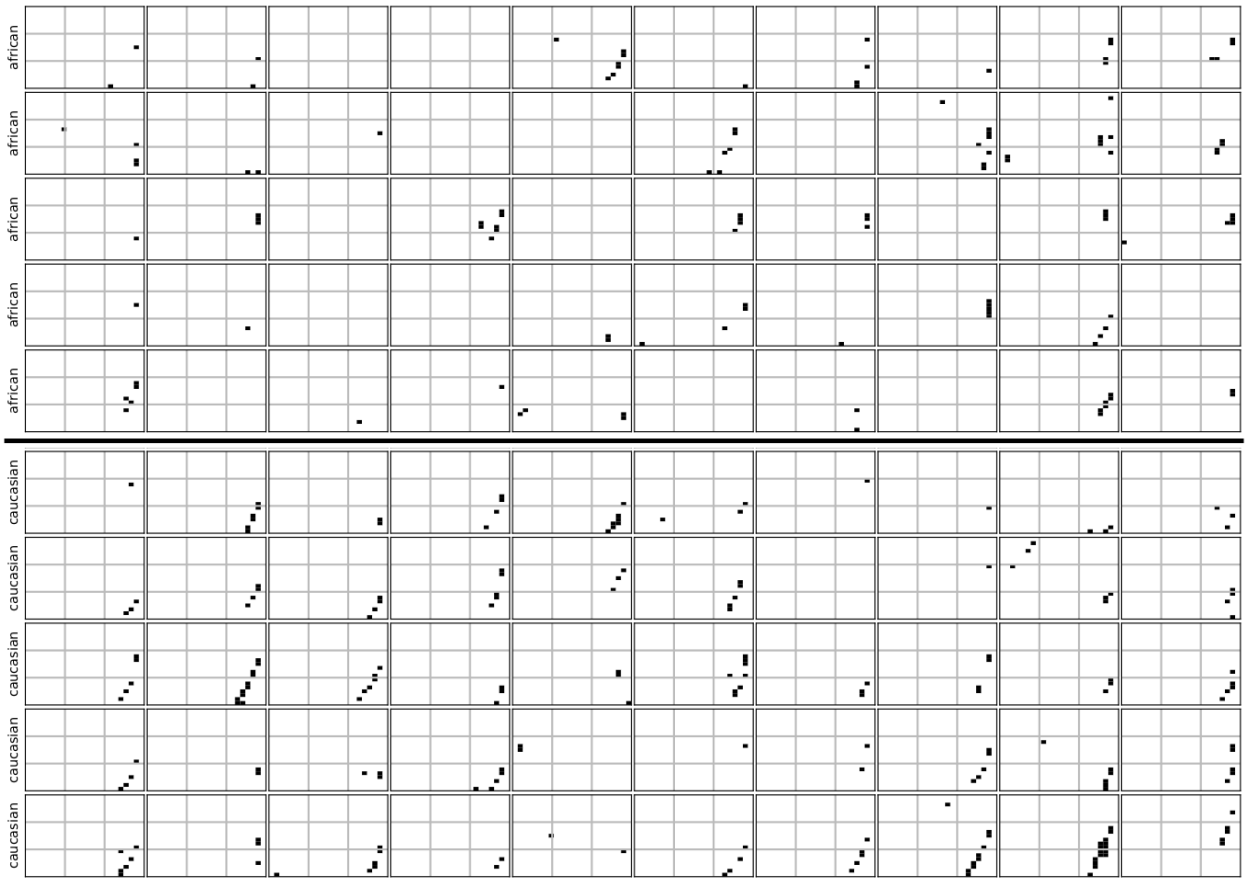
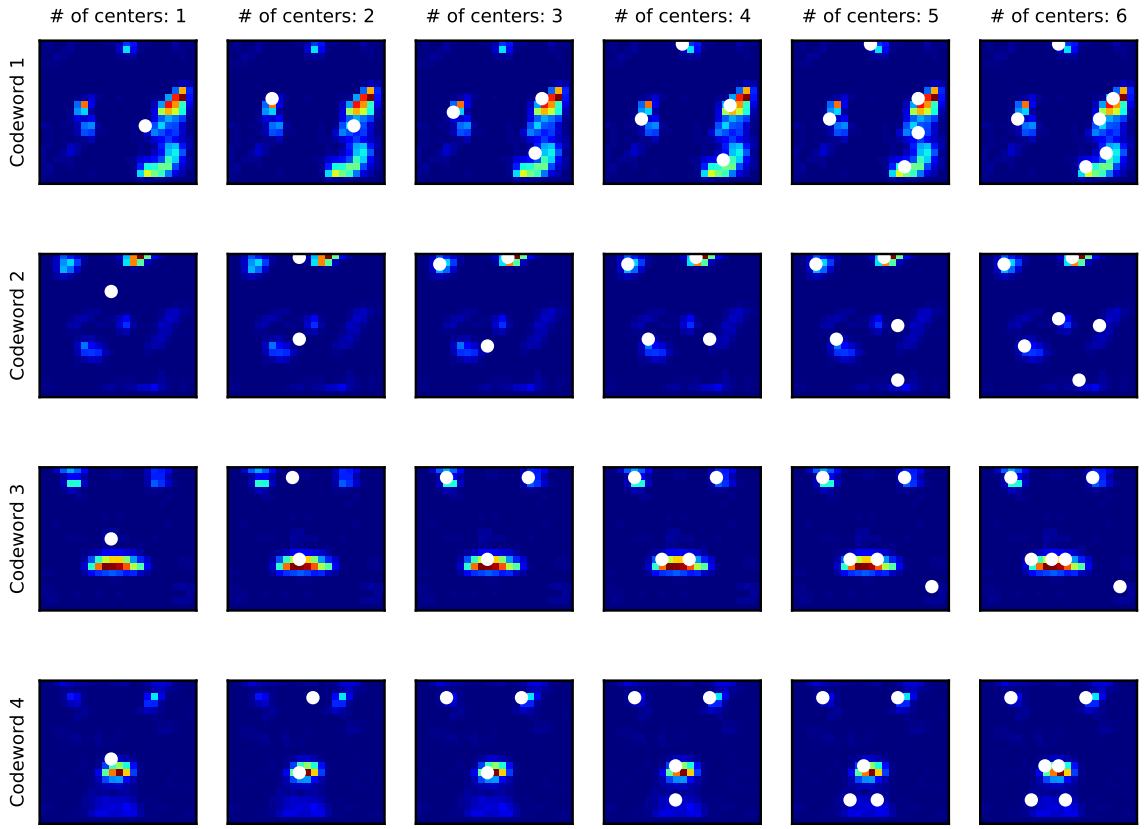


Figure 4.1: *Spatial positions of the image features encoded to a codeword* $\mathbf{c}_i$ *in 100 images of African and Caucasian subjects. Each image is subdivided into* 3×3 *regions.*

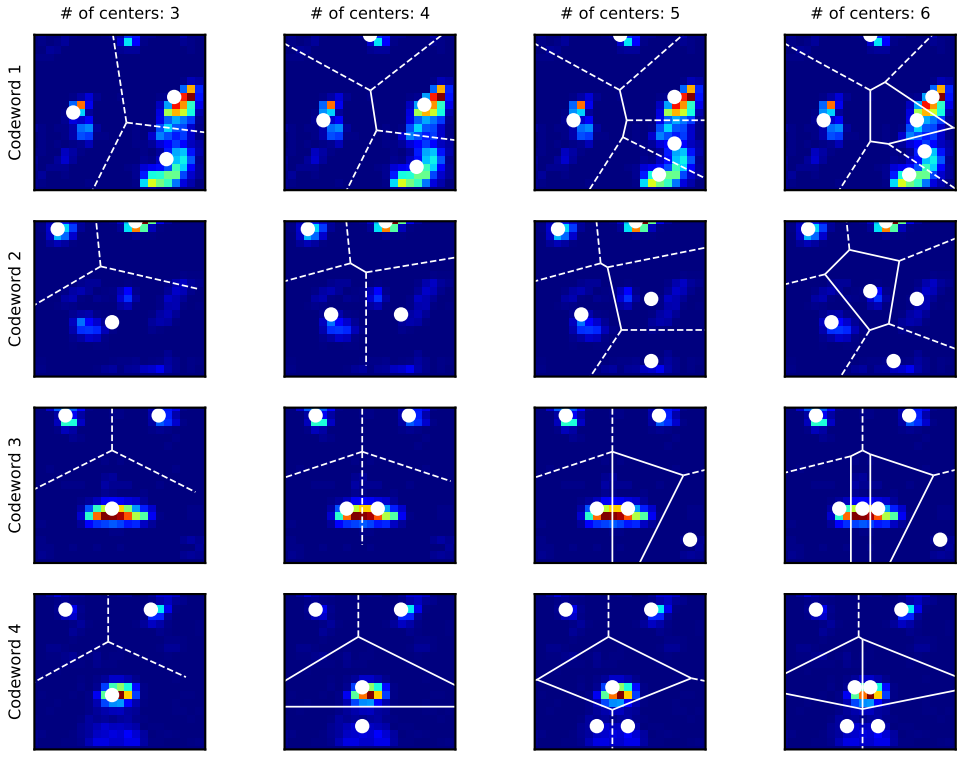
To tackle this problem, we propose a new way to represent these subregions, which uses some statistics about the spatial distribution of the image features. By knowing the spatial distribution of the image features for each codeword[1], it is possible to improve the physical arrangement of the subregions in a way that minimizes the number of unused bins in histogram $\mathbf{h}$. Note that this process assumes that each codeword $\mathbf{c}_i$ will have distinct subregions layouts, because different codewords will have their own feature spatial distributions.

Given the distribution of the features in a fixed codeword, one question that soon arises is how to divide the subregions in a way that respects the distribution layout and, at the same time, is more representative, when compared to the standard rectangular subdivision method (Lazebnik *et al.*, 2006). Moreover, such process should be computationally efficient and preferably easy to understand geometrically. For this purpose, we introduce the idea we call Centers of Incidence (CoIs).

[1]Given a codeword, we want to know all the features in the training set which are encoded to that codeword along with their relative spatial positions in the images.



a) Spatial distribution of features and the computed Centers of Incidence.



b) Computed Voronoi diagrams representing the new subdivision of spatial regions.

Figure 4.2: *2D histograms representing the distribution of the spatial positions of features in the training set for 4 codewords. The white circles represent the position of the Centers of Incidence with* $q = \{1, \ldots, 6\}$.

Algorithm 1 Centers of Incidence computation

Input: Codebook $\{\mathbf{c}_1, \ldots, \mathbf{c}_K\}$, descriptors of all training images $\mathcal{U}$, number of centers q.

Let $\mathbf{L}$ and $\mathbf{P}$ be two lists of size K, where $\mathbf{L}_w$ and $\mathbf{P}_w$ represent their w-th index.
Compute the descriptors labels $\{\alpha_1, \ldots, \alpha_{|\mathcal{U}|}\}$, where $\alpha_i = \underset{k \in \{1,\ldots,K\}}{\arg\min} ||\mathbf{u}_i - \mathbf{c}_k||_2^2$, $\forall \mathbf{u}_i \in \mathcal{U}$
for $i \leftarrow 1$, $|\mathcal{U}|$ **do**
  for $w \leftarrow 1$, K **do**
    if $\alpha_i = w$ **then**
      $[x, y] \leftarrow \mathbf{u}_i.x$, $\mathbf{u}_i.y$ {x and y are the coordinates of $\mathbf{u}_i$ in the dense SIFT grid}
      Append the coordinates $[x, y]$ to $\mathbf{L}_w$
for $w \leftarrow 1$, K **do**
  $\mathbf{P}_w \leftarrow k$-Means(points=$\mathbf{L}_w$, qty_centers=$q$) {Run k-Means for the points in $\mathbf{L}_w$}

Output: $\mathbf{P}$ {Returns the list $\mathbf{P}$, with each $\mathbf{P}_w$ being a set of q centroids}

The centers of incidence computation is detailed in Algorithm 1. This process is performed only once during the training phase and the result, a set of 2-dimensional coordinates, is stored for later use in classification. The algorithm first computes the labels of all descriptors in the training set. Then, it gets the (relative) spatial coordinate of each descriptor $\mathbf{u}$ in the (x, y) plane and insert it in a sublist that stores all the coordinates of the word w, which is the same word $\mathbf{u}$ is labeled. That sublist acts as a distribution "histogram" for a given word w. Finally, for each sublist compute the CoIs through the k-Means algorithm. Note that even we are using a clustering algorithm at the final step, its execution will be considerable fast, because the spatial information is two-dimensional and the number of CoIs (the number of clusters q) is relatively small.

The algorithm returns a list of CoIs for all codewords in the vocabulary. Figure 4.2a shows the spatial distribution of features in 4 distinct codewords and the position of the computed CoIs, represented by the white circles, for $q = \{1, \ldots, 6\}$. Figure 4.2b presents the computed Voronoi regions (Okabe *et al.*, 2009) for the same 4 codewords, for $q = \{3, \ldots, 6\}$. These will act as the new subregions from which a new BoVW histogram $\mathbf{h}$ will be computed, as previously discussed in Section 3.5.

4.3 Encoding BoVW with centers of incidence

The idea of Centers of Incidence can be seen as a generalization of the standard rectangular subregions division (Lazebnik *et al.*, 2006), since it is possible to use CoIs to achieve the same effect as these fixed regions. Figure 4.3 shows an example of the equivalence between them. Basically, CoIs can be arranged in fixed positions at the center of each subregion. Given a feature $\mathbf{u}_j$, which spatial position is inside a subregion r_i (Figure 4.3b), it will also have the point q_i (Figure 4.3c) as its closest CoI. In fact, if Voronoi diagrams are computed for the case of Figure 4.3c, they will delimit exactly the same regions as in Figure 4.3b.

Thus, this notion of distance between a feature $\mathbf{u}_j$ and a CoI q_i is the equivalent to say that $\mathbf{u}_j$ falls into the subregion r_i. Moreover, it is important to note that the definition of the

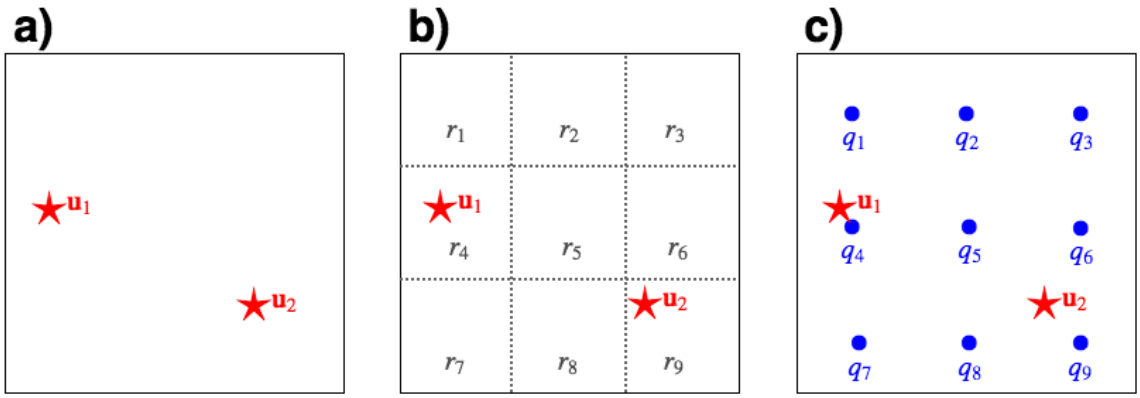


Figure 4.3: *An illustration of the equivalence between the idea of the standard rectangular subregions division (Grauman and Darrell, 2005; Lazebnik et al., 2006) and the Centers of Incidence: a) two descriptors (red stars) and their spatial positions; b) the same image divided into* 3×3 *regions; c)* 9 *centers of incidence (blue circles) positioned at the centers of each subregion. Note that the closest points from* $\mathbf{u}_1$ *and* $\mathbf{u}_2$ *are* q_4 *and* q_9 *respectively, hence* $r_i \equiv q_i$, $1 \leq i \leq 9$.

centers of incidence is more powerful, since each codeword has its own CoI layout, instead of fixed subregions with the same format as defined by Lazebnik *et al.* (2006).

The process to encode a (standard) BoVW histogram using CoIs is straightforward. First, compute the label α_j for a given descriptor $\mathbf{u}_j$, using the visual vocabulary. Then, get the (x, y) coordinates from $\mathbf{u}_j$ in the feature space (the dense SIFT/SURF grid, for example), and compute the distances between (x, y) and each CoI. Here, the ℓ_1 or ℓ_2-norms can be employed, but our experiments showed that the ℓ_2-norm commonly leads to a slightly better accuracy. After that, given the closest CoI coordinate $\mathbf{p}_i$ from (x, y), $1 \leq i \leq q$ and q being the number of CoIs per codeword, increase the bin (α_j, i) from the BoVW histogram $\mathbf{h}$ by one unit. This process is formally described in Algorithm 2.

Algorithm 2 Standard BoVW encoding using the Centers of Incidence

Input: Codebook $\{\mathbf{c}_1, \ldots, \mathbf{c}_K\}$, image descriptors $\mathbf{U} = \{\mathbf{u}_1, \ldots, \mathbf{u}_M\}$, list of centers of incidence $\mathbf{P}$, number of centers of incidence per visual word q.

1: Let $\mathbf{h}$ be a vector of size $K \cdot q$
2: Compute the descriptors labels $\{\alpha_1, \ldots, \alpha_m\}$, where $\alpha_j = \underset{k \in \{1,\ldots,K\}}{\arg\min} ||\mathbf{u}_j - \mathbf{c}_k||_2^2$, $\forall \mathbf{u}_j \in \mathbf{U}$
3: **for** $j \leftarrow 1$, M **do**
4: Compute the closest CoI $\mathbf{p}_i \in \mathbf{P}_{\alpha_j}$ from the (relative) spatial position of $\mathbf{u}_j$
5: $\mathbf{h}_{\alpha_j, i} \leftarrow \mathbf{h}_{\alpha_j, i} + 1$

Output: $\mathbf{h}$

4.4 New pooling method

Following the CoIs idea, we propose a new representation of images which extends the BoVW model. Essentially, our approach takes advantage of the quasi-invariability of the relative spatial positions of objects in some classes of datasets, such as face databases, as

exemplified in Figure 4.4. In these kind of images, it is known that there are eyes, eyebrows, nose, mouth, ears, etc. Furthermore, each image will have exactly two eyes, the eyebrows will be above them, one nose in the middle, a mouth below the nose, one ear in the left and another in the right side, and so on. Thus, the spatial relations between these objects would play an important role for image classification in such types of datasets.

Algorithm 3 Our pooling method

Input: Codebook $\{\mathbf{c}_1, \ldots, \mathbf{c}_K\}$, centroids means $\{\mu_1, \ldots, \mu_K\}$, centroids standard deviations $\{\sigma_1, \ldots, \sigma_K\}$, standard deviation threshold $\lambda_{\max}$, image descriptors $\mathbf{U}_i = \{\mathbf{u}_1, \ldots, \mathbf{u}_M\}$, list of centers of incidence $\mathbf{P}$, number of centers of incidence per visual word q, number of angle bins B, power parameter p.

for $w \leftarrow 1, K$ **do**
  Let $\mathscr{U} \subset \mathbf{U}_i$ be the set of descriptors which $\mathbf{c}_w$ is their the closest centroid
  $d_{\max} \leftarrow \mu_w + (\sigma_w \cdot \lambda_{\max})$
  Let $\mathscr{W} \subset \mathbf{U}_i$ be the set of descriptors in the range $d_{\max}$ from $\mathbf{c}_w$
  for all $\mathbf{p}_i \in \mathbf{P}_w$ **do**
    Compute τ, the number of elements in $\mathscr{U}$ where $\mathbf{p}_i$ is their closest center of incidence
    $\mathbf{t}_{w,i} \leftarrow \mathbf{t}_{w,i} + \tau$ {Equivalent to the standard BoVW pooling}
    for all $\mathbf{r}_j \in \mathscr{W}$ **do**
      $[x, y] \leftarrow \mathbf{r}_j.x \ , \ \mathbf{r}_j.y$ {x and y are the coordinates of $\mathbf{r}_j$ in the dense SIFT grid}
      Compute the angle θ between $\mathbf{p}_i$ and $[x, y]$ using Equation 3.4
      $h \leftarrow \left\lfloor (B \cdot \theta)/180 \right\rfloor + 1$
      if $h > B$ **then**
        $h \leftarrow B$ {In case $\theta = 180°$}
      $d \leftarrow ||\mathbf{p}_i - [x, y]||_1$ {ℓ_1-norm between the $\mathbf{p}_i$ and the x, y coordinates of $\mathbf{r}_j$}
      if $d \neq 0$ **then**
        $\mathbf{z}_{w,i,h} \leftarrow \mathbf{z}_{w,i,h} + d^{(-p)}$ {Inverse distance power}
      else
        $\mathbf{z}_{w,i,h} \leftarrow \mathbf{z}_{w,i,h} + 1$ {Avoids division by zero and increment by 1}
    $\mathbf{z}_{w,i} \leftarrow \mathbf{z}_{w,i}/||\mathbf{z}_{w,i}||_2$ {ℓ_2 normalization}
$\mathbf{t} \leftarrow \sqrt{\mathbf{t}}$ {Power normalization}
$\mathbf{h} \leftarrow [\mathbf{t}, \mathbf{z}]$ {Concatenate all the sub histograms $\mathbf{t}$ and $\mathbf{z}$}

Output: $\mathbf{h}$

The work by Khan *et al.* (2012) introduces a pooling algorithm based on the angles of the spatial positions of features on the images. The angular information between features gives relevant cues about the structure of the objects represented in an image. To simplify the model and make it more efficient, they limit the angle computation between features encoded to the same codeword.

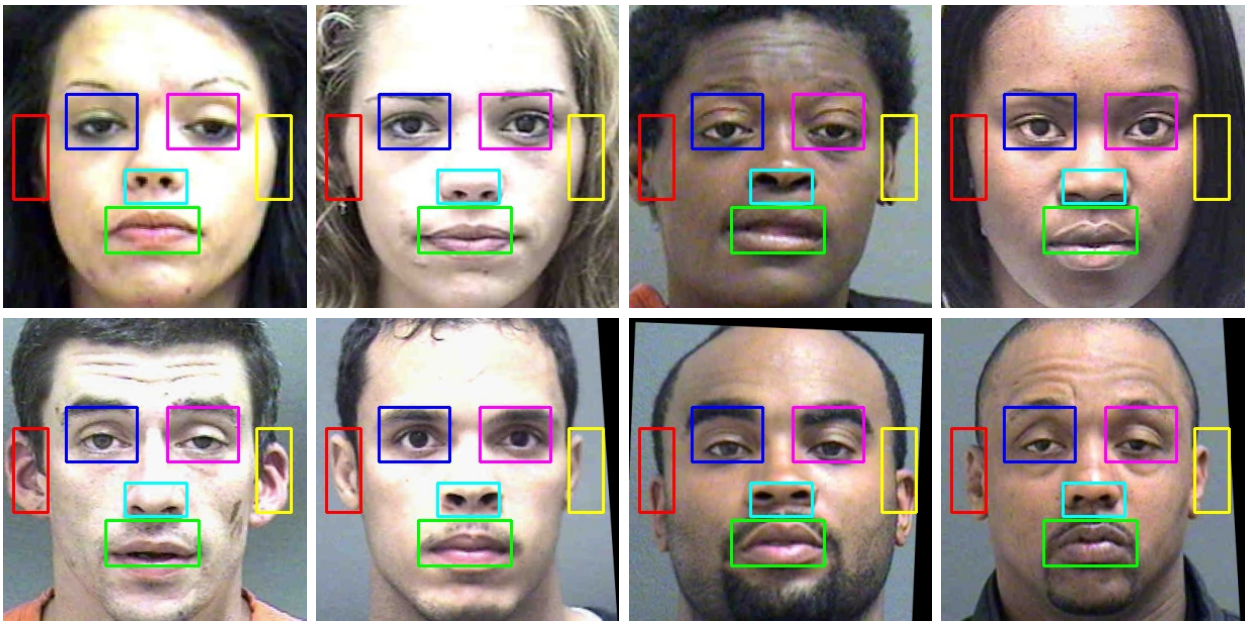


Figure 4.4: *Examples of the quasi-invariability of features in face images from the MORPH-II dataset, after face detection, alignment and rescaling. Rectangles with 6 distinct colors (each representing a specific face region) are placed over the same spatial positions in all images. Note that despite a small error margin, they all bound a certain part of the face: mouth, nose, eyes and ears.*

Algorithm 3 presents our pooling strategy based on the ideas of CoIs, angles and visual word ambiguities (Avila *et al.*, 2011, 2013; Khan *et al.*, 2012). Instead doing the intense task of computing $\binom{b}{2}$ angles between features like in Khan *et al.* (2012), with b being the number of features encoded in a given codeword, we are interested in the angles between features and the q CoIs of that codeword, which is much more efficient.

The final histogram $\mathbf{h}$ returned by the algorithm is composed by two parts ($\mathbf{t}$ and $\mathbf{z}$) as in Avila *et al.* (2011). The first part $\mathbf{t}$ encompasses a hard-assignment as in the standard BoVW pooling, and the second part $\mathbf{z}$ encodes the angular information in B bins, for each codeword. The resulting histogram also has the same size as in Avila *et al.* (2011) and Avila *et al.* (2013).

As can be seen in Algorithm 3, the set of descriptors in $\mathscr{W}$ are computed from a multiple of the mean and standard deviation of each codeword. This process is very similar to what BOSSA and BossaNova do, while it accounts to assignment ambiguities when encoding a descriptor to a codeword. It also controls the number of similar features that will be used for the computation of angles.

The angles are defined by the horizontal plane of the image and the 2D vectors formed between each feature position on the image and the spatial position of each CoI. The angles vary from 0° to 180° and are calculated according to Equation 3.4. Finally, the ℓ_1-norm d between each descriptor $\mathbf{r}_j$ coordinate (x, y) and each CoI $\mathbf{p}_i$ shall be computed, and then increment $d^{(-p)}$ (the inverse distance power) units to the bin $\mathbf{z}_{w,i,h}$ of subhistogram $\mathbf{z}_w$. This way, $\mathbf{z}_w$ keeps not only the angular information, but also how representative it is by storing how far the (x, y) coordinate of descriptor $\mathbf{r}_j$ is from the CoI $\mathbf{p}_i$.

Chapter 5

Dictionary Learning

This chapter presents a brief introduction to the field of dictionary learning and some direct contributions from this work to this area. It is also an extension of the paper Araujo *et al.* (2018), published by us at the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2018).

5.1 Introduction to Dictionary Learning

Dictionary learning is a field in signal processing which aims at finding a structure, known as dictionary, generally in the form of an overcomplete matrix (Mairal *et al.*, 2009b). The dictionary constitutes a set of vector basis called atoms, not required to be orthogonal, which can describe any element of a complex input signal via sparse representation, a form of a linear combination of these elements in which mostly are zeros (sparse).

Thus, one fundamental assumption in these techniques is that the dictionary must be inferred from the input data. The development of these methods in signal processing field was encouraged by the need to represent the input data using as few elements as possible. Dictionary learning has a wide range of applications, from data compression, signal recovery, signal/image denoising (Bao *et al.*, 2013; Dong *et al.*, 2011; Elad and Aharon, 2006), to unsupervised clustering (Ramirez *et al.*, 2010; Sprechmann and Sapiro, 2010), which will be the focus on this work.

In the sequence, Sections 5.2 and 5.3 present a brief introduction to two algorithms for dictionary learning and sparse representation, which can replace K-Means as a clustering algorithm in the BoVW model: K-SVD, and an improved version that incorporates the features labels, known as LC-KSVD. Then, Section 5.4 introduces a new method resulted from this work, which uses concave functions with robust dictionary learning to generate higher quality dictionaries and mitigate the effects outliers have in the learning process.

5.2 The K-SVD algorithm

The K-SVD (K-Singular Value Decomposition) is a dictionary learning algorithm proposed by Aharon *et al.* (2006), commonly used to create overcomplete dictionaries for sparse signal representation. It is also known as a generalization of the K-means algorithm (Aharon and Elad, 2006). Let $\mathbf{Y} \in \mathbb{R}^{N \times M}$ be a set of descriptors arranged as the columns of a matrix, where $M = \sum_{i=1}^{n} |\mathbf{U}_i|$ is the total amount of descriptors in the training set (see Section 3.4). The algorithm starts with an initial overcomplete dictionary $\mathbf{D} \in \mathbb{R}^{N \times K}$, and it aims to iteratively improve $\mathbf{D}$ to achieve sparser representations of $\mathbf{Y}$ by solving the following optimization problem:

$$\min_{D,X} \left\{ ||\mathbf{Y} - \mathbf{DX}||_F^2 \right\}, \text{ subject to } i \in [1, M], ||\mathbf{x}_i||_0 \leq T \tag{5.1}$$

where $\| \cdot \|_F$ is the Frobenius norm on matrices (Golub and Van Loan, 2012), $\mathbf{X} \in \mathbb{R}^{K \times M}$ is the sparse representation matrix, $\mathbf{x}_i$ is the i-th column of $\mathbf{X}$, $\| \bullet \|_0$ counts the number of non-zero elements of $\bullet$ and T is a sparsity constraint factor (i.e. each signal has at most T items in its decomposition).

The algorithm consists of two main steps that occur iteratively: the first involves the sparse-coding of a signal $\mathbf{Y}$, given the current dictionary estimate, producing the sparse matrix $\mathbf{X}$. The second concerns the improvement of $\mathbf{D}$, given the current sparse representations. The atoms (columns) of the dictionary are updated only one atom at a time based on the SVD decomposition, which optimizes the target function for an individual atom while the others are kept fixed. There are many algorithms to efficiently solve the sparse approximation problem, which is known to be NP-hard, such as the Orthogonal Matching Pursuit (OMP) (Pati *et al.*, 1993), the FOCUSS algorithm (Gorodnitsky and Rao, 1997) and others. Although FOCUSS gives nearly optimal solutions to the pursuit problem, it makes K-SVD more likely to be stuck in a local minima. Therefore it is recommended to use OMP rather than FOCUSS (Rubinstein *et al.*, 2008).

5.3 The LC-KSVD algorithm

The LC-KSVD (Label Consistent K-SVD) method aims to leverage the supervised information (input labels) of input signals in order to learn a discriminative dictionary for sparse signal representation. It incorporates a discriminative sparse coding error criterion and can also incorporate an optimal classification performance criterion into the objective function, which is optimized using the K-SVD algorithm. One of the advantages of this method lies on its complexity being bounded by the complexity of K-SVD (Jiang *et al.*, 2011).

Basically, each dictionary item is chosen so that it represents a subset of the training descriptors ideally from a single class. That means each dictionary item will be associated to a single class. Thus, there is an explicit correspondence between the class labels and the

dictionary items (Jiang *et al.*, 2011). Figure 5.1 shows some examples of sparse representations computed by LC-KSVD. Note that there are some peaked values due to the stronger correspondence between dictionary atoms and classes. In the sequence, we give an overview of the two variations of LC-KSVD. Further details can be found in Jiang *et al.* (2011).

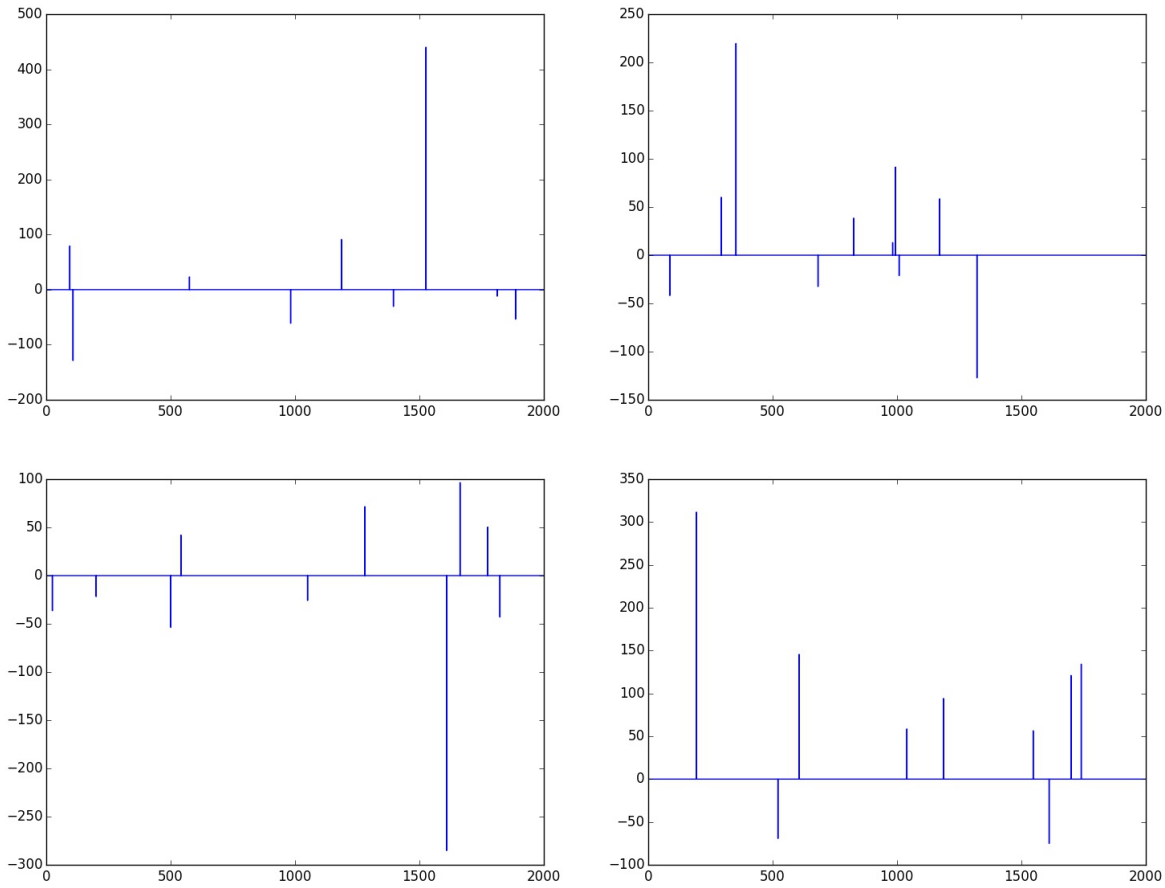


Figure 5.1: *Examples of sparse representations computed by LC-KSVD, with* $T = 10$ *(sparsity factor) and* $K = 2000$ *(number of atoms). Note the peaked nature of the histograms.*

5.3.1 LC-KSVD1

The first variation of LC-KSVD incorporated a discriminative sparse code error term $||\mathbf{Q} - \mathbf{AX}||_2^2$ to enforce that the sparse codes $\mathbf{X}$ approximate the discriminative sparse codes in matrix $\mathbf{Q} \in \mathbb{R}^{K \times M}$, where $M = \sum_{i=1}^{n} |\mathbf{U}_i|$ and $\mathbf{A} \in \mathbb{R}^{K \times K}$ is a linear transformation matrix that transforms the original sparse codes to be most discriminative in the feature space $\mathbb{R}^K$. The objective function for dictionary reconstruction is defined in Equation 5.2:

$$< \mathbf{D}, \mathbf{A}, \mathbf{X} >= \arg \min_{\mathbf{D},\mathbf{A},\mathbf{X}} ||\mathbf{Y} - \mathbf{DX}||_2^2 + \alpha||\mathbf{Q} - \mathbf{AX}||_2^2, \quad \text{subject to } i \in [1, M], ||\mathbf{x}_i||_0 \leq T \tag{5.2}$$

where α is a scalar that controlling the contribution between the reconstruction term and the label consistent regularization.

5.3.2 LC-KSVD2

The second variation of LC-KSVD includes the classification error term $||\mathbf{H}-\mathbf{WX}||_2^2$, in order to make the dictionary reconstruction optimal for classification. Thus, Equation 5.2 is rewritten as:

$$< \mathbf{D}, \mathbf{W}, \mathbf{A}, \mathbf{X} >= \arg \min_{\mathbf{D},\mathbf{W},\mathbf{A},\mathbf{X}} ||\mathbf{Y}-\mathbf{DX}||_2^2 + \alpha||\mathbf{Q}-\mathbf{AX}||_2^2 + \beta||\mathbf{H}-\mathbf{WX}||_2^2, \quad \text{subject to } i \in [1, M], ||\mathbf{x}_i||_0 \leq T \tag{5.3}$$

where matrix $\mathbf{H} \in \mathbb{R}^{\delta \times M}$ contains the class labels of input signals $\mathbf{Y}$, δ is the number of classes in the training set and matrix $\mathbf{W} \in \mathbb{R}^{\delta \times K}$ denotes the classifier parameters. α and β are scalars controlling the contribution of their respective terms.

Optimization

The K-SVD algorithm is used to find the optimal solution for all parameters simultaneously. Equation 5.2 (excluding the classification error term) and Equation 5.3 can be written as:

$$|| \begin{pmatrix} \mathbf{Y} \\ \sqrt{\alpha}\mathbf{Q} \\ \sqrt{\beta}\mathbf{H} \end{pmatrix} - \begin{pmatrix} \mathbf{D} \\ \sqrt{\alpha}\mathbf{A} \\ \sqrt{\beta}\mathbf{W} \end{pmatrix} \mathbf{X}||_2^2, \quad \text{subject to } \forall i, ||\mathbf{x}_i||_0 \leq T \tag{5.4}$$

The terms in Equation 5.4 can be represented as $\mathbf{Y}_{new} = (\mathbf{Y}^t, \sqrt{\alpha}\mathbf{Q}^t, \sqrt{\beta}\mathbf{H}^t)^t$, and $\mathbf{D}_{new} = (\mathbf{D}^t, \sqrt{\alpha}\mathbf{A}^t, \sqrt{\beta}\mathbf{W}^t)^t$, where $\mathbf{D}_{new}$ is ℓ_2 normalized column-wise. This means that the optimization of Equation 5.4 is equivalent to solving the following problem, which is exactly the problem K-SVD solves:

$$< \mathbf{D}_{new}, \mathbf{X} >= \arg \min_{\mathbf{D}_{new},\mathbf{X}} \{||\mathbf{Y}_{new} - \mathbf{D}_{new}\mathbf{X}||_2^2\}, \quad \text{subject to } \forall i, ||\mathbf{x}_i||_0 \leq T \tag{5.5}$$

5.3.3 LC-KSVD initialization

Before running LC-KSVD, parameters $\mathbf{D}_0$, $\mathbf{A}_0$ and $\mathbf{W}_0$ should be initialized. First, several iterations of K-SVD within each class are run, and then their outputs are combined to form $\mathbf{D}_0$. Thus, each dictionary item is initialized based on the class it corresponds and their labels

will remain fixed during the entire learning process. Moreover, the dictionary elements are uniformly allocated to each class, meaning the number of elements should be proportional to the dictionary size K.

Parameters $\mathbf{A}_0$ and $\mathbf{W}_0$ are initialized based on the multivariate ridge regression model, with the quadratic loss and ℓ_2-norm regularization (Jiang *et al.*, 2011):

$$\mathbf{A} = \arg\min_{\mathbf{A}} ||\mathbf{Q} - \mathbf{A}\mathbf{X}||^2 + \lambda_2 ||\mathbf{A}||_2^2 \tag{5.6}$$

which yields to Equation 5.7 for $\mathbf{A}_0$, and Equation 5.8 for $\mathbf{W}_0$:

$$\mathbf{A} = (\mathbf{X}\mathbf{X}^t + \lambda_2 \mathbf{I})^{-1}\mathbf{X}\mathbf{Q}^t \tag{5.7}$$

$$\mathbf{W} = (\mathbf{X}\mathbf{X}^t + \lambda_1 \mathbf{I})^{-1}\mathbf{X}\mathbf{H}^t \tag{5.8}$$

where the matrix $\mathbf{X}$ is computed using the initialized $\mathbf{D}_0$ and the training signals $\mathbf{Y}$.

5.4 Concave losses for robust dictionary learning

The remaining of this chapter describes our direct contributions to the field of dictionary learning and sparse coding.

5.4.1 Introduction and formalism

Formally, given a data matrix composed of n elements of dimension d, $\mathbf{X} \in \mathbb{R}^{d \times n}$ and each column being an example $\mathbf{x}_i$, the dictionary learning problem is given by:

$$\min_{\mathbf{D} \in \mathbb{R}^{d \times K}, \mathbf{A} \in \mathbb{R}^{K \times n}} \frac{1}{2} \sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{D}\mathbf{a}_i\|_2^2 + \Omega_D(\mathbf{D}) + \Omega_A(\mathbf{A}) \tag{5.9}$$

where Ω_D and Ω_A represent some constraints and/or penalties on the dictionary set $\mathbf{D}$ and the matrix coefficient $\mathbf{A}$, each column being a linear combination coefficients $\mathbf{a}_i$ so that $\mathbf{x}_i \approx \mathbf{D}\mathbf{a}_i$. Typical regularizers are sparsity-inducing penalty on $\mathbf{A}$, or unit-norm constraint on each dictionary element although a wide variety of penalties can be useful (Bach *et al.*, 2012; Rakotomamonjy, 2013; Tibshirani, 1996).

As depicted by the mathematical formulation of the problem, the learned dictionary $\mathbf{D}$ depends on training examples $\{\mathbf{x}_i\}_{i=1}^n$. However, because of the quadratic loss function in the data fitting term, $\mathbf{D}$ is in addition, very sensitive to outlier examples. Our goal here is to address the robustness of the approach to outliers. For this purpose, we consider loss functions that downweight the importance of outliers in $\mathbf{X}$ making the learned dictionary less sensitive to them.

Typical approaches in the literature, that aim at mitigating influence of outliers, use Frobenius norm or component-wise ℓ_1 norm as data-fitting term instead of the squared-

Frobenius one (Nie *et al.*, 2010; Wang *et al.*, 2016a). Some works propose loss functions such as the ℓ_q function, with $q \leq 1$ function or the capped function $g(u) = \min(u, \epsilon)$, for $u > 0$ (Jiang *et al.*, 2015; Wang *et al.*, 2013). Due to these non-smooth and non-convex loss function, the resulting dictionary learning problem is more difficult to solve than the original one given in Equation 5.9. As such, authors have developed algorithms based on an iterative reweighted least-square approaches tailored to the loss function ℓ_q or $\min(u, \epsilon)$ (Jiang *et al.*, 2015; Wang *et al.*, 2013).

Our contribution in this area is: (i) to introduce a generic framework for robust dictionary learning by considering as loss function the composition of the Frobenius norm and some concave loss functions (our framework encompasses previously proposed methods while enlarging the set of applicable loss functions); (ii) to propose a generic majorization-minimization algorithm applicable to concave, smooth or non-smooth loss functions. Furthermore, because the resulting learning problem is non-convex, its solution is sensitive to initial conditions, hence we propose a novel heuristic for dictionary initialization that helps in detecting outliers more efficiently during the learning process.

5.4.2 Framework and algorithm

In order to robustify the dictionary learning process against outliers, we need a learning strategy that puts less emphasis on examples that are not "correctly" approximated by the learned dictionary. Hence, we propose the following generic learning problem:

$$\min_{\mathbf{D},\mathbf{A}} \frac{1}{2} \sum_i F(\|\mathbf{x}_i - \mathbf{D}\mathbf{a}_i\|_2^2) + \Omega_D(\mathbf{D}) + \Omega_A(\mathbf{A}). \tag{5.10}$$

where $F(\bullet)$ is a function over $\mathbb{R}_{>0}$. Note that in the sequel, we will not focus on the penalty and constraints over the dictionary elements and coefficients $\mathbf{A}$. Hence, we consider them as the classical unit-norm constraint over $\mathbf{d}_j$ and the ℓ_1 sparsity-inducing penalty over $\{\mathbf{a}_i\}$.

The concavity of F is crucial for robustness as it helps in down-weighting influence of large $\|\mathbf{x}_i - \mathbf{D}\mathbf{a}_i\|_2$. For instance, if we set $F(\bullet) = \sqrt{\bullet}$, the above problem is similar to the convex robust dictionary learning proposed by Wang *et al.* (2016a). In order to provide better robustness, our goal is to introduce a generic form of F that leads to a concave loss with respect to $\|\mathbf{x}_i - \mathbf{D}\mathbf{a}_i\|_2$, instead of a linear, yet concave one as in Wang *et al.* (2016a).

In this work, we emphasize robustness by considering F as the composition of two concave functions $F(\bullet) = g(\bullet) \circ \sqrt{\bullet}$, with g a non-decreasing concave function over $\mathbb{R}_{>0}$, such as those used for sparsity-inducing penalties. Typically, $g(\bullet)$ can be the $q-$power, $q \leq 1$ function-inducing u^q, the log function $\log(\epsilon + u)$, the SCAD function (Fan and Li, 2001), or the capped-ℓ_1 function $\min(u, \epsilon)$, or the MCP function (Zhang *et al.*, 2010). A key property on F is that concavity is preserved by the composition of some specific concave functions as proved by the following lemma which proof is omitted due to its simplicty.

Lemma 1 *Let g be a non-decreasing concave function on $\mathbb{R}_{>0}$ and h be a concave function*

on a domain Ω to $\mathbb{R}_{>0}$, then $g \circ h$ is concave. Furthermore, if g is a strictly increasing function and h strictly concave, then $g \circ h$ is strictly concave.

Definition 2 *Let $C \subset \mathbb{R}^d$ be a convex set, and let $h : C \to \mathbb{R}$ be a concave function. We say a vector $\mathbf{p}$ is a supergradient of h at the point u_0 if for every u it satisfies the supergradient inequality,*

$$h(u_0) + \mathbf{p} \cdot (u - u_0) \geq h(u)$$

for every concave function h, the set of all supergradients of h at u_0 is called the superdifferential of h at u_0.

In our framework, h is the square-root function with $\Omega = \mathbb{R}_{>0}$. In addition, functions g, such as those given above, are either a concave or strictly concave functions and are all non-decreasing, hence $F = g \circ h$ is concave. Owing to concavity, for any u_0 and u in $\mathbb{R}_{>0}$,

$$F(u) \leq F(u_0) + F'(u_0)(u - u_0)$$

where $F'(u_0)$ is an element of the superdifferential of F at u_0. As F is concave, the superdifferential is always non-empty and if F is smooth at u_0, then $F'(u_0)$ is simply the gradient of F at u_0. However, since F is a composition of functions, in a non-smooth case, computing superdifferential is difficult unless the inner function is a linear function (Rockafellar , 2015). Next lemma provides a key result showing that a supergradient of $g \circ \sqrt{\bullet}$ can be simply computed using chain rule because $\sqrt{\bullet}$ is a bijective function on $\mathbb{R}_{>0}$ to $\mathbb{R}_{>0}$ and g is non-decreasing.

Lemma 3 *Let g a non-decreasing concave function on $\mathbb{R}_{>0}$ and h a bijective differentiable concave function on a domain $\mathbb{R}_{>0}$ to $\mathbb{R}_{>0}$, then if g_1 is a supergradient of g at z then $g_1 \cdot h'(s)$ is a supergradient of $g \circ h$ at a point s so that $z = h(s)$.*

Proof As $g_1 \in \partial g(z)$, we have $\forall y, g(y) \leq g(z) + g_1 \cdot (y - z)$. Owing to bijectivity of h, define t and s so that $y = h(t)$ and $z = h(s)$. In addition, concavity of h gives $h(t) - h(s) \leq h'(s)(t - s)$ and because g is non-decreasing, $g_1 \geq 0$. Combining everything, we have $g_1 \cdot (y - z) = g_1 \cdot (h(t) - h(s)) \leq g_1 h'(s)(t - s)$. Thus $\forall t, g(h(t)) \leq g(h(s)) + g_1 h'(s)(t - s)$ which concludes the proof since g_1 is a supergradient of g at $h(s)$.

Based on the above majorizing linear function property of concave functions and because in our case $F'(u_0)$ can easily be computed, we consider a majorization-minimization approach for solving Problem 5.10. Our iterative algorithm consists, at iteration τ, in approximating the concave loss function F at the current solution $\mathbf{D}_\tau$ and $\mathbf{A}_\tau$ and then solve the resulting approximate problem for $\mathbf{D}$ and $\mathbf{A}$. This yields in solving:

$$\min_{\mathbf{D},\mathbf{A}} \frac{1}{2} \sum_i \mathbf{s}_i \|\mathbf{x}_i - \mathbf{D}\mathbf{a}_i\|_2^2 + \Omega_D(\mathbf{D}) + \Omega_A(\mathbf{A}) \tag{5.11}$$

Algorithm 4 The proposed Robust DL method

Input: Data matrix $\mathbf{X} \in \mathbb{R}^{d \times n}$, dictionary size K, sparsity factor λ, ϵ, number of iterations M.

if $(K > d)$ and (*use undercomplete initialization*) **then**
  Initialize $\mathbf{D}$ and $\mathbf{s}$ with Algorithm 8
else
  random initialization of $\mathbf{D}$, $\mathbf{A}$
  $\mathbf{s}_j = 1$ for $j = 1, \ldots, n$
for $i = 1$ to M **do**
  repeat
    Update $\mathbf{D}$ with Algorithm 5
    for $j = 1$ to n **do**
      $\mathbf{a}_j \leftarrow \frac{1}{2}||\mathbf{x}_j - \mathbf{D}\mathbf{a}||_2^2 + \frac{\lambda}{\mathbf{s}_j}||\mathbf{a}||_1$
  until convergence
  for $j = 1$ to n **do**
    update $\mathbf{s}_j$ according to Equation 5.12

Output: $\mathbf{D}$, $\mathbf{s}$

Algorithm 5 Dictionary update

Input: Data matrix $\mathbf{X}$, dictionary $\mathbf{D}$, coefficient matrix $\mathbf{A}$, weights vector $\mathbf{s}$.

$\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_K] \in \mathbb{R}^{K \times K} = \sum_{i=1}^{n} \mathbf{s}_i \mathbf{a}_i \mathbf{a}_i^T$
$\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_K] \in \mathbb{R}^{d \times K} = \sum_{i=1}^{n} \mathbf{s}_i \mathbf{x}_i \mathbf{a}_i^T$
repeat
  for $j = 1$ to K **do**
    $\mathbf{u}_j \leftarrow \frac{1}{\mathbf{B}_{jj}}(\mathbf{z}_j - \mathbf{D}\mathbf{b}_j) + \mathbf{d}_j$ {update the j-th column of $\mathbf{D}$}
    $\mathbf{d}_j \leftarrow \frac{1}{\max(||\mathbf{u}_j||_2, 1)}\mathbf{u}_j$
until convergence

Output: $\mathbf{D}$

where $\mathbf{s}_i = [g \circ \sqrt{\bullet}\,]'$ at $\mathbf{D}_\tau$ and $\mathbf{a}_{\tau,i}$. Since, we have

$$[g \circ \sqrt{\bullet}\,]'(u_0) = \frac{1}{2\sqrt{u_0}} g'(\sqrt{u_0})$$

weights $\mathbf{s}_i$ can be defined as

$$\mathbf{s}_i = \frac{g'(\|\mathbf{x}_i - \mathbf{D}_\tau \mathbf{a}_{\tau,i}\|_2)}{2\|\mathbf{x}_i - \mathbf{D}_\tau \mathbf{a}_{\tau,i}\|_2}. \tag{5.12}$$

This definition of $\mathbf{s}_i$ can be nicely interpreted. Indeed, if g is so that $\frac{g'(u)}{u}$ becomes small as u increases, examples with large residual values $\|\mathbf{x}_i - \mathbf{D}_\tau \mathbf{a}_{\tau,i}\|_2$ have less importance in the learning Problem 5.11 because their corresponding values $\mathbf{s}_i$ are small.

Note how the composition $g \circ \sqrt{\bullet}$ allows us to write the data fitting term with respect to the squared residual norm so that at each iteration, the problem to solve is simply a weighted smooth dictionary learning problem, convex in each of its parameters, that can be addressed using off-the-shelf tools. As such, it can be solved alternatively for $\mathbf{D}$ with fixed $\mathbf{A}$ and then for $\mathbf{A}$ with fixed $\mathbf{D}$. For fixed $\mathbf{A}$, the optimization problem is thus:

$$\min_{\mathbf{D}} \frac{1}{2} \sum_i \|\tilde{\mathbf{x}}_i - \mathbf{D}\tilde{\mathbf{a}}_i\|_2^2 + \Omega_D(\mathbf{D}) \tag{5.13}$$

where $\tilde{\mathbf{x}}_i = \sqrt{\mathbf{s}_i}\mathbf{x}_i$ and $\tilde{\mathbf{a}}_i = \sqrt{\mathbf{s}_i}\mathbf{a}_i$. This problem can be solved using a proximal gradient algorithm or block-coordinate descent algorithm as given in Mairal *et al.* (2009a). For fixed $\mathbf{D}$, the problem is separable in $\mathbf{a}_i$ and each sub-problem is equivalent to a Lasso problem with regularization $\frac{\lambda}{\mathbf{s}_i}$.

The above algorithm is generic in the sense that it is applicable to any continuous concave and non-decreasing function g, even non-smooth ones. This is in constrast with algorithms proposed in Wang *et al.* (2013) and Jiang *et al.* (2015) which have been tailored to some specific functions g. In addition, the convergence in objective value of the algorithm is guaranteed for any of these g functions, by the fact that the objective value in Equation 5.10 decreases at each iteration while it is obviously lower bounded.

5.4.3 Online variant

We also propose a variant of Algorithm 4 to deal with mini-batches. Assuming that the training data $\mathbf{X}$ is composed by i.i.d. (independent and identically distributed) samples, Algorithm 6 picks h elements at a time and then uses standard sparse coding steps to compute a set of decompositions $[\mathbf{a}_{t-h+1}, \ldots, \mathbf{a}_t]$ of the mini-batch $[\mathbf{x}_{t-h+1}, \ldots, \mathbf{x}_t]$ over $\mathbf{D}_{t-1}$, the dictionary calculated in the previous iteration.

Algorithm 6 The proposed Online Robust DL method

Input: Data matrix $\mathbf{X} \in \mathbb{R}^{d \times n}$, dictionary size K, sparsity parameter λ, ϵ, number of iterations M, mini-batch data size h.

1: Initialize dictionary $\mathbf{D}$ and coefficient matrix $\mathbf{A}$.
2: Initialize $\mathbf{B}$ and $\mathbf{Z}$ as zero matrices.
3: $\mathbf{s}_j = 1$ **for** $j = 1$ to n
4: **for each** $\{\mathbf{x}_{t-h+1}, \ldots, \mathbf{x}_t\}$ in $\mathbf{X}$ **do**
5: **for** $i = 1$ to M **do**
6: **repeat**
7: $\hat{\mathbf{B}} \leftarrow \mathbf{B} + \sum_{j=(t-h+1)}^{t} \mathbf{s}_j \mathbf{a}_j \mathbf{a}_j^T$
8: $\hat{\mathbf{Z}} \leftarrow \mathbf{Z} + \sum_{j=(t-h+1)}^{t} \mathbf{s}_j \mathbf{x}_j \mathbf{a}_j^T$
9: $\mathbf{D} \leftarrow$ Algorithm 7$\big([\mathbf{x}_{t-h+1}, \ldots, \mathbf{x}_t], \mathbf{D}, \hat{\mathbf{B}}, \hat{\mathbf{Z}}\big)$
10: **for** $j = (t-h+1)$ to t **do**
11: $\mathbf{a}_j \leftarrow \frac{1}{2}||\mathbf{x}_j - \mathbf{D}\mathbf{a}||_2^2 + \frac{\lambda}{\mathbf{s}_j}||\mathbf{a}||_1$
12: **until** convergence
13: **for** $j = (t-h+1)$ to t **do**
14: update $\mathbf{s}_j$ according to Equation 5.12
15: $\mathbf{B} \leftarrow \hat{\mathbf{B}}$, $\mathbf{Z} \leftarrow \hat{\mathbf{Z}}$

Output: $\mathbf{D}$, $\mathbf{s}$

In the inner loop (starting from line 6), matrices $\mathbf{B}$ and $\mathbf{Z}$ carry all the "past information" from coefficients $[\mathbf{a}_1, \ldots, \mathbf{a}_{t-h}]$. Information from new data is computed in $\sum_{j=(t-h+1)}^{t} \mathbf{s}_j \mathbf{a}_j \mathbf{a}_j^T$

and $\sum_{j=(t-h+1)}^{t} \mathbf{s}_j \mathbf{x}_j \mathbf{a}_j^T$, which are combined with that past information and then stored in matrices $\hat{\mathbf{B}}$ and $\hat{\mathbf{Z}}$, respectively, thus keeping track of past and current coefficients $[\mathbf{a}_1, \ldots, \mathbf{a}_t]$. A very similar approach is used in other online dictionary learning methods such as the ones proposed by Mairal *et al.* (2009a), Mairal *et al.* (2010) and Lu *et al.* (2013).

A heuristic to speed up convergence is proposed by Mairal *et al.* (2009a). Since the information added to matrices $\hat{\mathbf{B}}$ and $\hat{\mathbf{Z}}$ have the same weight as the previous mini-batches, a straightforward and natural idea is to rescale the "past" information so that the newer coefficients $[\mathbf{a}_{t-h+1}, \ldots, \mathbf{a}_t]$ have more weight. This can be done by replacing lines 7 and 8 of Algorithm 6 with the following:

$$\begin{cases} \hat{\mathbf{B}} \leftarrow \beta \mathbf{B} + \sum_{j=(t-h+1)}^{t} \mathbf{s}_j \mathbf{a}_j \mathbf{a}_j^T \\ \hat{\mathbf{Z}} \leftarrow \beta \mathbf{Z} + \sum_{j=(t-h+1)}^{t} \mathbf{s}_j \mathbf{x}_j \mathbf{a}_j^T \end{cases} \tag{5.14}$$

with $\beta = \frac{\gamma+1-h}{\gamma+1}$, such that $\gamma = \tau h$ if $\tau < h$, or $\gamma = h^2 + \tau - h$ if $\tau \geq h$, where τ is the number of iterations of the outer loop (line 4 in Algorithm 6).

The online dictionary update step is presented in Algorithm 7, and it is a slightly modified version of its batch counterpart (Algorithm 5). In this setup, matrices parameters $\mathbf{B}$ and $\mathbf{Z}$ are previously computed iteratively inside the loop of Algorithm 6, according to the new incoming mini-batch.

Algorithm 7 Online dictionary update

Input: Dictionary $\mathbf{D}$, matrix $\hat{\mathbf{B}} = [\hat{\mathbf{b}}_1, \ldots, \hat{\mathbf{b}}_K] \in \mathbb{R}^{K \times K}$, matrix $\hat{\mathbf{Z}} = [\hat{\mathbf{z}}_1, \ldots, \hat{\mathbf{z}}_K] \in \mathbb{R}^{d \times K}$.

1: **repeat**
2: **for** $j = 1$ to K **do**
3: $\mathbf{u}_j \leftarrow \frac{1}{\hat{\mathbf{B}}_{jj}}(\hat{\mathbf{z}}_j - \mathbf{D}\hat{\mathbf{b}}_j) + \mathbf{d}_j$ {update the j-th column of $\mathbf{D}$}
4: $\mathbf{d}_j \leftarrow \frac{1}{\max(\|\mathbf{u}_j\|_2, 1)} \mathbf{u}_j$
5: **until** convergence

Output: $\mathbf{D}$

5.4.4 Undercomplete initialization

The problem our RDL method solves is non-convex, thus its solution is very sensitive to initialization. The existence of outliers in the data matrix $\mathbf{X}$ amplifies the effect, resulting in a biased dictionary $\mathbf{D}$, which does not generalize the non-outliers properly. Hence, a suitable initialization of the weights in $\mathbf{s}$ is essential in our iterative algorithm, based on Equation 5.11.

By identifying the outliers before learning, it would be possible to assign $\mathbf{s}_i \approx 0$ to those samples, therefore they would become irrelevant for the dictionary learning problem. Although this process seems straightforward, detecting outliers in a set of samples is a hard problem itself (Chandola *et al.*, 2007).

Algorithm 8 Undercomplete initialization

Input: Data matrix $\mathbf{X}$, dictionary $\mathbf{D} \in \mathbb{R}^{d \times K}$, with $d < K$, number of atoms in each batch $b < K$, parameters λ and ϵ.

1: $N \leftarrow \left\lceil \frac{K}{b} \right\rceil$ {number of batches}
2: $\mathbf{s} = \mathbf{0}$
3: Initialize $\mathbf{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_K]$ as a zero matrix
4: **for** $i = 0$ to $(N - 1)$ **do**
5: I = indices related to i-th batch
6: $\hat{\mathbf{D}}, \hat{\mathbf{s}} \leftarrow$ Algorithm 4$\big(\mathbf{X}, |I|, \lambda, \epsilon, 1\big)$
7: $\mathbf{D}_I \leftarrow \hat{\mathbf{D}}$ {assign learned dictionary to the appropriate indices}
8: $\mathbf{s} \leftarrow \mathbf{s} + \hat{\mathbf{s}}$ {accumulate weights}
9: $\mathbf{s} \leftarrow \frac{\mathbf{s}}{N}$ {compute average}

Output: $\mathbf{D}$, $\mathbf{s}$

In Algorithm 8 we propose an initialization strategy to tackle this problem. It assumes heuristically that if most samples belong to a linear subspace of $\mathbb{R}^d$, while outliers are outside this subspace, then those outliers could be easier identified by an undercomplete dictionary rather than an overcomplete one. Moreover, if the sparsity penalty is weak enough, it would be possible to approximate quite well any sample through an overcomplete dictionary, therefore even an outlier would be assigned with a large $\mathbf{s}_i$ value.

Thus, if the dimension of the problem is less than the number of dictionary elements, as it is in an overcomplete scheme, our heuristic initializes the dictionary $\mathbf{D}$ and weights vector $\mathbf{s}$ by learning mini-batches of dictionary atoms of size b, with $b < d$, as shown in the **for** loop in Algorithm 8. Each iteration of the loop calls one iteration of Algorithm 4, with the size of the dictionary (the parameter K) equals to b. If there is a small portion of outliers in the samples in $\mathbf{X}$, $\hat{\mathbf{s}}_i$ (line 6) will probably be bigger for non-outliers and smaller for outliers. A new $\hat{\mathbf{s}}$ vector is computed for each mini-batch, and its values accumulated in $\mathbf{s}$ (line 8). In the end, an average of $\mathbf{s}$ is computed by dividing its values by N, the number of mini-batches.

It is important to observe that, on average, $\hat{\mathbf{s}}$ will be bigger for non-outliers, once its value is derived from Equation 5.12 through one iteration of Algorithm 4. Thus, the concave function g will limit the effects the outliers cause on the reconstruction error $\|\mathbf{x}_i - \mathbf{D}\mathbf{a}_i\|_2$. Besides, during that iteration, the dictionary $\mathbf{D}$ is going to be undercomplete, which will also cause the error to be bigger for the outliers, in the average case.

Chapter 6

Experiments

Given the ideas introduced and proposed in the previous chapters, here we apply them into experiments and present empirical results. The first part focuses on experiments with synthetic data, while the second introduces the datasets, data preparation, sampling strategies and experimental setups.

6.1 Concave robust dictionary learning with synthetic data

To evaluate our RDL method presented in Section 5.4, we propose two experiments with synthetic generated datasets with outliers to demonstrate its robustness against them. The first experiment shows the visual behavior of our RDL algorithm using 2D samples, and the other displays its accuracy on data with higher dimension.

6.1.1 Synthetic experiment with 2D data

This experiment aims to visually show the capabilities of our RDL algorithm in the detection of outliers. This is much easier to observe and understand in two dimensions since the generated dataset can be trivially plotted.

It begins with two clusters are generated using two Gaussian distributions, each containing 250 points along with 50 outliers. As can be seen in Figure 6.1, the outliers are represented as the red triangles (top left sub-figure), far away from the clusters, resulting in a total of 550 points. The points are clustered using K-SVD (Aharon *et al.*, 2006) as well as our RDL method with $g(u) = u$ and also $g(u) = \log(\epsilon + u)$ functions, respectively.

We then compare how many of the original outliers are among the 50 highest reconstruction values. Our method, using the log function, proved to be the most robust against outliers, with 47 from the 50 true outliers detected. It is followed by the variant with the identity function, which identified 27 outliers, and finally by K-SVD, which was naturally not able to identify any of the original outliers. This example also shows that concavity of

function g (see Section 5.4.2) helps in better identifying outliers.

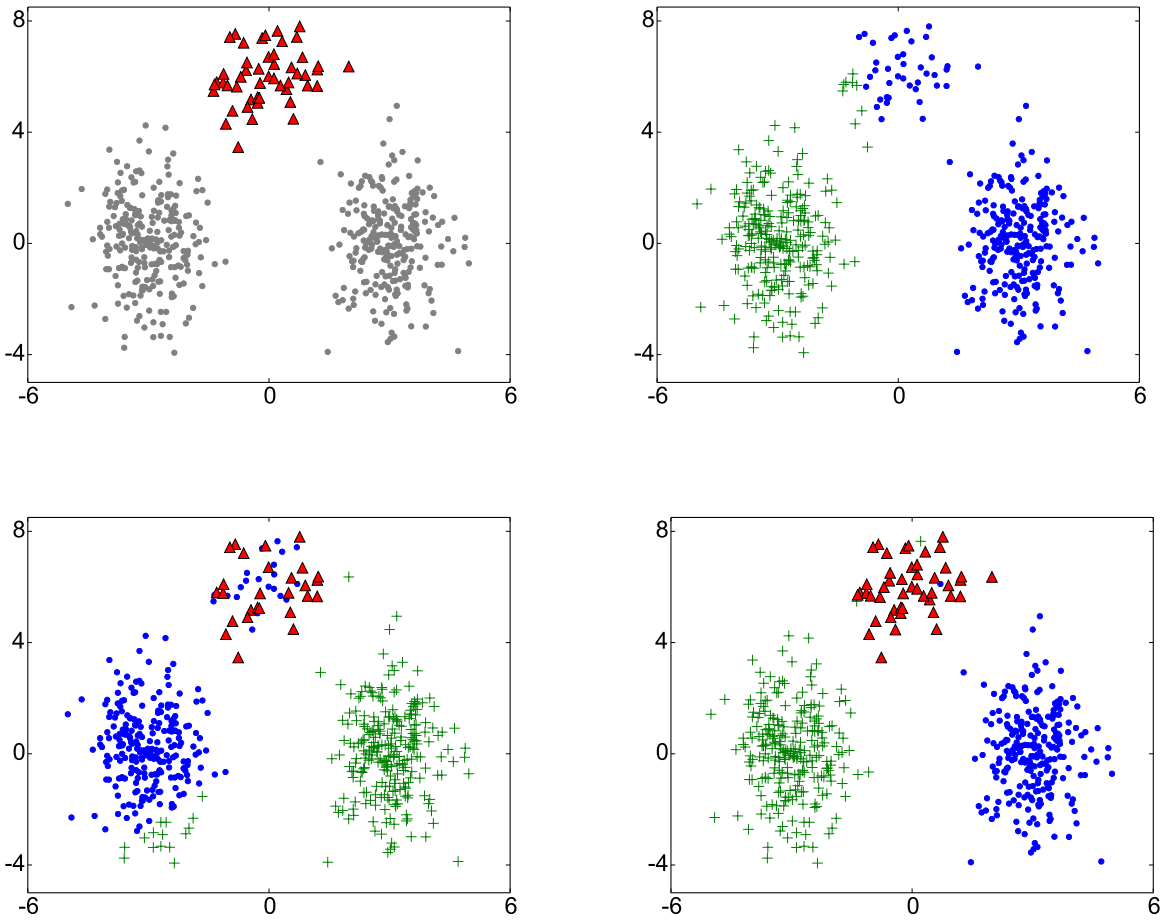
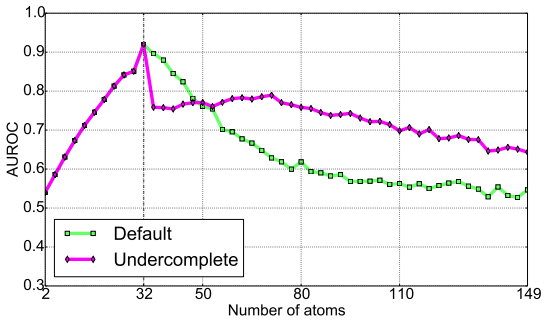


Figure 6.1: *Synthetic 2D data drawn from two Gaussian distributions. The outliers are represented as the red triangles. (top-left) Original data with outliers. (top-right) Clustering with K-SVD. (bottom-left) Clustering with our RDL method with the (identity) function $g(u) = u$. (bottom-right) Clustering our RDL method using the function $g(u) = \log(\epsilon + u)$.*

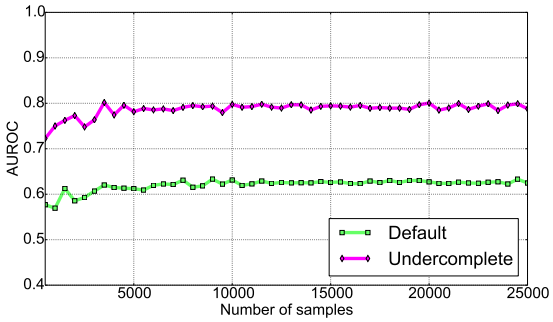
6.1.2 Synthetic experiment with high dimension data

After showing the behavior of our RDL methods in two dimensions, our aim with these experiments is to prove the capabilities of those methods to detect outliers under higher dimensional data. To achieve this, we generate synthetic data of 32 dimensions using a similar approach as described by Lu *et al.* (2013) to create the data based on a dictionary and sparse coefficients. Basically, generate n sparse coefficients $\{\beta_1, \ldots, \beta_n\} \in \mathbb{R}^{\kappa \times n}$ with a given sparsity ratio (20%, for example), a dictionary $\mathbf{D} \in \mathbb{R}^{32 \times \kappa}$, and finally the data $\mathbf{x}_i = \mathbf{D}\beta_i + \xi_i$, with $1 \leq i \leq n$. The term ξ_i is additive noise to simulate outliers.

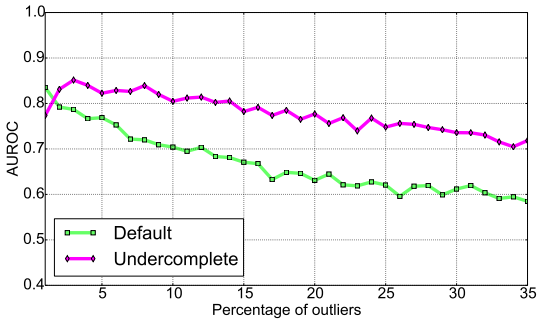
The metric adopted to compare the results is the AUC Curve (AUROC) (Fawcett, 2006) of outlier scores $\{\mathbf{s}_i\}$ after executing Algorithm 4: outliers should have scores $1/s_i$ larger than non-outliers, and each point in Figure 6.2 is the average of 5 runs using newly generated



a) Different dictionary sizes, with 1,000 samples and 10% are outliers.



b) Different number of samples, where 10% are outliers.



c) Different outlier ratios (%), with 1,000 samples and 64 atoms.

Figure 6.2: *Performance of our RDL method with multidimensional data using the standard and undercomplete initialization schemes.*

data.

We observe that the behavior for both lines is the same in Figure 6.2a until the number of atoms reach 32, since $K \leq d$ and the condition in the first line of Algorithm 4 is not met. The performance of the undercomplete initialization method also deteriorates for dictionary sizes a little bit greater than d, but as far as K starts to increase it becomes evident that this method outperforms the default initialization. Figure 6.2b shows that our method stays very stable independent of the number of samples, given a constant outlier ratio, regardless of the initialization method. Finally, Fig. 6.2c shows the behavior of both initialization strategies in scenarios where the outlier proportion changes. It can be noticed that the AUROC values decrease slowly as long as the number of outliers in the samples increase. This is natural since when the proportion of outliers is large, outliers can hardly be considered outliers anymore.

6.2 MORPH-II, FERET and AR databases

The design of a face image database is usually guided by certain specific hypothesis and objectives. Therefore, the databases are somehow different one from the others. For example, the SCfaceDB (Tome *et al.*, 2013) has around 4,000 images from 130 subjects taken from cameras in an indoor environment, with manually assigned landmarks. The LFW (Labeled Faces in the Wild) (Learned-Miller *et al.*, 2016) has 13,000 images of faces from 1,680 subjects, all collected from the Web. Most of these images were taken in unconstrained environments, which makes it a challenging database, but some images have more than one face in them. On the other hand, the 10k US Adult Faces Database (Bainbridge *et al.*, 2013) has around 10,000 images, mostly taken from Google Images, and it was originally created for a study on the memorability of face photographs. All the faces of this database had their backgrounds removed, while the faces are presented inside a circle.

Although some of these face databases have labels for ethnicity or gender, most of them were not primarely designed to be used in these kinds of problems. Each one have its own drawback: the dataset is considerable small, the classes are very unbalanced (some classes have a very limited number of images), the presence of more than one subject in one image, there is some level of occlusion (like the ones provoked by the background removal in the 10k US Adult Faces Database), and so on.

In the following we describe in more detail three face image datasets: MORPH-II, FERET and AR, used to assess the performance of the proposed approaches. They were elected to be used in this work because they have labels for ethnicity (the first two) and gender and they were used extensively in other studies about this problem.

MORPH (Ricanek Jr and Tesafaye, 2006) is a face image dataset that consists of two "albums": Albums 1 and 2. For this study, we chose Album 2 (better known as MORPH-II) since it contains around 55,000 frontal face images and includes meta data for gender and ethnicity. There are 5 classes, unbalanced distributed in the following way: African (77.2%), Caucasian (19.2%), Hispanic (3.2%), Asian ($< 0.3\%$) and "Others" ($< 0.15\%$). Furthermore, gender distribution inside each ethnicity is also unbalanced, with the predominance of males from 70% up to 94%, depending on the class.

The FERET database (Phillips *et al.*, 2000) is smaller than MORPH-II, but it also comes with labels for gender and ethnicity. It has many collections of images from around 1,000 subjects, divided by pose, illumination and facial expressions. Most of these collections are small, with less than 300 images. For this reason we only consider the bigger collection, which has around 1,000 frontal face images with regular facial expression, also known by its two letter code as the "Fa" album. Considering the five major ethnic groups, the images are distributed as: African (7.98%), Asian (17.50%), Middle-eastern (5.42%), Caucasian (63.25%) and Hispanic (5.83%), which means FERET is also very unbalanced for ethnicity, with a large predominance of Caucasian subjects. Gender distribution is a bit fairer for African (56.41% females, 43.59% males), Caucasian (41.42% females, 58.58% males) and

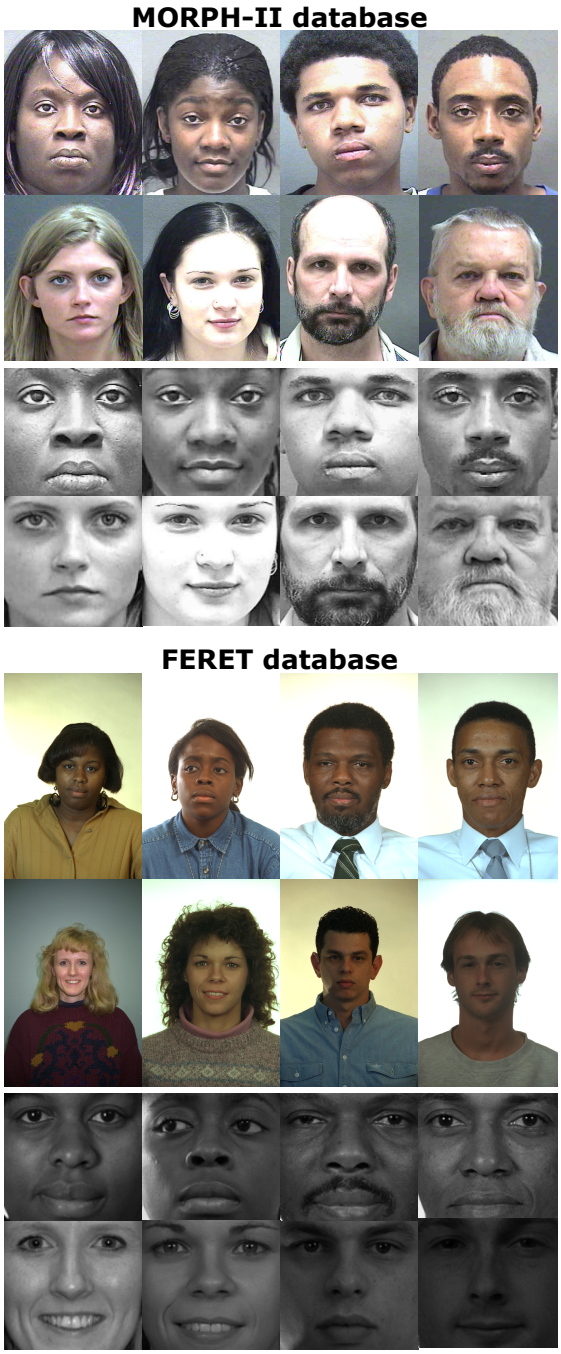


Figure 6.3: *Examples of African and Caucasian subjects from MORPH-II and FERET databases. The grayscale images are obtained after the preprocessing phase described in Section 6.2.1*

Hispanic (52.63% females, 47.37% males). Ethnic groups Asian (33.33% females, 66.67% males) and Middle-eastern (11.32% females, 88.68% males) are the most unbalanced for gender. Figure 6.3 shows some examples of images taken from FERET and MORPH-II datasets.

The AR dataset (Martinez, 1998) is a much smaller but widely used collection of face images. It consists of many subsets featuring different facial expressions and levels of occlusion. It has labels for gender only, and each subset have about 100 images (50 for men and 50 for women). Figure 6.4 shows images of two subjects extracted from the AR database,

Figure 6.4: *Examples of a female and a male subject from the AR database in different conditions: "neutral expression" (left), "all side lights on" (center), "wearing sun glasses" (right).*

each with examples of 3 subsets (from left to the right): "neutral expression", "all side lights on" and "wearing sun glasses".

6.2.1 Face preprocessing

In spite of following an acquisition protocol, sometimes even frontal images have some imperfections as, for instance, being slightly bent. Therefore, in order to mitigate the complexity of the problem and achieve high accuracy, the images must be preprocessed. The major goals of this step are: (i) to eliminate most of the background and areas unrelated to the subject's face (e.g., most of the hair, clothing parts, etc) and (ii) to align a face with respect to its eyes. In this work we used OpenCV's eye and face detectors, which are based on Haar feature-based cascade classifiers (Viola and Jones, 2001) to locate the eyes (to estimate the alignment correction) and face (to eliminate background).

The algorithm to correct these two problems is based on the steps presented in Figure 6.5:

- Apply a pre-trained classifier for frontal face detection (*haarcascade_frontalface_default.xml* or *haarcascade_frontalface_alt.xml*); if one of them fails, we try the other and if both fail the image is discarded from the training set.
- Find both eyes inside the face region, estimate the angle of inclination (θ as in Figure 6.5b) and perform an affine transform to realign the image. Two eye detectors provided by OpenCV have been used: *haarcascade_eye.xml* and *haarcascade_eye_tree_eyeglasses.xml*. If both eyes are located, we evaluate if their regions overlap or if the angle does not exceed a given threshold $t_\theta = \pm 30°$ in any direction.
- Realign the image by the estimated θ (Figure 6.5c) and detect the face in the adjusted image (Figure 6.5d).

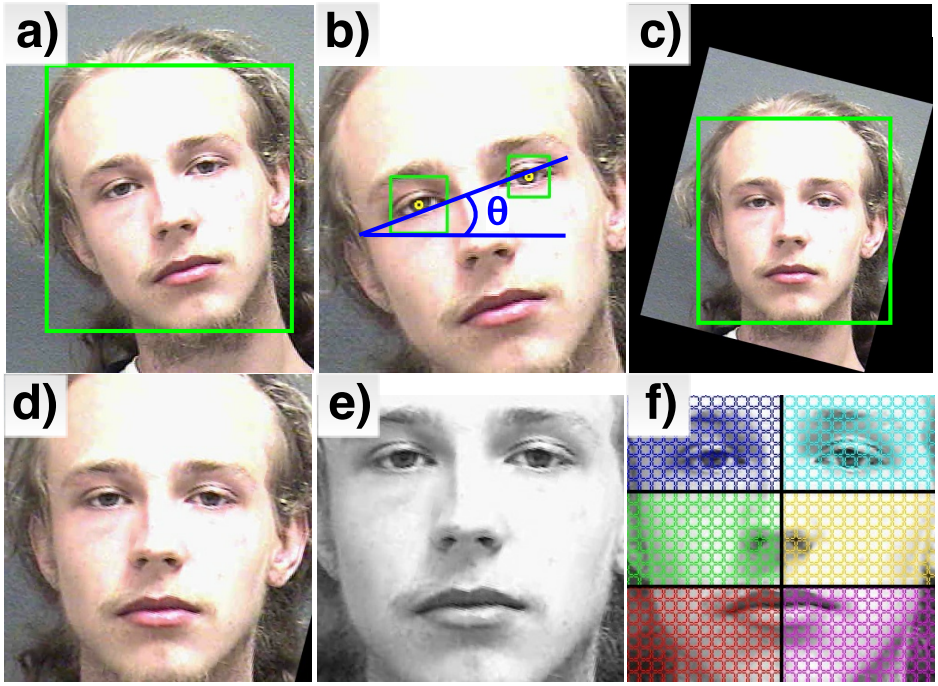


Figure 6.5: *Image preprocessing phases: a) face detection; b) angular evaluation and alignment correction; c) face detection over the aligned image; d) aligned face; e) gray scale conversion, cropping and normalization; f) spatial regions over the face and dense keypoint sampling.*

- Convert to grayscale and crop it according to the following proportions: from left to the right, the first $\frac{1}{8}$ and the last $\frac{1}{8}$ of the domain are cut out, and from top to the bottom, the first $\frac{1}{4}$ and the last $\frac{1}{20}$ of the domain are also excluded (Figure 6.5e). This step eliminate undesired background and allows us to take advantage of the facial symmetries.

Ethnicity	**Female**	**Male**	**Female + Male**
African	4,701 (AF)	28,173 (AM)	32,874 (A)
Caucasian	2,098 (CF)	5,844 (CM)	7,942 (C)
Hispanic	86 (HF)	1,366 (HM)	1,452 (H)
African + Caucasian	6,799	34,017	40,816
African + Caucasian + Hispanic	6,885	35,383	42,268

Table 6.1: *Number of images in MORPH-II used for our study (classes aliases in parentheses).*

Although MORPH-II provides manually annotated eye coordinates, that information is not present in other datasets. FERET has these annotations for some of its images only, for example, and they are certainly absent in real applications. The method described here worked properly for over 75% of the images available in both MORPH-II and FERET. Tables 6.1 and 6.2 present the number of images for which the process was successful for each ethnicity and for each gender.

Ethnicity	Female	Male	Female + Male
African	39	30	69
Caucasian	203	273	476
African + Caucasian	242	303	545

Table 6.2: *Number of images in FERET used for our study.*

6.3 Experiments with real data

6.3.1 Dense SIFT vs dense SURF

An important step in the BoVW model is how to sample the features throughout the images (Fei-Fei and Perona, 2005a). We adopted the dense sampling in our experiments, where features are computed from an evenly sampled grid of patches in each image, since this strategy is known to lead to superior results (Van De Sande *et al.*, 2010). But yet, there is the need to find the ideal patch size.

To accomplish this, we performed a series of experiments varying the vocabulary size, along with the size of the patches in the grid used for dense sampling, to evaluate how those parameters affect ethnicity and gender classification for the SIFT and SURF descriptors. The training data is composed by 3 sets of 600 images each (300 images per class). Since classifying all images from Table 6.1 a hundred times would be computationally expensive and due to the unbalanced amount of images per class in MORPH-II, we made a new balanced test set by selecting 2,000 images per class for the test phase. This test set is composed by 1,000 African females and 1,000 males, plus 1,000 Caucasian females and 1,000 males.

We compare the sampling results for gender and ethnicity with dense SIFT and dense SURF. The selected values for the patch sizes are: 6, 10, 20, 30, 40 and 50, with the following vocabulary sizes: 50, 100, 150, 200, 250 and 300 codewords. That means each graph represent the result of 36 experiments, leading to a total of 144 experiments considering the 4 graphs.

The results are presented in Figure 6.6 as surface plots of the overall accuracies obtained for the test sets, and represent the average accuracy with each of the 3 training sets. They show that SIFT achieves the best results with patches of 10×10 pixels, in contrast to SURF which achieves best results with larger patches. An interesting fact that can be easily observed is that the correct choice of patch size influences the accuracy more than the vocabulary size. Another fact one has to analyze is the trade-off between patch sizes, accuracy and efficiency, because the smaller the patch, the greater the quantity of keypoints to sample in an image. Consequently, more computer resources will be needed to compute their descriptors, encoding to histograms in the pooling phase, and so on.

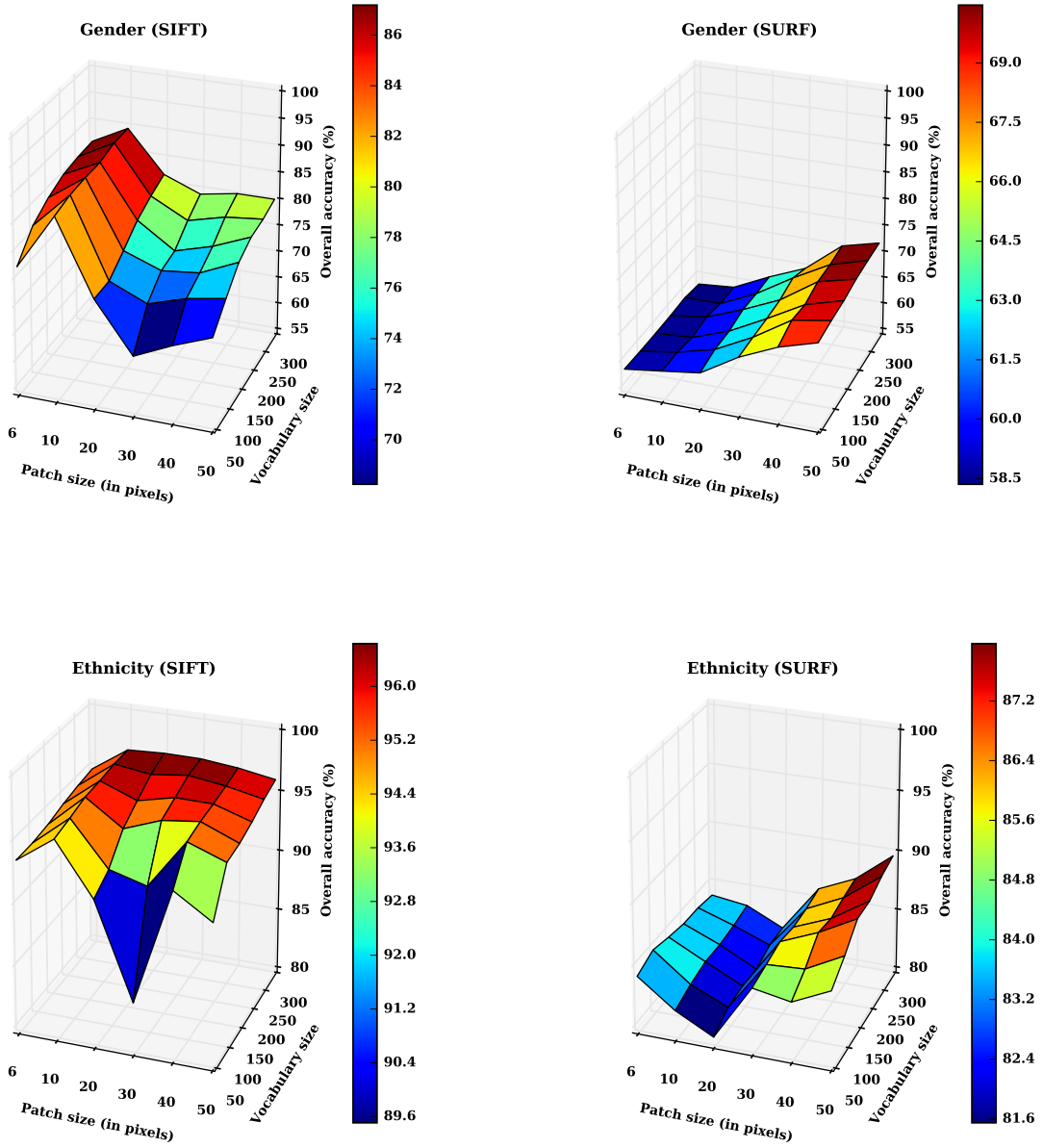


Figure 6.6: *A comparison between the overall accuracy in different configurations of patch and vocabulary sizes for SIFT and extended SURF descriptors.*

6.3.2 Classifier setup

To build the gender and ethnicity classifiers, we adopted a lightweight setup to gain efficiency while maintaining high accuracy rates. Given the results from the experiments in Section 6.3.1 we use a patch size of 10×10 pixels for the dense SIFT sampling. Sections 6.3.3 and 6.3.4 use a vocabulary size of 200 codewords, while Section 6.3.5 improves the quality of the training set by using more images and a bigger visual vocabulary with 400 codewords, which increases accuracies dramatically, but on the other hand it makes the classification a little more expensive.

Since all images belong to the same categorical group (faces) and the ones in MORPH-II are not in the same scale, the images used for training and test are resized to a fixed width and height (300×300 pixels), between the steps described in Figure 6.5c and Figure 6.5e at the preprocessing phase. This ensures that very similar face regions in distinct images and scales could generate spatially close descriptors in the feature space. Furthermore, the original proportions are kept because the face detectors from OpenCV return images with equal width and height.

Our main focus is the MORPH-II dataset, since it is one of the largest labelled available for gender and ethnicity classification, with a reasonable diversity of subjects and image resolutions. Additional experiments are performed with the FERET and AR datasets. In Sections 6.3.3, 6.3.4 and 6.3.5 we compare the accuracy of the BoVW model in the problem of gender and ethnicity classification by replacing the clustering algorithms and pooling strategies with the following:

- **Clustering algorithms**:
  - K-SVD (see Section 5.2)
  - LC-KSVD1 (see Section 5.3.1)
  - LC-KSVD2 (see Section 5.3.2)
  - Our online RDL with $g(u) = \log(\epsilon + u)$ (see Algorithm 6)
  - Our RDL with its standard initialization and $g(u) = \log(\epsilon + u)$ (see Algorithm 4)
  - Our RDL with the undercomplete initialization and $g(u) = \log(\epsilon + u)$ (see Section 5.4.4)
- **Pooling strategies:**
  - Standard BOVW (the standard hard-assignment scheme, see Section 3.4)
  - BOSSA, with $B = 4$ and $\lambda_{\max} = 2$ (see Section 3.6.2)
  - BossaNova, with $B = 4$, $\lambda_{\min} = 2$ and $\lambda_{\max} = 2$ (see Section 3.6.3)
  - PIWAH, with $B = 4$ (see Section 3.6.1)
  - Our pooling proposal, with $B = 4$, $\lambda_{\max} = 2$ and $p = 0.5$ (see Section 4.4)

After the pooling phase, the computed histograms are used in SVMs (Support Vector Machine) for training and then classification. Each SVM use a RBF (Radial Basis Function) kernel (Broomhead and Lowe, 1988), with optimal parameters previously computed using a grid search method. All the results are the average of the overall accuracy of 10 groups (folds) of training data.

6.3.3 Gender classification on the MORPH-II dataset

The training set for these experiments is composed by 810 images (405 for each class) from the MORPH-II database. For this task, we select the two largest ethnic groups (African and Caucasian), totalling 40,816 images (see Table 6.1), i.e., the other 40,006 images are kept for classification.

Table 6.3 shows the results of gender classification using the studied methods. These experiments use a 3×2 spatial regions layout, as shown in Figure 6.5f, due to the good discriminative power achieved while keeping a reasonable histogram size. These experiments reveal great improvement in accuracy when using our RDL methods in the clustering phase of the BoVW model.

Pooling method	**Clustering strategy**					
	K-SVD	**LC-KSVD1**	**LC-KSVD2**	**RDL online**	**RDL standard**	**RDL undercomplete**
Standard BoVW	84.20% (0.799)	85.31% (0.812)	84.91% (1.098)	85.46% (0.515)	85.59% (0.703)	**86.07%** (0.490)
BOSSA	85.45% (0.614)	85.66% (0.711)	85.28% (1.064)	86.63% (0.508)	86.75% (0.537)	**87.20%** (0.489)
BossaNova	85.60% (1.010)	85.58% (0.828)	85.75% (1.305)	86.62% (0.546)	86.83% (0.219)	**87.29%** (0.509)
PIWAH	84.34% (1.172)	84.96% (0.675)	85.19% (1.022)	85.40% (0.324)	85.59% (0.892)	**85.64%** (0.424)

Table 6.3: *Average classification accuracies and standard deviations for gender on the MORPH-II dataset using the standard subregions approach.*

In another set of experiments we repeated all the scenarios, but instead of using the 3×2 region layout, we replaced them by 6 CoIs. The results in Table 6.4 demonstrate the discriminative power of our centers of incidence approach. They also show that our pooling strategy achieves the best results in all clustering schemes, since it needs the CoIs to work.

Pooling method	**Clustering strategy**					
	K-SVD	**LC-KSVD1**	**LC-KSVD2**	**RDL online**	**RDL standard**	**RDL undercomplete**
Standard BoVW	84.47% (0.847)	85.57% (1.004)	85.36% (1.084)	86.15% (0.667)	86.37% (0.608)	**86.87%** (0.668)
BOSSA	85.51% (0.903)	85.86% (1.016)	85.68% (1.017)	86.63% (0.508)	86.88% (0.599)	**87.44%** (0.582)
BossaNova	85.71% (0.979)	85.95% (0.489)	85.91% (1.335)	86.76% (0.807)	86.91% (0.680)	**87.66%** (0.491)
PIWAH	84.54% (0.834)	85.25% (1.077)	85.30% (0.879)	85.41% (0.270)	85.51% (0.863)	**85.76%** (0.647)
Ours	87.27% (0.488)	87.10% (0.784)	87.00% (0.819)	88.13% (0.524)	88.19% (0.493)	**88.36%** (0.376)

Table 6.4: *Average classification accuracies and standard deviations for gender on the MORPH-II dataset using the CoI approach.*

6.3.4 Ethnicity classification on the MORPH-II dataset

Following the gender classification, we performed a similar set of experiments with ethnicity. This time, we use two scenarios: the first with 2 classes (African and Caucasian) and the other with 3 classes (African, Caucasian and Hispanic). The former uses 810 images and

the last 1,215 images for training (405 per class) for each of the 10 folds. The remaining images (40,006 for 2 classes and 41,053 for 3 classes) are used in the classification step.

Table 6.5 displays the results with the 3×2 region layout. Again, note that our RDL approach achieves the best results for all pooling strategies, while remarkably our classifiers obtained higher accuracies for ethnicity.

Pooling method	Clustering strategy					
	K-SVD	**LC-KSVD1**	**LC-KSVD2**	**RDL online**	**RDL standard**	**RDL undercomplete**
Standard BoVW	96.22% (0.160)	96.27% (0.312)	96.44% (0.282)	96.72% (0.120)	96.96% (0.357)	**96.88%** (0.117)
BOSSA	96.81% (0.077)	96.39% (0.364)	96.49% (0.270)	96.66% (0.105)	96.99% (0.256)	**97.03%** (0.196)
BossaNova	96.88% (0.107)	96.69% (0.273)	96.73% (0.246)	96.99% (0.099)	97.08% (0.210)	**97.26%** (0.103)
PIWAH	96.34% (0.152)	96.39% (0.434)	96.46% (0.262)	96.95% (0.084)	97.10% (0.214)	**97.21%** (0.072)

Table 6.5: *Average classification accuracies and standard deviations for ethnicity (with 2 classes) on the MORPH-II dataset using the standard subregions approach.*

Table 6.6 shows the accuracies for the same set of experiments, but again replacing the 3×2 region layout by 6 CoIs. The last rows contains the results for our pooling strategy, which also got the best results. One more time, these results show the discriminative power of our centers of incidence approach over the standard subregion division.

Pooling method	Clustering strategy					
	K-SVD	**LC-KSVD1**	**LC-KSVD2**	**RDL online**	**RDL standard**	**RDL undercomplete**
Standard BoVW	96.72% (0.148)	96.83% (0.249)	96.89% (0.236)	97.07% (0.144)	97.18% (0.269)	**97.33%** (0.109)
BOSSA	96.86% (0.157)	96.93% (0.444)	96.98% (0.215)	96.98% (0.102)	97.03% (0.236)	**97.14%** (0.196)
BossaNova	96.99% (0.102)	97.15% (0.258)	97.13% (0.215)	97.23% (0.121)	97.47% (0.192)	**97.60%** (0.078)
PIWAH	96.60% (0.221)	96.84% (0.341)	96.87% (0.217)	97.09% (0.099)	97.14% (0.196)	**97.33%** (0.071)
Ours	97.71% (0.090)	97.73% (0.181)	97.79% (0.198)	98.00% (0.085)	98.07% (0.073)	**98.13%** (0.067)

Table 6.6: *Average classification accuracies and standard deviations for ethnicity (with 2 classes) on the MORPH-II dataset using the CoI approach.*

Finally, Tables 6.7 and 6.8 present the results for both subregions and CoIs approaches, respectively. The performance of the classifiers is affected by the new Hispanic class, but our pooling method was able to achieve over 91% of accuracy (see Table 6.8). This phenomenon was already noted in the past by Guo and Mu (2010) when using the MORPH-II dataset, because it is harder to distinguish between hispanics and caucasians. In fact we got this same problem, with a lot of caucasian subjects being classified as hispanics and vice-versa, rather than being mistaken by african subjects.

Pooling method	Clustering strategy					
	K-SVD	**LC-KSVD1**	**LC-KSVD2**	**RDL online**	**RDL standard**	**RDL undercomplete**
Standard BoVW	88.53% (0.235)	88.64% (0.589)	88.55% (0.594)	89.03% (0.298)	89.11% (0.374)	**89.38%** (0.192)
BOSSA	89.02% (0.193)	89.19% (0.514)	89.11% (0.543)	90.06% (0.215)	90.27% (0.262)	**90.45%** (0.173)
BossaNova	89.07% (0.311)	89.53% (0.760)	89.65% (0.789)	90.14% (0.241)	90.41% (0.291)	**90.57%** (0.272)
PIWAH	88.61% (0.270)	88.72% (0.692)	88.86% (0.636)	89.08% (0.226)	89.22% (0.310)	**89.29%** (0.167)

Table 6.7: *Average classification accuracies and standard deviations for ethnicity (with 3 classes) on the MORPH-II dataset using the standard subregions approach.*

Pooling method	Clustering strategy					
	K-SVD	**LC-KSVD1**	**LC-KSVD2**	**RDL online**	**RDL standard**	**RDL undercomplete**
Standard BoVW	89.30% (0.306)	89.59% (0.477)	89.68% (0.400)	89.85% (0.194)	89.94% (0.304)	**89.98%** (0.125)
BOSSA	89.87% (0.268)	90.02% (0.395)	89.96% (0.400)	90.40% (0.142)	90.45% (0.305)	**90.56%** (0.104)
BossaNova	89.91% (0.406)	90.15% (0.692)	90.18% (0.779)	90.52% (0.275)	90.67% (0.357)	**90.73%** (0.262)
PIWAH	88.65% (0.316)	89.32% (0.532)	89.38% (0.499)	89.42% (0.245)	89.50% (0.399)	**89.66%** (0.116)
Ours	90.27% (0.237)	90.37% (0.454)	90.49% (0.459)	91.21% (0.197)	91.29% (0.149)	**91.42%** (0.144)

Table 6.8: *Average classification accuracies and standard deviations for ethnicity (with 3 classes) on the MORPH-II dataset using the CoI approach.*

6.3.5 Improvements and applying our classifier to other datasets

In this section we improve our classifier by increasing the number of training images to 1,998 (999 per class), for both gender and ethnicity[1] classification tasks. As in the previous experiments, all the images in the training set are taken from the MORPH-II dataset. The number of visual words is also increased to 400, which still offers a reasonable trade-off between efficiency and accuracy. Since our RDL techniques outperformed the others, we only use the batch versions (with standard and undercomplete dictionary initializations) for simplicity

In the first part, we apply the improved classifiers to the MORPH-II database, so we can assess the enhancement in accuracies, compared to the ones with smaller training sets and vocabularies from Sections 6.3.3 and 6.3.4. Then, we aim to prove empirically that our approach is robust by performing some domain adaptation experiments by classifying the images from FERET (ethnicity and gender) and the AR (gender only) databases.

Tables 6.9 and 6.10 presents the results for both problems in the MORPH-II database. Note the sharp increase in accuracy for the gender classification problem (up to 95%, compared with the previous result, a bit more than 88%). Improvements can also be noted for ethnicity, but since the previous setup have already lead to higher accuracies, it seems to approach to the limits of our BoVW scheme.

[1]This time we only focus on the 2 classes problem for ethnicity classification, since there would be few images left for hispanics.

	Pooling method	Clustering strategy RDL standard	RDL undercomplete
Subregions	Standard BoVW	92.19% (0.058)	**93.46%** (0.272)
	BOSSA	93.28% (0.073)	**94.27%** (0.212)
	BossaNova	93.19% (0.179)	**94.54%** (0.260)
	PIWAH	92.87% (0.116)	**93.49%** (0.393)
CoIs	Standard BoVW	92.57% (0.105)	**93.99%** (0.125)
	BOSSA	93.64% (0.045)	**94.66%** (0.129)
	BossaNova	93.63% (0.110)	**94.68%** (0.211)
	PIWAH	93.00% (0.096)	**93.63%** (0.277)
	Ours	94.95% (0.069)	**95.32%** (0.111)

Table 6.9: *Average classification accuracies and standard deviations for gender on the MORPH-II dataset.*

	Pooling method	Clustering strategy RDL standard	RDL undercomplete
Subregions	Standard BoVW	97.16% (0.064)	**97.57%** (0.106)
	BOSSA	97.34% (0.045)	**97.51%** (0.095)
	BossaNova	**97.55%** (0.102)	97.54% (0.095)
	PIWAH	97.03% (0.050)	**97.36%** (0.143)
CoIs	Standard BoVW	97.49% (0.060)	**97.87%** (0.095)
	BOSSA	97.81% (0.049)	**97.95%** (0.085)
	BossaNova	97.81% (0.090)	**97.98%** (0.085)
	PIWAH	97.40% (0.059)	**97.73%** (0.067)
	Ours	98.14% (0.062)	**98.17%** (0.043)

Table 6.10: *Average classification accuracies and standard deviations for ethnicity (with 2 classes) on the MORPH-II dataset.*

Figure 6.7: *Examples of misclassified gender in the MORPH-II dataset: a) Females classified as Males; b) Males classified as Females.*

Figures 6.7 and 6.8 show examples of misclassified images in the experiments from Tables 6.9 and 6.10, respectively. Note that some images are quite challenging to describe due to specific traits in the hairstyle, facial expression or even color skin, although these are subjective matters.

Figure 6.8: *Examples of misclassified ethnicity in the MORPH-II dataset: a) Africans classified as Caucasians; b) Caucasians classified as Africans.*

Results with the FERET dataset are shown in Tables 6.11 and 6.12. Even with all the training images being from the MORPH-II dataset, the setup using the proposed RDL method along with our pooling strategy is able to achieve up to 91% for gender and up to 81% for ethnicity classification. Thus, we can have an idea of how powerful our model is, because different datasets have distinct illumination and scenario conditions, and even the camera quality and resolution could affect results. We believe that these results could be further improved if more images were used on the training step, especially if they were from different datasets, which is not the case here.

	Pooling method	**Clustering strategy**	
		RDL standard	**RDL undercomplete**
Subregions	Standard BoVW	86.55% (0.734)	**87.39%** (0.614)
	BOSSA	87.72% (0.627)	**88.95%** (0.567)
	BossaNova	87.85% (1.401)	**89.18%** (0.681)
	PIWAH	87.18% (1.228)	**87.83%** (0.841)
ColS	Standard BoVW	86.70% (1.314)	**87.42%** (0.406)
	BOSSA	87.88% (0.668)	**89.40%** (0.642)
	BossaNova	87.99% (1.744)	**89.44%** (1.011)
	PIWAH	87.21% (0.609)	**89.21%** (0.505)
	Ours	88.89% (1.292)	**90.25%** (0.975)

Table 6.11: *Average classification accuracies and standard deviations for gender on the FERET dataset.*

Finally, Tables 6.13, 6.14 and 6.15 show the results of gender classification in the AR dataset with the "neutral expression", "all side lights on" and "wearing sun glasses" subsets.

	Pooling method	Clustering strategy: RDL standard	Clustering strategy: RDL undercomplete
Subregions	Standard BoVW	77.32% (0.865)	**78.42%** (2.866)
	BOSSA	80.26% (1.568)	**80.42%** (0.513)
	BossaNova	**80.71%** (1.527)	80.59% (1.155)
	PIWAH	79.06% (0.784)	**79.51%** (2.619)
CoIs	Standard BoVW	77.93% (0.787)	**80.42%** (2.116)
	BOSSA	**80.70%** (0.851)	80.54% (1.403)
	BossaNova	80.85% (3.195)	**80.98%** (2.234)
	PIWAH	**80.27%** (0.774)	80.08% (1.979)
	Ours	81.17% (2.222)	**81.36%** (1.108)

Table 6.12: *Average classification accuracies and standard deviations for ethnicity (with 2 classes) on the FERET dataset.*

The neutral expression set is very similar to the ones in MORPH-II and FERET. On the other hand, the set with great variation in light conditions poses a challenge to our classifiers, as well as the set with sunglasses, which is one of the most common types of face occlusion when dealing with such a classification problem.

	Pooling method	Clustering strategy: RDL standard	Clustering strategy: RDL undercomplete
Subregions	Standard BoVW	70.00% (1.844)	**72.50%** (6.859)
	BOSSA	71.70% (1.418)	**78.50%** (5.696)
	BossaNova	74.80% (2.960)	**78.60%** (4.884)
	PIWAH	70.70% (1.900)	**77.20%** (6.258)
CoIs	Standard BoVW	73.80% (3.124)	**78.40%** (5.276)
	BOSSA	79.40% (2.289)	**82.70%** (3.035)
	BossaNova	79.90% (5.467)	**82.90%** (4.721)
	PIWAH	74.50% (2.110)	**80.90%** (2.700)
	Ours	82.40% (2.200)	**85.20%** (1.077)

Table 6.13: *Average classification accuracies and standard deviations for gender on the AR ("neutral expression") dataset.*

One important detail that prevented the AR dataset to perform better is the fact that the faces there are already cropped, as seen in Figure 6.4. Note that the ears, part of the hair and part of the chin are missing, what makes it difficult to apply the face processing step described in Section 6.2.1. Since all our training examples are from MORPH-II and they all passed by those steps, it means that the sampling step of our BoVW scheme will be far from ideal for the AR images.

Nevertheless our results were reasonably good when compared with the work of Borgi *et al.* (2014), which used half the images from AR for training and half for classification, so things are tougher for us since we use a complete different dataset for training and the full AR set (double the images) for test. More recently, the work of Juefei-Xu *et al.* (2016) uses a much more complex classifier using deep learning techniques to achieve better results, but their training set consists of about 89,000 images extracted from 5 distinct face databases.

	Pooling method	**Clustering strategy**	
		RDL standard	**RDL undercomplete**
Subregions	Standard BoVW	65.20% (2.600)	**70.10%** (5.186)
	BOSSA	69.70% (2.410)	**70.10%** (5.069)
	BossaNova	69.80% (3.458)	**72.20%** (8.588)
	PIWAH	71.10% (2.914)	**75.00%** (5.762)
CoIs	Standard BoVW	76.10% (7.892)	**79.40%** (1.685)
	BOSSA	76.30% (7.721)	**81.00%** (2.966)
	BossaNova	76.70% (1.952)	**81.50%** (5.886)
	PIWAH	76.30% (5.951)	**80.70%** (1.418)
	Ours	80.00% (3.873)	**84.20%** (2.182)

Table 6.14: *Average classification accuracies and standard deviations for gender on the AR ("all side lights on") dataset.*

	Pooling method	**Clustering strategy**	
		RDL standard	**RDL undercomplete**
Subregions	Standard BoVW	69.30% (5.883)	**71.10%** (2.468)
	BOSSA	69.60% (7.046)	**71.80%** (1.720)
	BossaNova	69.90% (3.400)	**71.40%** (2.615)
	PIWAH	68.40% (3.693)	**71.70%** (2.002)
CoIs	Standard BoVW	73.00% (2.530)	**73.60%** (4.271)
	BOSSA	73.70% (5.001)	**76.00%** (2.022)
	BossaNova	74.00% (4.123)	**76.30%** (2.973)
	PIWAH	69.30% (1.900)	**72.40%** (4.005)
	Ours	76.40% (5.269)	**76.80%** (2.088)

Table 6.15: *Average classification accuracies and standard deviations for gender on the AR ("wearing sun glasses") dataset.*

6.3.6 CoIs configuration: Standard vs Fixed vs Random positions

This section aims to show empirically the advantages of using clustering techniques to compute the Centers of Incidence (CoIs), as presented in Section 4.2. To achieve this, we perform experiments with three different ways to set the CoIs spatial positions:

- **Standard positions**, set using the K-Means algorithm (as defined in Section 4.2).
- **Fixed positions**, by setting the positions of the CoIs in the centers of each of the 3×2 subregions.
- **Random positions**, by randomly selecting the position of the CoIs in the descriptors' space.

Figure 6.9 shows some examples of each of these three approaches, where the CoIs are represented by the white circles. The images represents a 2D histogram of the spatial distribution of descriptors for each codeword in the training set.

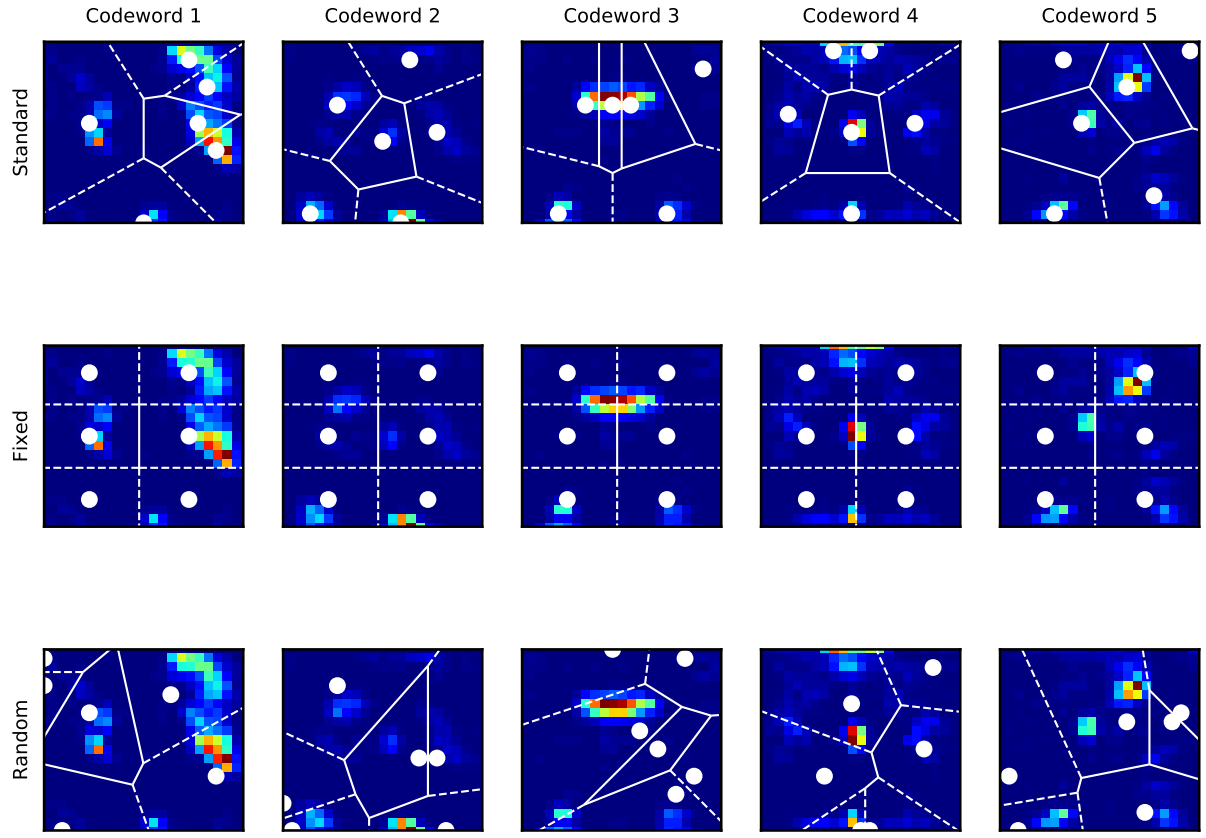


Figure 6.9: *Examples on five codewords in different configurations of Centers of Incidence (white points) and the Voronoi regions they define: Standard (with clustering, as defined in Section 4.2), Fixed (equivalent to the 3×2 subregion configuration), and Randomly distributed centers.*

The experiments are accomplished using the MORPH-II dataset and as in our previous setups they are made for both gender and ethnicity classification. For the clustering step

we used our RDL undercomplete approach with 200 visual words, and for simplicity we compare the results using two pooling methods: the standard BoVW (using CoIs instead of subregions, as defined in Section 4.3) and our pooling method. The results are presented in Tables 6.16 and 6.17, respectively.

Type of CoI configuration	**Pooling method**	
	Standard BoVW with CoIs	**Ours**
Standard (with clustering)	86.87% (0.668)	**88.36%** (0.376)
Fixed positions	86.07% (0.488)	**87.68%** (0.410)
Randomly distributed	84.99% (0.908)	**87.52%** (0.385)

Table 6.16: *Average classification accuracies and standard deviations for gender on the MORPH-II dataset using three approaches for CoIs distribution.*

For both cases, the best results are achieved by using the clustering (standard) approach, which corroborates with the theory discussed by us earlier in Sections 4.2 and 4.3. We believe that this happens because this approach makes the CoIs more representative by making the final BoW histogram (i.e., the image representation) less sparse. Another interesting result happens with the fixed positions approach, which gives similar results when using the standard BoVW pooling with 3×2 subregions (compare with Table 6.3 for gender and Table 6.5 for ethnicity). This supports our claim that the CoIs approach can be seen as a generalization of the subregion division, as discussed in Section 4.3. Finally, those experiments clearly show that the randomly distributed CoI positions is the worst approach.

Type of CoI configuration	**Pooling method**	
	Standard BoVW with CoIs	**Ours**
Standard (with clustering)	97.33% (0.109)	**98.13%** (0.067)
Fixed positions	96.88% (0.118)	**98.01%** (0.074)
Randomly distributed	96.62% (0.308)	**97.96%** (0.072)

Table 6.17: *Average classification accuracies and standard deviations for ethnicity on the MORPH-II dataset using three approaches for CoIs distribution.*

Chapter 7

Conclusion

In this chapter we review the activities and key contributions achieved in this work, as well as suggestions for future directions in this research.

This text presents the theory behind the proposed classifiers, literature review on known and state-of-the-art methods, as well as information about well known datasets with labels for gender and ethnicity. The central objective of this work was to employ the bag-of-visual-words model, which is simple and computationally efficient, to perform gender and ethnicity classification based solely on face images. This type of model requires a relatively small set of images for training, which makes sense for this kind of categorization, where the labelled datasets have a few thousand images in most cases.

The BoVW model can be divided in 4 phases: (i) sampling, (ii) clustering, (iii) pooling and (iv) classifying. Our goal was to propose new methods for phases (ii) and (iii), which can greatly improve accuracy. For (ii) we proposed a robust dictionary learning algorithm that can mitigate the influence of outliers in the result of the unsupervisied learning step. Furthermore, we proposed an algorithm to better initialize the dictionary and improve the detection of those outliers in the input data. For (iii) we proposed a new way to subdivide an image in subregions for the pooling step in BoVW models. Finally, we proposed a pooling algorithm that can take advantage of this new subdivision and get more discriminative information based on angles between image features and key positions on how the features are distributed over the images.

7.1 Suggestions for future works

- Experiments with images taken from unconstrained environments, preferably if they are taken from high quality and high resolution surveillance cameras, once this kind of classification would be of great value for law enforcement applications, for example.

- Experiments with age estimation, since MORPH-II, FERET and AR subjects are concentrated in a very specific and close range of ages it is impractical to perform such experiments without adding new images from external sources. That could cause the

final set to be biased, since the environmental conditions where and how the images were taken can influence badly on the classifier.

- Use of other feature detection algorithms, specially methods that encode color information.

- Experiments with big datasets with more than 2 or 3 ethnic classes. Although it is difficult to construct and label this kind of data, it would be interesting to analyze how powerful a model can be and how it discriminates some combinations of "visually similar" ethnic groups. For example: Chinese vs Japanese, or Caucasian vs Hispanic.

Appendix A

Data from Examples

A.1 SIFT descriptors and centroids from Figure 3.1

Points	SIFT descriptors
P_{1a}	⟨ 85, 26, 11, 18, 2, 1, 2, 28, 11, 6, 96, 54, 4, 3, 14, 10, 4, 27, 121, 11, 0, 1, 14, 10, 10, 24, 24, 3, 2, 1, 1, 3, 111, 12, 34, 94, 26, 13, 16, 51, 10, 12, 121, 121, 23, 40, 102, 58, 42, 91, 121, 33, 4, 19, 82, 72, 79, 42, 16, 3, 12, 6, 6, 53, 66, 4, 5, 24, 28, 48, 68, 43, 12, 16, 37, 55, 19, 73, 121, 76, 24, 17, 13, 20, 12, 18, 121, 121, 31, 9, 4, 6, 10, 3, 11, 51, 43, 16, 9, 1, 3, 3, 13, 11, 4, 3, 6, 3, 4, 4, 27, 9, 7, 5, 40, 46, 42, 11, 13, 16, 10, 22, 98, 40, 13, 6, 5, 10 ⟩
P_{2a}	⟨ 8, 9, 22, 44, 19, 55, 115, 33, 12, 14, 25, 24, 10, 28, 115, 109, 42, 17, 3, 8, 7, 4, 44, 97, 6, 3, 3, 9, 32, 38, 6, 3, 6, 6, 8, 2, 1, 8, 40, 8, 4, 2, 19, 33, 33, 9, 35, 23, 19, 25, 115, 66, 31, 9, 12, 25, 14, 51, 104, 47, 19, 14, 3, 3, 21, 14, 10, 3, 2, 1, 2, 7, 5, 4, 15, 34, 78, 29, 11, 4, 23, 23, 85, 59, 94, 83, 69, 34, 37, 50, 61, 30, 26, 37, 77, 45, 17, 17, 10, 14, 8, 2, 2, 7, 7, 12, 71, 101, 27, 13, 53, 27, 4, 29, 115, 100, 17, 39, 115, 38, 7, 65, 115, 24, 3, 30, 115, 27 ⟩
P_{3a}	⟨ 2, 1, 10, 20, 44, 16, 5, 2, 16, 18, 76, 45, 55, 41, 32, 22, 33, 57, 68, 33, 21, 26, 41, 27, 54, 42, 13, 2, 6, 7, 11, 23, 5, 8, 46, 68, 27, 14, 30, 15, 7, 27, 116, 79, 28, 56, 116, 36, 17, 69, 116, 26, 8, 38, 116, 51, 27, 64, 45, 7, 6, 15, 27, 26, 7, 7, 45, 75, 21, 11, 75, 43, 3, 22, 116, 84, 26, 27, 116, 34, 13, 81, 116, 16, 3, 42, 116, 20, 21, 74, 46, 7, 11, 52, 60, 12, 16, 20, 18, 15, 5, 4, 17, 20, 4, 12, 19, 21, 14, 20, 99, 24, 8, 18, 25, 13, 5, 28, 95, 20, 7, 16, 26, 26, 25, 27, 20, 5 ⟩
P_{1b}	⟨ 115, 7, 19, 17, 7, 1, 8, 62, 54, 17, 115, 32, 1, 7, 39, 45, 9, 64, 115, 5, 1, 9, 16, 7, 9, 38, 22, 7, 2, 0, 0, 1, 115, 8, 21, 27, 16, 7, 18, 82, 18, 16, 115, 66, 16, 38, 92, 39, 45, 73, 111, 14, 9, 39, 48, 34, 66, 38, 10, 5, 14, 9, 8, 40, 113, 3, 6, 13, 9, 29, 49, 94, 24, 18, 57, 32, 28, 87, 115, 57, 47, 28, 32, 11, 17, 35, 115, 115, 57, 7, 1, 2, 8, 8, 16, 47, 89, 22, 5, 2, 3, 2, 9, 29, 14, 13, 10, 9, 4, 5, 36, 15, 16, 16, 22, 69, 28, 5, 20, 20, 22, 39, 62, 31, 10, 7, 7, 14 ⟩
P_{2b}	⟨ 16, 7, 17, 14, 11, 54, 104, 28, 21, 14, 24, 13, 17, 31, 113, 92, 64, 10, 3, 7, 7, 9, 38, 63, 7, 3, 3, 3, 16, 32, 14, 4, 19, 24, 13, 5, 2, 2, 17, 10, 14, 12, 20, 85, 38, 7, 19, 18, 42, 87, 113, 91, 34, 21, 21, 35, 24, 75, 113, 45, 20, 22, 18, 21, 16, 41, 24, 7, 1, 0, 0, 2, 11, 17, 28, 50, 48, 35, 16, 16, 25, 47, 65, 53, 44, 74, 91, 81, 33, 45, 60, 29, 11, 43, 99, 103, 30, 19, 8, 5, 4, 2, 0, 2, 6, 18, 65, 74, 17, 8, 26, 16, 6, 22, 113, 52, 4, 14, 108, 34, 7, 88, 113, 12, 1, 23, 90, 19 ⟩
P_{3b}	⟨ 7, 9, 15, 41, 29, 13, 5, 6, 26, 60, 65, 74, 49, 54, 53, 49, 38, 71, 99, 54, 19, 49, 82, 80, 27, 46, 15, 5, 7, 10, 15, 35, 9, 20, 40, 40, 16, 9, 7, 10, 7, 16, 120, 46, 13, 27, 76, 44, 16, 81, 120, 13, 2, 24, 88, 66, 6, 34, 65, 24, 13, 10, 14, 18, 5, 7, 34, 57, 27, 18, 50, 27, 10, 20, 120, 53, 7, 21, 120, 51, 14, 102, 120, 13, 3, 64, 120, 20, 20, 63, 67, 10, 5, 17, 107, 27, 5, 4, 6, 4, 7, 9, 16, 13, 7, 11, 14, 6, 7, 15, 48, 21, 10, 34, 24, 3, 1, 19, 80, 14, 9, 13, 11, 11, 6, 8, 17, 11 ⟩

Table A.1: *SIFT descriptors from the points in Figures 3.1a and 3.1b.*

Points	SIFT descriptors
P_{1c}	⟨ 30, 2, 8, 4, 1, 2, 13, 107, 20, 24, 48, 7, 1, 18, 55, 75, 4, 24, 44, 5, 7, 55, 112, 14, 4, 6, 10, 6, 12, 60, 92, 16, 36, 3, 18, 21, 10, 39, 50, 90, 13, 44, 120, 65, 10, 94, 120, 38, 20, 105, 120, 21, 19, 62, 103, 51, 35, 47, 44, 12, 14, 18, 32, 36, 40, 14, 9, 7, 5, 35, 41, 25, 17, 19, 50, 29, 10, 101, 120, 45, 21, 16, 29, 29, 37, 75, 120, 84, 78, 12, 2, 3, 35, 25, 48, 68, 39, 38, 11, 2, 1, 1, 1, 5, 15, 21, 20, 4, 1, 4, 13, 12, 9, 12, 24, 32, 29, 15, 24, 14, 45, 25, 30, 12, 23, 10, 8, 12 ⟩
P_{2c}	⟨ 9, 11, 18, 8, 3, 42, 71, 14, 9, 9, 16, 20, 21, 47, 114, 46, 61, 8, 2, 5, 25, 17, 50, 53, 20, 4, 2, 7, 59, 51, 35, 19, 16, 20, 16, 2, 1, 1, 4, 9, 11, 15, 31, 36, 35, 18, 17, 16, 52, 57, 76, 27, 35, 22, 18, 18, 27, 18, 55, 39, 35, 17, 5, 10, 20, 40, 30, 20, 11, 2, 2, 6, 19, 30, 114, 83, 41, 35, 22, 23, 17, 71, 114, 68, 40, 60, 77, 29, 36, 53, 110, 62, 13, 29, 58, 34, 30, 74, 31, 40, 32, 4, 13, 21, 20, 26, 86, 47, 24, 18, 114, 66, 6, 40, 114, 29, 13, 33, 114, 37, 10, 19, 114, 35, 7, 33, 114, 30 ⟩
P_{3c}	⟨ 7, 10, 23, 21, 25, 16, 7, 9, 24, 55, 79, 26, 26, 23, 17, 9, 29, 25, 74, 51, 19, 12, 7, 14, 22, 11, 19, 8, 6, 6, 4, 10, 13, 23, 114, 67, 22, 18, 40, 26, 13, 47, 114, 52, 22, 46, 114, 37, 27, 53, 114, 55, 14, 45, 114, 40, 36, 79, 75, 26, 12, 18, 29, 12, 20, 26, 40, 21, 17, 16, 87, 50, 4, 38, 114, 27, 22, 38, 114, 30, 21, 20, 114, 42, 9, 31, 114, 45, 35, 27, 33, 47, 49, 37, 60, 14, 12, 89, 44, 3, 2, 5, 10, 4, 3, 30, 56, 19, 11, 21, 53, 13, 6, 12, 60, 52, 8, 6, 48, 24, 18, 7, 23, 94, 31, 3, 5, 9 ⟩
P_{1d}	⟨ 101, 21, 19, 6, 1, 2, 15, 67, 12, 8, 71, 19, 3, 3, 36, 40, 5, 23, 62, 7, 2, 2, 8, 6, 1, 7, 25, 5, 1, 0, 0, 0, 118, 26, 59, 38, 10, 15, 29, 40, 22, 26, 120, 117, 27, 56, 85, 52, 55, 95, 120, 38, 16, 55, 45, 61, 24, 32, 55, 10, 7, 16, 18, 46, 95, 15, 12, 17, 16, 35, 61, 41, 32, 34, 64, 38, 38, 120, 120, 80, 64, 26, 50, 25, 20, 79, 120, 120, 22, 3, 1, 1, 6, 16, 22, 45, 79, 55, 10, 5, 12, 9, 4, 11, 22, 22, 15, 10, 9, 13, 30, 22, 11, 10, 13, 28, 13, 5, 18, 28, 10, 8, 8, 8, 3, 2, 2, 8 ⟩
P_{2d}	⟨ 19, 22, 15, 9, 12, 35, 61, 22, 16, 13, 24, 19, 12, 23, 76, 69, 16, 8, 6, 10, 3, 5, 12, 22, 4, 7, 5, 5, 16, 28, 12, 4, 47, 39, 16, 8, 5, 4, 8, 22, 16, 15, 25, 39, 37, 13, 16, 35, 20, 37, 81, 30, 19, 12, 13, 33, 22, 26, 66, 21, 22, 30, 6, 7, 52, 53, 29, 8, 2, 2, 2, 13, 21, 27, 96, 27, 41, 32, 16, 11, 5, 47, 125, 43, 70, 77, 53, 15, 43, 47, 125, 30, 7, 14, 43, 79, 33, 60, 36, 27, 12, 4, 12, 8, 19, 56, 125, 54, 14, 35, 125, 35, 8, 62, 125, 48, 12, 67, 125, 27, 7, 60, 125, 37, 9, 30, 125, 37 ⟩
P_{3d}	⟨ 11, 8, 10, 7, 17, 8, 5, 9, 4, 27, 71, 18, 38, 29, 13, 11, 22, 24, 96, 25, 8, 12, 11, 25, 28, 11, 22, 10, 9, 12, 7, 18, 16, 33, 117, 37, 10, 11, 43, 15, 7, 37, 118, 51, 32, 69, 118, 17, 16, 72, 118, 31, 11, 32, 118, 69, 18, 49, 90, 23, 17, 16, 31, 18, 20, 41, 48, 17, 12, 36, 58, 26, 6, 57, 118, 25, 9, 61, 118, 16, 10, 34, 118, 44, 6, 22, 118, 44, 19, 29, 51, 35, 25, 37, 48, 21, 12, 79, 27, 1, 3, 15, 8, 7, 4, 31, 56, 9, 8, 32, 69, 12, 4, 13, 57, 33, 4, 5, 68, 35, 3, 1, 26, 97, 22, 14, 11, 12 ⟩

Table A.2: *SIFT descriptors from the points in Figures 3.1c and 3.1d.*

Centroids	Centroid coordinates
c_1	⟨ 82.75, 14.0, 14.25, 11.25, 2.75, 1.5, 9.5, 66.0, 24.25, 13.75, 82.5, 28.0, 2.25, 7.75, 36.0, 42.5, 5.5, 34.5, 85.5, 7.0, 2.5, 16.75, 37.5, 9.25, 6.0, 18.75, 20.25, 5.25, 4.25, 15.25, 23.25, 5.0, 95.0, 12.25, 33.0, 45.0, 15.5, 18.5, 28.25, 65.75, 15.75, 24.5, 119.0, 92.25, 19.0, 57.0, 99.75, 46.75, 40.5, 91.0, 118.0, 26.5, 12.0, 43.75, 69.5, 54.5, 51.0, 39.75, 31.25, 7.5, 11.75, 12.25, 16.0, 43.75, 78.5, 9.0, 8.0, 15.25, 14.5, 36.75, 54.75, 50.75, 21.25, 21.75, 52.0, 38.5, 23.75, 95.25, 119.0, 64.5, 39.0, 21.75, 31.0, 21.25, 21.5, 51.75, 119.0, 110.0, 47.0, 7.75, 2.0, 3.0, 14.75, 13.0, 24.25, 52.75, 62.5, 32.75, 8.75, 2.5, 4.75, 3.75, 6.75, 14.0, 13.75, 14.75, 12.75, 6.5, 4.5, 6.5, 26.5, 14.5, 10.75, 10.75, 24.75, 43.75, 28.0, 9.0, 18.75, 19.5, 21.75, 23.5, 49.5, 22.75, 12.25, 6.25, 5.5, 11.0 ⟩
c_2	⟨ 13.0, 12.25, 18.0, 18.75, 11.25, 46.5, 87.75, 24.25, 14.5, 12.5, 22.25, 19.0, 15.0, 32.25, 104.5, 79.0, 45.75, 10.75, 3.5, 7.5, 10.5, 8.75, 36.0, 58.75, 9.25, 4.25, 3.25, 6.0, 30.75, 37.25, 16.75, 7.5, 22.0, 22.25, 13.25, 4.25, 2.25, 3.75, 17.25, 12.25, 11.25, 11.0, 23.75, 48.25, 35.75, 11.75, 21.75, 23.0, 33.25, 51.5, 96.25, 53.5, 29.75, 16.0, 16.0, 27.75, 21.75, 42.5, 84.5, 38.0, 24.0, 20.75, 8.0, 10.25, 27.25, 37.0, 23.25, 9.5, 4.0, 1.25, 1.5, 7.0, 14.0, 19.5, 63.25, 48.5, 52.0, 32.75, 16.25, 13.5, 17.5, 47.0, 97.25, 55.75, 62.0, 73.5, 72.5, 39.75, 37.25, 48.75, 89.0, 37.75, 14.25, 30.75, 69.25, 65.25, 27.5, 42.5, 21.25, 21.5, 14.0, 3.0, 6.75, 9.5, 13.0, 28.0, 86.75, 69.0, 20.5, 18.5, 79.5, 36.0, 6.0, 38.25, 116.75, 57.25, 11.5, 38.25, 115.5, 34.0, 7.75, 58.0, 116.75, 27.0, 5.0, 29.0, 111.0, 28.25 ⟩
c_3	⟨ 6.75, 7.0, 14.5, 22.25, 28.75, 13.25, 5.5, 6.5, 17.5, 40.0, 72.75, 40.75, 42.0, 36.75, 28.75, 22.75, 30.5, 44.25, 84.25, 40.75, 16.75, 24.75, 35.25, 36.5, 32.75, 27.5, 17.25, 6.25, 7.0, 8.75, 9.25, 21.5, 10.75, 21.0, 79.25, 53.0, 18.75, 13.0, 30.0, 16.5, 8.5, 31.75, 117.0, 57.0, 23.75, 49.5, 106.0, 33.5, 19.0, 68.75, 117.0, 31.25, 8.75, 34.75, 109.0, 56.5, 21.75, 56.5, 68.75, 20.0, 12.0, 14.75, 25.25, 18.5, 13.0, 20.25, 41.75, 42.5, 19.25, 20.25, 67.5, 36.5, 5.75, 34.25, 117.0, 47.25, 16.0, 36.75, 117.0, 32.75, 14.5, 59.25, 117.0, 28.75, 5.25, 39.75, 117.0, 32.25, 23.75, 48.25, 49.25, 24.75, 22.5, 35.75, 68.75, 18.5, 11.25, 48.0, 23.75, 5.75, 4.25, 8.25, 12.75, 11.0, 4.5, 21.0, 36.25, 13.75, 10.0, 22.0, 67.25, 17.5, 7.0, 19.25, 41.5, 25.25, 4.5, 14.5, 72.75, 23.25, 9.25, 9.25, 21.5, 57.0, 21.0, 13.0, 13.25, 9.25 ⟩

Table A.3: *Centroids computed by K-Means algorithm (with $K = 3$) from the SIFT descriptors in Tables A.1 and A.2.*

Bibliography

Aharon and Elad (2006) Michal Aharon and Michael Elad. *Overcomplete dictionaries for sparse representation of signals*. Tese de Doutorado, Computer Science Department, Technion. Cited on page(s) 28

Aharon *et al.* **(2006)** Michal Aharon, Michael Elad and Alfred Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *Signal Processing, IEEE Transactions on*, 54(11):4311–4322. Cited on page(s) 13, 28, 39

Araujo *et al.* **(2018)** Rafael Will M de Araujo, Roberto Hirata and Alain Rakotomamonjy. Concave losses for robust dictionary learning. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2176–2180. IEEE. Cited on page(s) 3, 27

Avila *et al.* **(2011)** Sandra Avila, Nicolas Thome, Matthieu Cord, Eduardo Valle and A de A Araújo. Bossa: Extended bow formalism for image classification. In *2011 18th IEEE International Conference on Image Processing*, pages 2909–2912. IEEE. Cited on page(s) 15, 19, 25

Avila *et al.* **(2013)** Sandra Avila, Nicolas Thome, Matthieu Cord, Eduardo Valle and Arnaldo De A AraúJo. Pooling in image representation: The visual codeword point of view. *Computer Vision and Image Understanding*, 117(5):453–465. Cited on page(s) 15, 17, 19, 25

Bach *et al.* **(2012)** Francis Bach, Rodolphe Jenatton, Julien Mairal, Guillaume Obozinski *et al.* Optimization with sparsity-inducing penalties. *Foundations and Trends® in Machine Learning*, 4(1):1–106. Cited on page(s) 31

Bainbridge *et al.* **(2013)** Wilma A Bainbridge, Phillip Isola and Aude Oliva. The intrinsic memorability of face photographs. *Journal of Experimental Psychology: General*, 142(4): 1323. Cited on page(s) 42

Bao *et al.* **(2013)** Chenglong Bao, Jian-Feng Cai and Hui Ji. Fast sparsity-based orthogonal dictionary learning for image restoration. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3384–3391. Cited on page(s) 27

Bay *et al.* **(2006)** Herbert Bay, Tinne Tuytelaars and Luc Van Gool. Surf: Speeded up robust features. In *Computer vision–ECCV 2006*, pages 404–417. Springer. Cited on page(s) 10

Borgi *et al.* **(2014)** Mohamed Anouar Borgi, Maher El'Arbi, Demetrio Labate and Chokri Ben Amar. Face, gender and race classification using multi-regularized features learning. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 5277–5281. IEEE. Cited on page(s) 6, 55

Broomhead and Lowe (1988) David S Broomhead and David Lowe. Radial basis functions, multi-variable functional interpolation and adaptive networks. Technical report, Royal Signals and Radar Establishment Malvern (United Kingdom). Cited on page(s) 48

Chandola *et al.* (2007) Varun Chandola, Arindam Banerjee and Vipin Kumar. Outlier detection: A survey. *ACM Computing Surveys*. Cited on page(s) 36

Chu *et al.* (2010) Wen-Sheng Chu, Chun-Rong Huang and Chu-Song Chen. Identifying gender from unaligned facial images by set classification. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 2636–2639. IEEE. Cited on page(s) 5, 6

Csurka *et al.* (2004) Gabriella Csurka, Christopher Dance, Lixin Fan, Jutta Willamowski and Cédric Bray. Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, volume 1, pages 1–22. Prague. Cited on page(s) 9, 12, 13

Dong *et al.* (2011) Weisheng Dong, Xin Li, Lei Zhang and Guangming Shi. Sparsity-based image denoising via dictionary learning and structural clustering. In *CVPR 2011*, pages 457–464. IEEE. Cited on page(s) 27

Duan *et al.* (2018) Mingxing Duan, Kenli Li, Canqun Yang and Keqin Li. A hybrid deep learning cnn–elm for age and gender classification. *Neurocomputing*, 275:448–461. Cited on page(s) 6

Elad and Aharon (2006) Michael Elad and Michal Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image processing*, 15(12):3736–3745. Cited on page(s) 27

Fan and Li (2001) Jianqing Fan and Runze Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American statistical Association*, 96 (456):1348–1360. Cited on page(s) 32

Fawcett (2006) Tom Fawcett. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874. Cited on page(s) 40

Fei-Fei and Perona (2005a) Li Fei-Fei and Pietro Perona. A bayesian hierarchical model for learning natural scene categories. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 524–531. IEEE. Cited on page(s) 46

Fei-Fei and Perona (2005b) Li Fei-Fei and Pietro Perona. A bayesian hierarchical model for learning natural scene categories. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 524–531. IEEE. Cited on page(s) 10

Fu *et al.* (2014) Siyao Fu, Haibo He and Zeng-Guang Hou. Learning race from face: A survey. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36(12):2483–2509. Cited on page(s) 1, 6

Gallagher and Chen (2009) Andrew C Gallagher and Tsuhan Chen. Understanding images of groups of people. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 256–263. IEEE. Cited on page(s) 5

Golub and Van Loan (2012) Gene H Golub and Charles F Van Loan. *Matrix computations*, volume 3. JHU Press. Cited on page(s) 28

Gorodnitsky and Rao (1997) Irina F Gorodnitsky and Bhaskar D Rao. Sparse signal reconstruction from limited data using focuss: A re-weighted minimum norm algorithm. *IEEE Transactions on signal processing*, 45(3):600–616. Cited on page(s) 28

Grauman and Darrell (2005) Kristen Grauman and Trevor Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1458–1465. IEEE. Cited on page(s) xi, 14, 23

Guo and Mu (2010) Guodong Guo and Guowang Mu. A study of large-scale ethnicity estimation with gender and age variations. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pages 79–86. IEEE. Cited on page(s) 6, 7, 50

Harris (1954) Zellig S. Harris. Distributional structure. *Word*, 10:(2-3):142–162. doi: 10.1080/00437956.1954.11659520. Cited on page(s) 9

Huang *et al.* **(2007)** Gary B Huang, Manu Ramesh, Tamara Berg and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical report, Technical Report 07-49, University of Massachusetts, Amherst. Cited on page(s) 5

Jiang *et al.* **(2015)** Wenhao Jiang, Feiping Nie and Heng Huang. Robust dictionary learning with capped l1-norm. In *IJCAI*, pages 3590–3596. Cited on page(s) 32, 35

Jiang *et al.* **(2011)** Zhuolin Jiang, Zhe Lin and Larry S Davis. Learning a discriminative dictionary for sparse coding via label consistent k-svd. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1697–1704. IEEE. Cited on page(s) 13, 28, 29, 31

Juefei-Xu *et al.* **(2016)** Felix Juefei-Xu, Eshan Verma, Parag Goel, Anisha Cherodian and Marios Savvides. Deepgender: Occlusion and low resolution robust facial gender classification via progressively trained convolutional neural networks with attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 68–77. Cited on page(s) 6, 55

Khan *et al.* **(2012)** Rahat Khan, Cécile Barat, Damien Muselet and Christophe Ducottet. Spatial orientations of visual word pairs to improve bag-of-visual-words model. In *Proceedings of the British Machine Vision Conference*, pages 89–1. BMVA Press. Cited on page(s) xi, 15, 16, 19, 24, 25

Krizhevsky *et al.* **(2012)** Alex Krizhevsky, Ilya Sutskever and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105. Cited on page(s) 7

Lazebnik *et al.* **(2006)** Svetlana Lazebnik, Cordelia Schmid and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2169–2178. IEEE. Cited on page(s) xi, 2, 13, 14, 19, 20, 22, 23

Learned-Miller *et al.* **(2016)** Erik Learned-Miller, Gary B Huang, Aruni RoyChowdhury, Haoxiang Li and Gang Hua. Labeled faces in the wild: A survey. In *Advances in face detection and facial image analysis*, pages 189–248. Springer. Cited on page(s) 42

Liu *et al.* **(2011)** Lingqiao Liu, Lei Wang and Xinwang Liu. In defense of soft-assignment coding. In *2011 International Conference on Computer Vision*, pages 2486–2493. IEEE. Cited on page(s) 17

Lowe (1999) David G Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee. Cited on page(s) 9

Lu *et al.* **(2013)** Cewu Lu, Jiaping Shi and Jiaya Jia. Online robust dictionary learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 415–422. Cited on page(s) 36, 40

Mahalanobis (1936) Prasanta Chandra Mahalanobis. On the generalized distance in statistics. In *Proceedings of the National Institute of Sciences (Calcutta)*. National Institute of Science of India. Cited on page(s) 10

Mairal *et al.* **(2009a)** Julien Mairal, Francis Bach, Jean Ponce and Guillermo Sapiro. Online dictionary learning for sparse coding. In *Proceedings of the 26th annual international conference on machine learning*, pages 689–696. ACM. Cited on page(s) 35, 36

Mairal *et al.* **(2009b)** Julien Mairal, Jean Ponce, Guillermo Sapiro, Andrew Zisserman and Francis R. Bach. Supervised dictionary learning. In D. Koller, D. Schuurmans, Y. Bengio and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 1033–1040. Curran Associates, Inc. URL http://papers.nips.cc/paper/3448-supervised-dictionary-learning.pdf. Cited on page(s) 27

Mairal *et al.* **(2010)** Julien Mairal, Francis Bach, Jean Ponce and Guillermo Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11(Jan):19–60. Cited on page(s) 36

Martinez (1998) Aleix M Martinez. The ar face database. *CVC Technical Report24*. Cited on page(s) 2, 6, 43

Ng *et al.* **(2012)** Choon Boon Ng, Yong Haur Tay and Bok Min Goi. Vision-based human gender recognition: A survey. *arXiv preprint arXiv:1204.1611*. Cited on page(s) 1

Ng *et al.* **(2015)** Choon-Boon Ng, Yong-Haur Tay and Bok-Min Goi. A review of facial gender recognition. *Pattern Analysis and Applications*, 18(4):739–755. Cited on page(s) 1, 5

Nie *et al.* **(2010)** Feiping Nie, Heng Huang, Xiao Cai and Chris H Ding. Efficient and robust feature selection via joint $\ell_{2,1}$-norms minimization. In *Advances in neural information processing systems*, pages 1813–1821. Cited on page(s) 32

Ojala *et al.* **(1996)** Timo Ojala, Matti Pietikäinen and David Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern recognition*, 29(1):51–59. Cited on page(s) 5

Ojala *et al.* **(2002)** Timo Ojala, Matti Pietikäinen and Topi Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 24(7):971–987. Cited on page(s) 5

Okabe *et al.* **(2009)** Atsuyuki Okabe, Barry Boots, Kokichi Sugihara and Sung Nok Chiu. *Spatial tessellations: concepts and applications of Voronoi diagrams*, volume 501. John Wiley & Sons. Cited on page(s) 22

Pati *et al.* **(1993)** Yagyensh Chandra Pati, Ramin Rezaiifar and PS Krishnaprasad. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Signals, Systems and Computers, 1993. 1993 Conference Record of The Twenty-Seventh Asilomar Conference on*, pages 40–44. IEEE. Cited on page(s) 28

Phillips *et al.* **(2000)** P Jonathon Phillips, Hyeonjoon Moon, Syed A Rizvi and Patrick J Rauss. The feret evaluation methodology for face-recognition algorithms. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(10):1090–1104. Cited on page(s) 2, 5, 42

Rakotomamonjy (2013) A Rakotomamonjy. Applying alternating direction method of multipliers for constrained dictionary learning. *Neurocomputing*, 106:126–136. Cited on page(s) 31

Ramirez *et al.* **(2010)** Ignacio Ramirez, Pablo Sprechmann and Guillermo Sapiro. Classification and clustering via dictionary learning with structured incoherence and shared features. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3501–3508. IEEE. Cited on page(s) 27

Ramón-Balmaseda *et al.* **(2012)** Enrique Ramón-Balmaseda, Javier Lorenzo-Navarro and Modesto Castrillón-Santana. Gender classification in large databases. In *Progress in pattern recognition, image analysis, computer vision, and applications*, pages 74–81. Springer. Cited on page(s) 5, 6

Ricanek Jr and Tesafaye (2006) Karl Ricanek Jr and Tamirat Tesafaye. Morph: A longitudinal image database of normal adult age-progression. In *Automatic Face and Gesture Recognition, 2006. FGR 2006. 7th International Conference on*, pages 341–345. IEEE. Cited on page(s) 2, 5, 42

Rockafellar (2015) Ralph Tyrell Rockafellar. *Convex analysis*. Princeton University Press. Cited on page(s) 33

Rubinstein *et al.* **(2008)** Ron Rubinstein, Michael Zibulevsky and Michael Elad. Efficient implementation of the k-svd algorithm using batch orthogonal matching pursuit. *CS Technion*, 40(8):1–15. Cited on page(s) 28

Sprechmann and Sapiro (2010) Pablo Sprechmann and Guillermo Sapiro. Dictionary learning and sparse coding for unsupervised clustering. In *2010 IEEE international conference on acoustics, speech and signal processing*, pages 2042–2045. IEEE. Cited on page(s) 27

Thomaz and Giraldi (2010) Carlos Eduardo Thomaz and Gilson Antonio Giraldi. A new ranking method for principal components analysis and its application to face image analysis. *Image and Vision Computing*, 28(6):902–913. Cited on page(s) 6

Tibshirani (1996) Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288. Cited on page(s) 31

Tome *et al.* **(2013)** Pedro Tome, Julian Fierrez, Ruben Vera-Rodriguez and Daniel Ramos. Identification using face regions: Application and assessment in forensic scenarios. *Forensic science international*, 233(1-3):75–83. Cited on page(s) 42

Van De Sande *et al.* **(2010)** Koen Van De Sande, Theo Gevers and Cees Snoek. Evaluating color descriptors for object and scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1582–1596. Cited on page(s) 46

Viola and Jones (2001) Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–511. IEEE. Cited on page(s) 44

Wang *et al.* **(2016a)** De Wang, Feiping Nie and Heng Huang. Fast robust non-negative matrix factorization for large-scale data clustering. In *25th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2104–2110. Cited on page(s) 32

Wang *et al.* **(2017)** Fang Wang, Hu Han, Shiguang Shan and Xilin Chen. Deep multi-task learning for joint prediction of heterogeneous face attributes. In *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, pages 173–179. IEEE. Cited on page(s) 7

Wang *et al.* **(2013)** Hua Wang, Feiping Nie, Weidong Cai and Heng Huang. Semi-supervised robust dictionary learning via efficient l0-norms minimization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1145–1152. Cited on page(s) 32, 35

Wang *et al.* **(2016b)** Wei Wang, Feixiang He and Qijun Zhao. Facial ethnicity classification with deep convolutional neural networks. In *Chinese Conference on Biometric Recognition*, pages 176–185. Springer. Cited on page(s) 7

Zhang *et al.* **(2010)** Cun-Hui Zhang *et al.* Nearly unbiased variable selection under minimax concave penalty. *The Annals of statistics*, 38(2):894–942. Cited on page(s) 32