FEDERAL UNIVERSITY OF TECHNOLOGY - PARANÁ
GRADUATE PROGRAM IN ELECTRICAL AND COMPUTER ENGINEERING

HUGO ALBERTO PERLIN

A CONTRIBUTION TO SEMANTIC DESCRIPTION OF IMAGES AND VIDEOS: AN APPLICATION OF SOFT BIOMETRICS

DOCTORAL THESIS

CURITIBA

2015

HUGO ALBERTO PERLIN

A CONTRIBUTION TO SEMANTIC DESCRIPTION OF IMAGES AND VIDEOS: AN APPLICATION OF SOFT BIOMETRICS

Doctoral Thesis presented to the Graduate Program in Electrical and Computer Engineering of the Federal University of Technology - Paraná as partial fulfillment of the requirements for the title of "Doctor of Science (D.Sc.)" – Concentration Area: Computer Engineering.

Thesis Advisor: Heitor Silvério Lopes

CURITIBA

2015

Dados Internacionais de Catalogação na Publicação

P451c
2015

Perlin, Hugo Alberto
A contribution to semantic description of images and vídeos : an application of soft biometrics / Hugo Alberto Perlin.-- 2015.
109 f. : il. ; 30 cm.

Texto em inglês, com resumo em português
Tese (Doutorado) - Universidade Tecnológica Federal do Paraná. Programa de Pós-graduação em Engenharia Elétrica e Informática Industrial, Curitiba, 2015
Bibliografia: p. 103-109

1. Visão por computador. 2. Aprendizado do computador. 3. Processamento de imagens. 4. Interpretação de imagens. 5. Linguagem de programação (Computadores) - Semântica. 6. Engenharia elétrica - Dissertações. I. Lopes, Heitor Silvério. II. Universidade Tecnológica Federal do Paraná - Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial. III. Título.

CDD: Ed. 22 -- 621.3

Biblioteca Central da UTFPR, Câmpus Curitiba

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
Câmpus Curitiba

Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial

Título da Tese Nº. 129

A Contribution to Semantic Description of Images and Videos: an Application of Soft Biometrics

por

Hugo Alberto Perlin

Orientador: Prof. Dr. Heitor Silvério Lopes

Esta tese foi apresentada como requisito parcial à obtenção do grau de DOUTOR EM CIÊNCIAS – Área de Concentração: **Engenharia de Computação**, pelo Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial – CPGEI – da Universidade Tecnológica Federal do Paraná – UTFPR, às **14:00h** do dia **08 de dezembro de 2015**. O trabalho foi aprovado pela Banca Examinadora, composta pelos doutores:

Prof. Dr. Heitor Silvério Lopes
(Presidente – UTFPR)

Prof. Dr. Alessandro Lameiras Koerich
(ÉCOLE DE TECHNOLOGIE SUPÉRIEURE)

Prof. Dr. Alceu de Souza Britto Jr.
(PUC-PR)

Prof. Dr. Chidambaram Chidambaram
(UDESC – São Bento do Sul)

Profª. Drª. Lúcia Valéria Ramos de Arruda
(UTFPR)

Prof. Dr. Hugo Vieira Neto
(UTFPR)

Visto da Coordenação:

Prof. Dr. Emilio Carlos Gomes Wille
(Coordenador do CPGEI)

To my beloved son Bento.

ACKNOWLEDGEMENTS

During the doctoral program, we are constantly put to test. Several times there are many more questions and doubts rather than clarity on how to proceed. But certainly, the people who were constantly at my side, near or far, were responsible for getting me to the end. To them, I am very grateful.

First, my wife Michele and my son Bento, by love, support, patience, warmth and comfort. Several times my presence was far, but even so, they continued to support and encourage me. This work was possible because of them.

My parents Antonio and Marinez, and my brother João, who taught me the true values of life and built the foundation for my growth.

I would like to thank Professor Heitor Silvério Lopes for the confidence, patience and persistence. From a mentoring relationship, a good friendship raised. Thank you for the lessons and the kilometers shared.

To friends and companions of Bioinfo laboratory, Cesar Vargas Benitez, Ademir Gabardo, Chidambaram Chidambaram, Jonas Fries Krause, Rodrigo Silva, Manassés Ribeiro, Leandro Hattori, André Barros, Gláucio Porcides and Fernando Carvalho de Souza, for good conversations and laughter, the huge amount of coffee, the discussion of ideas and suggestions. Much of this work has a good piece of your contribution.

The Federal Institute of Paraná, for their support in allowing part of the PhD to be carried out with exclusive dedication.

In short, everyone who in one way or another showed their appreciation for me. Thank you.

“We can only see a short distance ahead, but we can see plenty there that needs to be done.”

Alan Turing

ABSTRACT

PERLIN, HUGO ALBERTO. A CONTRIBUTION TO SEMANTIC DESCRIPTION OF IMAGES AND VIDEOS: AN APPLICATION OF SOFT BIOMETRICS. 110 f. Doctoral Thesis – Graduate Program in Electrical and Computer Engineering, Federal University of Technology - Paraná. Curitiba, 2015.

Humans have a high ability to extract information from visual data acquired by sight. Trough a learning process, which starts at birth and continues throughout life, image interpretation happens almost instinctively. At a glance, one can easily describe a scene with reasonable accuracy, naming its main components. Usually, this is done by extracting low-level features such as edges, shapes and textures, and associating them to high level meanings. In this way, a semantic description of the scene is done. An example of this is, the human ability to recognize and describe other people's physical and behavioural characteristics, or biometrics. Soft-biometrics also represents inherent characteristics of human body and behaviour, but they do not allow unique person identification. The computer vision area aims to develop methods able to performing visual interpretation with human similar performance. This thesis aims to propose computer vision methods which allows high level information extraction from images and videos in the form of soft biometrics. This problem is approached in two ways, trough unsupervised and supervised learning methods. The first way, is intended to group images via an automatic feature extraction learning, using both convolution techniques, evolutionary computing and clustering. In the 1st approach the images used contain faces and people. The second approach employs convolutional neural networks, which have the ability to operate directly on raw images, learning both feature extraction and classification processes. Here, images are classified according to gender and clothes, divided into upper and lower parts of the human body. The first approach, when tested with different image datasets obtained an accuracy of approximately 80% for faces and non-faces and 70% for people and non-people. The second approach, which was tested using images and videos, have obtained an accuracy of about 70% for gender, 80% for the upper clothes and 90% for lower clothes. The results of these case studies show that the proposed methods are promising, allowing the realization of automatic annotation of high level image information. This opens possibilities for development of applications in diverse areas such as content-based image and video search and automatic video surveillance, reducing human effort in the task of manual annotation and monitoring.

Keywords: COMPUTER VISION, MACHINE LEARNING, DEEP LEARNING, SEMANTIC DESCRIPTION, SOFT BIOMETRICS

RESUMO

PERLIN, HUGO ALBERTO. UMA CONTRIBUIÇÃO PARA DESCRIÇÃO SEMÂNTICA DE IMAGENS E VÍDEOS: UMA APLICAÇÃO DE BIOMETRIAS FRACAS. 110 f. Tese de doutorado – Programa de Pós-graduação em Engenharia Elétrica e Informática Industrial, Universidade Tecnológica Federal do Paraná. Curitiba, 2015.

Os seres humanos possuem uma alta capacidade de extrair informações de dados visuais, adquiridos por meio da visão. Através de um processo de aprendizado, que se inicia ao nascer e continua ao longo da vida, a interpretação de imagens passa a ser feita de maneira quase instintiva. Em um relance, uma pessoa consegue facilmente descrever com certa precisão os componentes principais que compõem uma determinada cena. De maneira geral, isto é feito extraindo-se características de baixo nível, como arestas, texturas e formas, e associando-as com significados de alto nível. Ou seja, realiza-se uma descrição semântica da cena. Um exemplo disto é a capacidade de reconhecer pessoas e descrever suas características físicas e comportamentais. A área de visão computacional tem como principal objetivo desenvolver métodos capazes de realizar uma interpretação visual com desempenho similar aos humanos. Estes métodos englobam conhecimentos de aprendizado de máquina e processamento de imagens. Esta tese tem como objetivo propor métodos de visão computacional que permitam a extração de informações de alto nível na forma de biometrias fracas. Estas biometrias representam características inerentes ao corpo e ao comportamento humano. Porém, não permitem a identificação unívoca de uma pessoa. Para tanto, este problema foi abordado de duas formas, utilizando aprendizado não-supervisionado e supervisionado. A primeira busca agrupar as imagens através de um processo de aprendizado automático de extração de características, empregando técnicas de convoluções, computação evolucionária e agrupamento. Nesta abordagem as imagens utilizadas contêm faces e pessoas. A segunda abordagem emprega redes neurais convolucionais, que possuem a capacidade de operar diretamente sobre imagens cruas, aprendendo tanto o processo de extração de características quanto a classificação. Aqui as imagens são classificadas de acordo com gênero e roupas, divididas em parte superior e inferior do corpo humano. A primeira abordagem, quando testada com diferentes bancos de imagens, obteve uma acurácia de aproximadamente 80% para faces e não-faces e 70% para pessoas e não-pessoas. A segunda, testada utilizando imagens e vídeos, obteve uma acurácia de cerca de 70% para gênero, 80% para roupas da parte superior e 90% para a parte inferior. Os resultados destes estudos de casos, mostram que os métodos propostos são promissores, permitindo a realização de anotação automática de informações de alto nível. Isto abre possibilidades para o desenvolvimento de aplicações em diversas áreas, como busca de imagens e vídeos baseada em conteúdo e segurança por vídeo, reduzindo o esforço humano nas tarefas de anotação manual e monitoramento.

Palavras-chave: VISÃO COMPUTACIONAL, APRENDIZADO DE MÁQUINA, APRENDIZADO PROFUNDO, DESCRIÇÃO SEMÂNTICA, BIOMETRIAS FRACAS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS AND ABBREVIATIONS

ANN	Artificial Neural Networks
SVM	Support Vector Machine
SIFT	Scale Invariant Feature Transform
GLOH	Gradient Location and Orientation Histogram
BRISK	Binary Robust Invariant Scalable Keypoints
BRIEF	Binary Robust Independent Elementary Features
HOG	Histogram of Oriented Gradients
SURF	Speeded Up Robust Features
LBP	Local Binary Patter
PCA	Principal Component Analysis
LDA	Linear Discriminant Analysis
CNN	Convolutional Neural Networks
BRS	biometrics recognition system
GD	Gradient Descent
SGD	Stochastic Gradient Descent
GPUs	Graphical Processing Units
AE	Evolutionary Algorithms
SC	Silhouette Coefficient
KNN	K-nearest neighbours
GMM	Guassian Mixture Model
IG	Information Gain
KW	Kruskal-Wallis score
GPGPU	General-Purpose computation on Graphics Processing Units
CPU	Central Processing Unit
WEKA	Waikato Environment for Knowledge Analysis

CONTENTS

1 INTRODUCTION

Human vision is a complex and incredible sense that allows us to interact with the surrounding world in an effortless way. Evolved throughout millions of years, our eyes and brain work together to process and extract information from data received in the form of light. With a blink of an eye, we are able to locate ourselves into a 3D world, recognize objects, identify other people and even their emotions. All this is done almost unconsciously.

Vision is one the most important human senses, constituting the main data interface for a seeing person. A clue to its importance can be checked in our cities, society and entertainments, which are mainly thought and designed for seeing people. A better understanding of the human visual system has drawn attention from various knowledge areas, such as biology, neurology and computer science.

Although it seems that our visual capabilities are innate, we are continuously improving our visual comprehension mechanisms. Since birth, we develop our visual comprehension abilities by using both supervised and unsupervised methods. This process builts a virtual model, which is later used as part or the recognition process (MILNER; GOODALE, 1995).

As stated before, we receive huge amounts of data through our vision. That is known as low-level data, such as dots, edges, shapes, color and textures. Our brain is responsible of merging and analysing these data, seeking to associate with a semantic meaning or definition. Based on this, we can make decisions and take actions. Here, semantic meaning or high level interpretation is defined as the results of how humans can describe their surroundings by using concepts and definitions determined by their culture and society. In this way, two different people could achieve very different interpretations about the same image. In other words, this process is sometimes ambiguous and it does not have a correct answer.

Since vision is our main information gathering interface, an interesting choice would be to replicate this capability using digital computers. This has been the research field of several teams all around the world, and it is known as computer vision. This area can be viewed as the combination of machine learning and image processing areas, enabling visual comprehension to computers. Developing methods that could allow computers to extract high level information similarly as we do, could allow a lot of applications in areas such as security, health, entertainment and others.

An example of this use could be search in image or video databases looking for a specific content. Nowadays, to achieve such capacity, a huge human effort is requested for a systematic annotation. Since the size of those databases increases rapidly, this seems to be an intractable problem. Developing automatic computer methods to insert appropriate annotations could help humans to have a better use of such data.

Therefore, the main focus of this work is related to exploration and development of methods that allow computers to receive, process and deliver high level interpretation on images and videos.

1.1 PROBLEM DEFINITION

The problem addressed in this work consists in, given a set of images or videos that contains people, perform their description using high level concepts, such as soft biometrics.

Some attributes that can be used to describe humans, but cannot be used to unequivocally determine their identities are known as soft biometrics. Gender, hair, gait, ethnicity, height, weight, clothes, are all examples of soft biometrics. A main motivation factor to explore this kind of information is the requirement of little or no cooperation of individuals.

More specifically, in this work the focus is to classify people according to three different attributes: upper clothes, lower clothes and gender. More precisely, given the image of a person, we aim to identify the type of upper and lower clothes he/she is using, as well as his/her gender. Additionally, we are interested in other kind of biometrics, such as those ones that define a person's face and a person's body. In general words, this is a pattern recognition problem.

A motivation for the development of methods to deal this problem is to allow content-based image and video retrieval. Such methods could enable, for instance, to search a video database using high level queries, such as "Find all men with blue t-shirt and black pants". Hence, this type of system could save many hours of human effort to analyse and classify those images.

Although it seems very easy for a human to describe a person's clothes or gender, when approached by a computer, this problem becomes very complex. The main cause for this complexity remains on the need for performing high level semantic interpretation based on the input data, images and videos, composed by luminosity values captured from the environment. The distance between the input data and the high level information is known as the semantic gap, which is present in most computer vision problems (SNOEK; SMEULDERS, 2010).

A possible way to reduce the semantic gap and, consequently, to find appropriate solutions for the soft biometric extraction problem, is to improve and employ machine learning methods. The basic idea is to mimic the human learning capacity and to allow computers to learn.

In general, approaches to the pattern recognition problem are based on two parts, a feature extraction operation, followed by a machine learning method. The main objective of feature extraction is to transform the input data, into a more informative representation, allowing a better performance of the classifier/clustering method. Several techniques are available in the literature. Finding an appropriate feature extractor is a complex problem which requires a solid knowledge of both method and application.

Machine learning techniques can be roughly divided into three types, supervised, unsupervised and semi-supervised. In the first one, a series of annotated examples are presented to a given classifier, which should find a way to associate each example to the class that it belongs to. Consequently, humans have to create the example set, which consists in the manual classification of each example according to specific purposes. Classification performance of supervised machine learning methods are directly related to the quality, diversity and quantity of the provided examples. In unsupervised learning, an unlabelled dataset is given to the algorithm, which seeks to discover some internal structure within the data, allowing the creation of a virtual data model. Here, almost no human effort is needed, since it is not necessary to classify the data previously. The third category is based on both types mentioned, by using a small set of labelled samples, semi-supervised methods try to learn a classification model and extend this model to unlabelled data.

In this work, we explore both supervised and unsupervised methods, trying to learn features directly from the input data, therefore leading to the desired classification. A supervised method was chosen to achieve the extraction of high level information in the form of soft biometrics, by using the human knowledge embedded into labelled examples.

Since the image annotation process requires human effort, and unlabelled images are largely available with a low cost, approaches seeking to a better use of this kind of data are desired. During the research process a question came up: is it possible to separate raw unlabelled images into classes? To verify this possibility, we propose an unsupervised feature learning method. It was constructed to learn a feature extractor method based on the raw unlabelled input images. The feature vectors should allow to separate images into classes, showing that the images bring some information about its semantics together with the low level data.

In an overview, both research topics developed in this work are concerned to deal with the extraction of high level information from images and videos.

1.2 OBJECTIVE

The objective of this work is to propose a method for high level semantic interpretation, in the form of soft biometrics, for images and videos.

1.3 SPECIFIC OBJECTIVES

The objective is subdivided into some specific objectives to clarify our approaches and contributions:

- to propose a method for unsupervised feature learning for raw image separation;
- to verify if it is possible to separate raw images based only on input data, achieving class information;
- to propose a method based on Convolutional Neural Networks for the extraction of soft biometric information from images and videos;
- to verify the classification performance of the methods using different image datasets;

1.4 STRUCTURE OF THE THESIS

This work is structured as follows. The background concepts necessary to the understanding of the proposed methods, as well as a review of the related literature is done in Chapter 2. Chapter 3 discusses the proposed methods for both unsupervised feature learning and extraction of soft biometrics from images and videos. Chapter 4 shows the experiments, results obtained and their discussion. Finally, considerations about the work and the results, general conclusions, proposals for future work, and contributions are shown in Chapter 5.

2 BACKGROUND AND RELATED WORK

This chapter is aimed at bringing to reader the basic concepts used in the work, enabling the comprehension of the proposed methods.

2.1 COMPUTER VISION AND IMAGE UNDERSTANDING

Vision is the human sense responsible for the major amount of information received from the external world. It allows us to interact with the components of the world in an effortless way. A large part of our brain is devoted to process the visual data, allowing the extraction of information and the consequent decision making. When we are observing the surrounding world, visual stimulus are received by our eyes, converted to electrical signals and processed by our brain. In essence, the result of this process is a set of descriptions, each one in respect to the components of the scene. In this way, image understanding is the process of extracting high level information from visual input data (XIE et al., 2011).

Analysis of image contents is a complex process. Studies in the neuroscience field try to understand better how a human being is capable of comprehending the surrounding world by means of the images received by the eyes. Eyes and brain work together, and based on previous knowledge, objects, people, places and landscapes are recognized and identified (HOFFMAN; LOGOTHETIS, 2009). This kind of information, obtained through the human interpretation capability, is defined in this work as ”high level information”.

An image is a composition of low-level artifacts, such as edges, shapes, gradients, which are less informative when analysed separately. Usually, we are interested in high-level information such as the concepts inside an image. A concept is a description about the scene contents, done by a specific human under a specific point-of-view. In this way, two humans could describe the same image using totally different concepts. Making direct link between some set of low-level content and a high-level concept is a hard task, even when made by humans.

Between the low-level data and high level interpretation there is a well know gap. It is called semantic gap, imposed by the difficulty in finding a direct correspondence between these two extremes. In a broad sense, a work in computer vision seeks to narrow this gap, and achieve visual information extraction automatically.

Since childhood, we are trained to look at the surrounding world and to try to make sense from it through visual information. This training process is continuously done by the exposition to examples, from which we construct models and store them for future comparisons and references. This learning process is extremely complex and, at the same time, constructive. The brain takes care of doing the work, based on an extensive visual experience, built along life, since our visual interpretation capacity is not an innate process (HOFFMAN; LOGOTHETIS, 2009).

The way the human brain performs this process can be divided into two parts: discrimination and generalization. The first one deals with which types of features relating to a particular object are extracted and stored for a later classification. In other words, this is the actual learning process. The second role is as important as learning, because even with major changes in the visual appearance of an object, such as the presence of occlusion, change of brightness, or even a certain level of distortion, the recognition ability remains high. Our brain has a high capacity to generalize and associate characteristics during visual information extraction. This phenomenon is essential to improve our visual capacity (HOFFMAN; LOGOTHETIS, 2009).

Developing methods and software which allow an automatic comprehension of the world by computers is, essentially, the main objective of computer vision. In the last decades, the advances in the area have produced great results, but there is still a long journey to achieve something comparable to the human vision capacity.

Computer vision is a research field with many open problems. One of these is pattern recognition in digital images, where a specific pattern should be discovered by computer algorithms and used, for instance, for the classification into categories (BISHOP, 2006). This pattern can be an object, face or person, for example. There are several applications in which pattern recognition is a fundamental step, such as person recognition, video surveillance, scene description, autonomous location and navigation.

The number of possible applications of computer vision methods is big, mainly nowadays where the number of image capturing devices has increased exponentially and are spread in many places. These leads to enormous images and video databases, which are complex to manage, specially when someone wants to search for a specific kind of visual content. A central step in visual interpretation is the ability to find and extract patterns, which is discussed in the next section.

2.1.1 PATTERN RECOGNITION

The definition of a pattern is subtle and broad in the sense that it depends on the context. However, in general terms, a pattern is a certain regularity found in data that tends to be repeated at different times/locations (BISHOP, 2006). In general, complex patterns are composed of some primitive structures and have a semantic meaning associated to them, concerning classification problems. The great difficulty in automatic pattern recognition is to find the correct relationship between these basic structures (edges, shapes, area, color) and the semantic meaning, which is commonly known as a semantic gap (SNOEK; SMEULDERS, 2010).

In an attempt to narrow the semantic gap, several methods have been employed by researchers. Most of them are based on transformations of the input data, known as feature extraction. Instead of dealing directly with pixel intensity, a better way is to employ methods to transform the images into a more informative representation. An example, would be extract gradient information from image, since it would give a better description of edges than the hole image itself.

Using the extracted features, the next step is to find a decision boundary that can split the input data into the desired classes. Usually, the feature vector that represents the images requires a non-linear boundary for a precise separation. This implies in the use of classifiers capable of finding a non-linear separation model. A vast collection of classifiers is available in the literature, covering different types of applications and needs. The most common include Artificial Neural Networks (ANN), Support Vector Machines (SVM), decision trees and random forests. Each one has its pros and cons, regarding application, type of input data, performance, and other specifications. Choosing the most suitable classifier requires a greater amount of expertise by the researcher or an empirical trial-and-error procedure.

Many papers produced by the computer vision and machine learning community, related to image classification, are based on three steps: pre-processing, feature extraction and classification, as shown in the block diagram in Figure 1. Some authors merge pre-processing and feature extraction into a single step (DUDA et al., 2001; BISHOP, 2006).

The pre-processing step seeks to reduce the variability of the input data, as well as avoiding noise, where a series of image processing techniques are employed. These operations

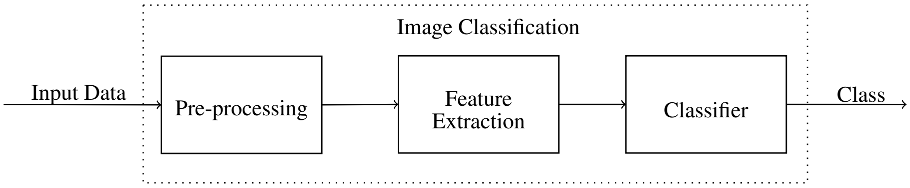

Figure 1: Steps of a traditional image classification process.

include several types of filtering, normalization, contrast and brightness adjustments and segmentation. A discussion of pre-processing methods lies outside the scope of this work. For readers who wish more information on this topic, the work of (GONZALEZ; WOODS, 2006) is recommended.

2.1.2 FEATURE EXTRACTORS

Even with a pre-processed image, dealing directly with the pixel intensities usually tends to produce poor results in the classification. The common approach here is to apply a transformation in the input image, seeking to make it more informative, in other words, easier to be classified. Engineering methods to find useful features that have high discriminatory information and are fast to compute have received considerable attention in the computer vision community. The design should take care not to discard important data during the transformation (DUDA et al., 2001).

Usually, feature extraction methods are based on a hand-designed process, where a field expert analyses the problem and attempts to find an interesting way to extract or transform the input data. This can be done by extracting: visual features (such as edges, textures and contours); statistical features (histograms and statistical moments); transform coefficients (Fourier descriptors); and other algebraic features, using algebraic operations as a way of transforming the data (TUYTELAARS; MIKOLAJCZYK, 2008; PENATTI et al., 2012).

Several feature extraction methods are available in the literature, and these should be chosen depending on the context of the application. The most popular include the Scale-Invariant Feature Transform (SIFT) (LOWE, 2004), Gradient Location and Orientation Histogram (GLOH) (MIKOLAJCZYK et al., 2005), Binary Robust Invariant Scalable Keypoints (BRISK) (LEUTENEGGER et al., 2011), Binary Robust Independent Elementary Features (BRIEF) (CALONDER et al., 2012), and many others. A discussion of these features and their application are beyond the scope of this work. The reader is referred to (TUYTELAARS; MIKOLAJCZYK, 2008; PENATTI et al., 2012). In this work, three methods were chosen to be used during the experimentation process. These methods were picked since they can extracted complementary features from images.

One of the most popular image descriptor, specially for person detection, is the Histogram of Oriented Gradient Histogram of Oriented Gradient (HOG). Proposed by Dalal e Triggs (2005), it works dividing the input image into small overlapped sub-images, called cells. Within each cell, an histogram of the orientation of the gradients is accumulated. Then, each histogram contributes to the formation of a global histogram, which is used as the feature vector to describe the image. This method allows capturing gradient structure, that is very characteristic of local shape. Also it is relatively invariant to local geometric transformations, such as small rotations and translations.

When dealing with image matching, in general, local invariant features are used to provide a efficient representation. Bay et al. (2008) proposed the Speeded Up Robust Features (SURF), which can be viewed as an extension/improvement of the SIFT method. The algorithm works by find image interest points using the determinant of Hessian matrix. The major interest points in scale space are selected based on a non-maximal suppression. The feature direction is also computed using Haar transformations, seeking to generate rotationally invariant features. Finally, the features vectors are computed using a square descriptor window centered on each interest point.

Local Binary Pattern (LBP) is a very simple but efficient texture descriptor proposed by (OJALA et al., 2002). It works by dividing the input image into cells. Each pixel inside a cell is compared against its eight neighbours, composing a 8-d binary code. Value 0 indicates that the center pixel is greater than the neighbour, and value 1 indicates otherwise. Over the cell, a histogram of codes is computed. Finally, all histograms are concatenated forming the image descriptor.

The existence of so many descriptors can be realized trough the complexity of extracting and analysing the contents of an image, requiring many operations to be performed with the maximum efficiency. The choice of the right feature extractor depends on the user's knowledge of both application and extractor properties. Therefore, it would be interesting if the feature extraction process did not depend on a previously projected extractor, but rather built based on the input data. In this work, this is the approach employed as a possible solution.

2.1.3 LEARNING

The main objective of feature extraction is to transform input data into a more separable space, making easier to obtain an accurate classification performance. This process of classification, in general, employs a learning procedure. Three common types of learning strategies can be found in literature: supervised, semi-supervised and unsupervised. All three types are concerned in separating the input data according to some relationship.

Let $X = (\vec{x}^1, \vec{x}^2, ..., \vec{x}^n)$ be a set of n data samples or points in $\mathbb{R}^d$. Typically, it is assumed that the points are independently and identically distributed by a common distribution. In unsupervised learning, nothing else is known about the data, and its goal is to find some kind of interesting structure from data. In general, this structure could be defined in terms of a density estimation about the distribution, or other kind of metrics, such as Euclidean distance, in the case of clustering algorithms. Another interesting application of unsupervised learning is related to dimensionality reduction, where it is wanted to find a compact and meaningful version of the original data. Methods such as Principal Component Analysis (PCA) and Liner Discriminant Analysis (LDA) are two examples of dimensionality reduction procedures. The main advantage of unsupervised learning is its capability to deal with unlabelled data, which are cheap and largely available (DUDA et al., 2001; BISHOP, 2006).

If the input data is presented in the form of a pair (x^i, y^i), where $y^i \in Y$ and Y is a set of predefined labels, there is an initial knowledge about the data. The goal of the learning method is to use this information and find a mapping function $f_w(x) = \hat{y}$, namely classify x according to a set of classes, where $\hat{y}$ represents the predicted class. This is the case of supervised learning. Usually, the input dataset X is divided into two subsets, one for training and another for testing. A given algorithm should be adjusted using the training set and, later, check its prediction performance using the test set. This is the usual method for determining the generalization power of the model found by the learning algorithm. Several methods could be used to adjust this function, such as ANN, Convolutional Neural Networks (CNN), SVM, decision trees, random forests, and others (DUDA et al., 2001; BISHOP, 2006).

Between supervised and unsupervised learning, there is semi-supervised learning, where just a portion of the input data is given with supervision, i.e., with defined labels, and the remaining is unlabelled. The goal here is to learn from the labelled data and extend the knowledge to the unlabelled data. Discussion related to semi-supervised learning is beyond the scope of this work. Interested readers should see (CHAPELLE et al., 2010) for further information.

2.2 CONTENT-BASED VIDEO RETRIEVAL

An application of computer vision and pattern recognition can be the content-based retrieval, which deals with search of multimedia artifacts using their attributes, which can be extracted automatically or semi-automatically. The search terms could be low-level features (histograms, edges, shape, textures) and/or high-level concepts (person, car, children, dog, sunset). Content-based video retrieval (CBVR) derives naturally from content-based image retrieval (CBIR). Both seek to filter the image/video databases in order to respond to the user queries (SHANDILYA; SINGHAI, 2010; PATEL; MESHRAM, 2012).

In general, textual descriptions attached to videos (metadata) are used as the main keys for searching. This seems to be a natural way to do the search, since written/spoken language is the main form of human communication. A major drawback related to this method is the requirement of manual annotation, which is a tedious and inefficient work. Besides, different people could diverge about the concepts of a scene, leading to ambiguous annotations. Another related problem is that, after a certain time, the user may not remember which terms were used to describe images or videos.

Seeking to alleviate the burden of manual annotation, which demands humans to analyse and describe the contents of each video, computational methods were developed. These methods should analyze the video contents in an automatic way, doing recognition and annotation. This demands intelligent methods capable of interpreting the visual contents of a video as close as possible to humans. Recently, many research teams across the world have devoted efforts to improve the quality of automatic video analysis.

Among the methods reported in the literature, to deal with content-based retrieval, there are: ontology construction (ZHA et al., 2012; BAI et al., 2011; EROZEL et al., 2008); graph and partial differential equations (TANG et al., 2009); hidden Markov chains (ZHANG et al., 2009); extreme learning machines (LU et al., 2013); self-organizing maps (KOSKELA et al., 2009); and others. It is important to notice that these methods are developed for specific domains.

2.3 SOFT BIOMETRICS

A possible application of a CBVR system is related to video surveillance, where video cameras are used to monitor any relevant area. The key content of such videos is the people who could appear. In other words, the system is used to monitor people activity. With effect, a person could be seen as a set of highly defined features, called biometrics.

Biometrics are commonly used by humans to identify other known individuals (JAIN et al., 2000). The acquisition and recognition of biometrics are done naturally and effortless by humans. Inspired by that, it would be interesting and useful to develop an automatic system able of extracting and identifying human biometrics and find them in a monitored environment.

A biometric recognition system (BRS) associates unequivocally an individual to a known identity using his/her body features or his/her behaviour. This kind of system could be understood as a pattern recognition system based on predetermined body and/or behavioral features (DELAC; GRGIC, 2004).

There are several types of biometrics varying, for instance, from physiological (such as fingerprint, DNA, retina) to behavioral (such as voice and gait). In most cases, the acquisition of such kind of biometrics requires the cooperation of the target person (KIM et al., 2012). On the other hand, another kind of biometric data that can be extracted from images/videos are the soft biometrics. This kind of information is related to human attributes, such as clothing, gender, and ethnicity, for example. By using these attributes, it is not possible to identify a person unambiguously. However, it is possible to reduce the range of possibilities when searching for a specific individual. The advantage of the soft biometrics is that the cooperation of the subject is not needed for data acquisition (REID et al., 2013), such that it fits perfectly surveillance purposes.

This is a very hard problem in several ways. There is a high variance in the way that people are dressed; people can be in many different poses. They also can be partially occluded by other objects or other individuals. Finally, the background which contain the target subject can be complex and different from scene to scene, imposing more difficulties to the identification of a person.

An example of describing an image of a person using soft biometrics is shown in Figure 2. In that image, attributes such as gender, upper clothes and lower clothes are used to obtain a high level person description. It is important to notice that, although is relatively easy to a human achieve this kind of interpretation, it is very complex to be done automatically by computers.

2.4 CONVOLUTIONAL NEURAL NETWORKS

As mentioned before, traditional pattern recognition methods are based on two main steps, feature extraction and classification. According to Section 2.1.1, feature extraction is

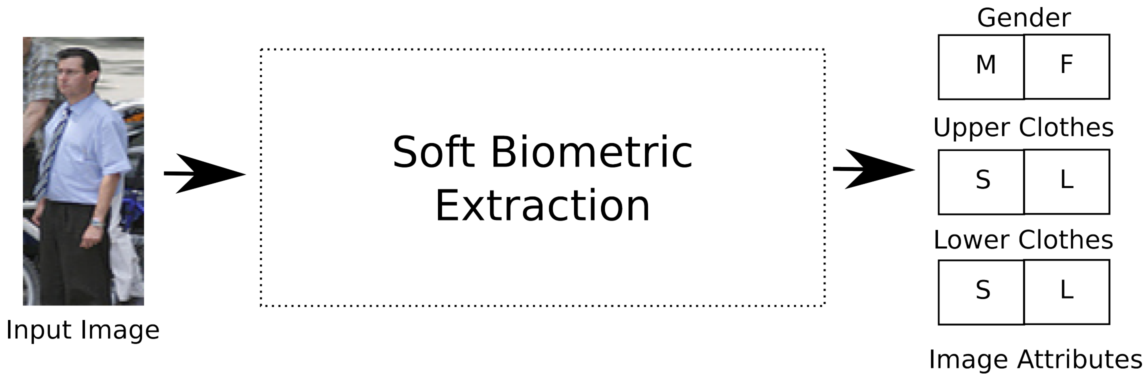

Figure 2: Example of describing an image of a person using soft biometric attributes.

the transformation of the input data into a more meaningful representation, helping to improve the classification performance. Finding a suitable representation demands human labour, who should deeply analyse the data and the application domain, trying to devise that transformation.

An alternative to this process would be to learn better representations automatically from the input data. CNN is a method that has the capability of learning both the feature extractor and the classifier during its training procedure. This section review both conventional and convolutional neural networks.

2.4.1 FEED-FORWARD NEURAL NETWORKS

A neural network, as the name suggests, attempts to mimic in a very simplified form the dynamics of the human brain, using a set of processing units called neurons, and their interconnections, called synapses (BISHOP, 2006).

The simplest neural network is composed by just one unit, known as step neuron. It has a series of inputs $(x_0, x_1, ..., x_d)$, where x_0 represents a bias, whose value is assumed to be always 1, and one output in the form of $y = \varphi(z)$. For each input a connection weight w_i is defined. The function φ is a nonlinear activation function, usually the step function, sigmoid (Equation 1) or the hyperbolic tangent (Equation 2), and z is a weighted linear combination of its inputs. A general schematic of a neuron is shown in Figure 3.

$$sigm(z) = \frac{1}{1+e^{z}} \tag{1}$$

$$tanh(z) = \frac{e^{z}-e^{-z}}{e^{z}+e^{z}} \tag{2}$$

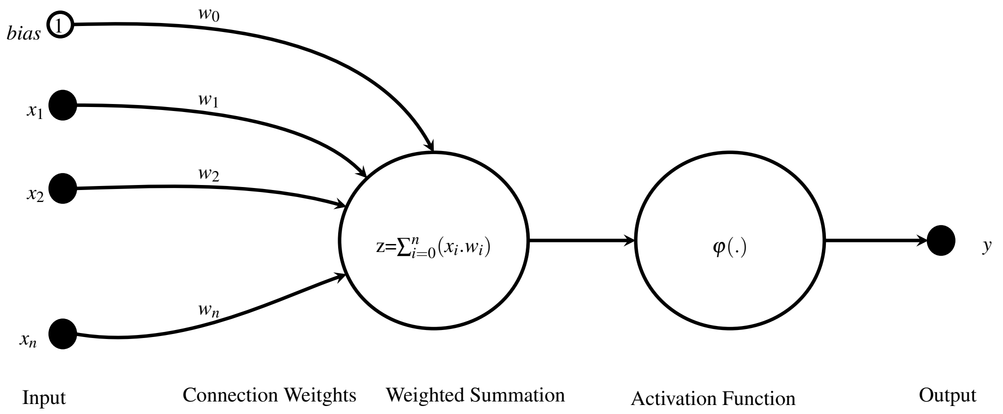

Figure 3: Example of a neuron, showing the inputs, the weighted sum, the activation function and the output.

Given a dataset X in the form of (x^i, y^i), where x is the input data and y is the label. Notice that the superscripts denotes each sample in X. The neuron is aimed at finding a nonlinear function $f(x)$, which maps the input x^i to its label y^i, for all data samples. Since $f(x)$ is parametrized by the weigths, i.e., $f(x,w)$, the mapping is done by adjusting the connection weights by an iterative learning process. The computation capacity of a single neuron is limited, being capable of dealing only with small and simple problems.

The solution to real problems requires a more powerful method to find a map, able to deal with non-linearities. This could be achieved with something like a human brain, gathering several single neurons into a well defined multilayer structured architecture. It can be represented as a directed graph of neurons, organized into layers, as shown in Figure 4. The left-most layer is called input layer, where the input data are introduced into the network. In this example, the data are represented as a 4-dimensional vector, but the size may vary according to the application and, especially, to the feature extraction method. The layers in the middle are denoted as hidden, in the sense that they do not have external inputs or outputs, and all their neurons are connected to the neurons in the previous layer. The number of neurons in each layer is a design choice and may vary, especially in accordance with the complexity of the input data. The right-most layer is the output layer, which is responsible for producing the network results; and the number of units is related to the expected output (e.g. number of classes). The architecture details, such as the number of neurons in each layer and the number of layers should be determined concerning the application.

Formally, the i^{th} neuron within layer l has the form as shown in Equation 3, where i represents the unit. All units in all layers, except the input layer, operate using this formulation. This process is called feed-forward, since the data flow is propagated forward throughout the network.

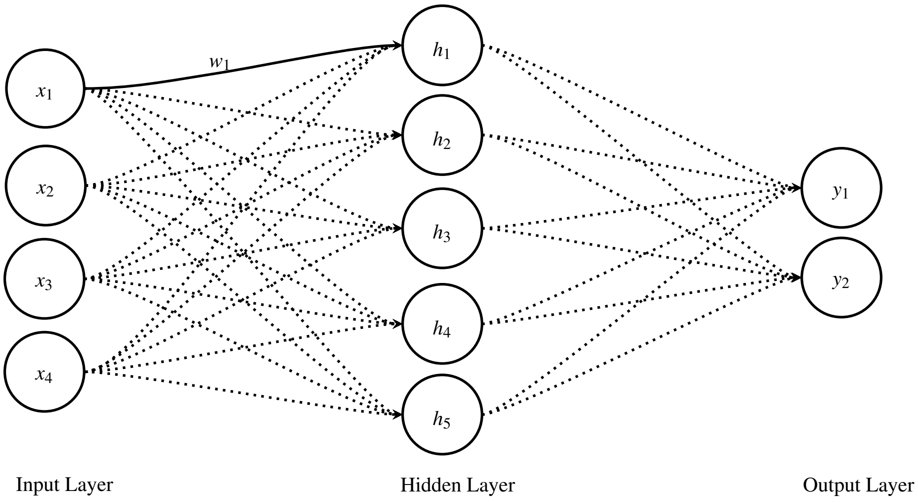

Figure 4: Example of an multilayer network.

$$y_{i,l} = \varphi(\sum_{i=1}^{d} w_{i,l-1}x_{i,l-1}), \forall l \in L, \forall y \in l \tag{3}$$

Figure 4 associated to Equation 3, shows that the output of the first hidden layer of an ANN will be a new representation of the input data concerning to the linear combination with the connection weights. The weight between x_1 and h_1 is w_1 (solid line). All the remaining weights (dashed lines) are omitted for better visualization. The outputs of the next layers will lead to a higher representation of the input data. The output of the last layer will be the mapping between the input data and the desired classes.

Choosing appropriate weight values enables to find arbitrary decisions boundaries, leading to the desired classification. Theoretically, a three layer neural network could approximate any function, given the necessary input samples and sufficient hidden units (BISHOP, 2006).

The common procedure for adjusting the weights of the network is by an interactive process, where an example is forwarded through the network until an output is obtained. A cost function Q is used to calculate the difference between the network output and the expected result. In general, Q is defined as Equation 4, and is called mean squared error.

$$Q(x,w) = \frac{1}{2n}\sum_{x}(f(x,w) - y)^2 \tag{4}$$

Here, N represents the total number of training samples, $f(x,w)$ is the output of the network and y is the desired response. It is possible to see, that the output of the network depends directly on x and w. When the network output becomes close to the desired output, the value of $Q(x,w)$ must become close to 0. In this way, the weights must be adjusted to minimize the value of $Q(x,w)$.

A common way to proceed this minimization process is to use the gradient descent algorithm (GD). It is based on the partial derivative $\frac{\partial Q}{\partial w}$ of the cost function with respect to the weights in the network. This derivative allows to understand how the cost function changes when the connection weights are changed. Based on this, GD makes small steps in the opposite direction of the gradient, leading to the consequent minimization of Q. Equation 5 shows how to update the weights using GD (BISHOP, 2006).

$$w^t = w^{t-1} - \eta \frac{\partial Q}{\partial w}, \tag{5}$$

where t represents an iteration in the training process, called epoch, η represents the size of the GD step, and is commonly named as learning rate. Choosing the appropriate value for the learning rate is essential to GD convergence.

It is important to notice that the derivatives must be calculated for every training sample, and then be averaged before the weight update. This is known as *batch* gradient descent. When the training set is big, this process can take a long time, and the learning procedure could be slow.

Instead of looking at the whole training set, a modification in the procedure updates the weights after random sampling. This is done using the entire dataset. This is called *stochastic* gradient descent. The calculation in this case is faster than the GD, but the convergence to the minimum is somewhat random.

Between both there is the *mini-batch* stochastic gradient descent, in which a small group of training samples are randomly selected and used to update the weights. Again, the entire dataset is employed. In general, the size of a mini-batch is approximate 100 samples. This approach is a common choice in the literature.

The process of using the derivative to adjust the weights is called delta learning rule. A detailed discussion of the backpropagation algorithm can be found in (DUDA et al., 2001; BISHOP, 2006).

2.4.2 CONVOLUTIONAL NEURAL NETWORKS

The common process for automatic classification, as stated above, is mainly based on two stages: a feature extractor and a classifier. Both are chosen when modelling the solution for the classification problem. A hand-designed feature extractor requires the presence of a human expert to find the most suitable data manipulation for achieving good classification performance. To overcome this, the features could be automatically learnt from the input data. Therefore, the extractor is adjusted to fit the requirements for a given classification task automatically.

A method conceived to address these characteristics is the CNN, which is a special type of feed-forward neural network, in whose architecture various hidden layers are employed, developed to deal with 2D input data, such as images (LECUN et al., 1998).

The main characteristic of this method is the ability to learn a hierarchy of features, allowing a more abstract representation of the input data. Thus, raw images can be introduced into the network without any kind of preprocessing or feature extraction. Therefore a trained CNN is an end-to-end classifier. Another advantage is that the feature extractors are constructed based on the data used for training, the network learns to manipulate the input data as a way of performing the classification (LECUN et al., 2010). The architecture of a CNN is designed to incorporate two modules, as shown in Figure 5: the feature extractor and the classifier.

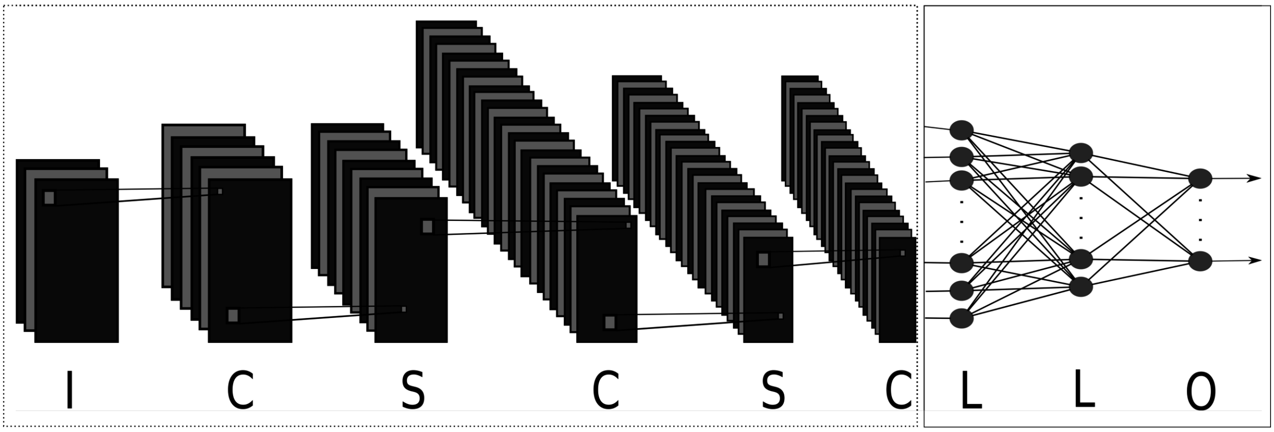

Figure 5: An example of a CNN architecture. The dashed rectangle denotes the feature extractor layers. The continuous rectangle denotes the classifier layers. The (I) layer represents the raw input image. The (C) layers represent convolution operations. The (S) denotes sub-sampling operations. The linear layers are represented by (L), and (O) represents the output layer.

The feature extractor module is composed by multiple stages, and its inputs and outputs are called feature maps. Each stage has three different layers: convolution, non-linear filtering and sub-sampling.

The convolution layer is responsible for storing the trainable weights, which allows the customization of the feature extractor. The input of a convolution layer is a two-dimension array for gray level images, or a three-dimensional array for color images, with m 2D feature maps of size $w \times h$. Each feature map is denoted by m_i^l, where l denotes the layer and i the map at that layer. A $k_1 \times k_2$ trainable convolution kernel is used to connect the input feature maps

m^{l-1} to the output feature map m_i^l. This connection is performed by a 2D discrete convolution operator using the kernel as the convolution mask.

The trainable kernel is called a receptive field, and the idea is to restrict connections from a given neuron to a small neighbourhood in the immediately preceding layer, thus forming a tiny filter for feature extraction. This restriction was inspired by the complex arrangement of cells within the visual cortex, described by (FUKUSHIMA, 1980). The use of a 2D kernel allows capturing the spacial relation between adjacent pixels in images.

Since a common feature may be located at different positions in the input image, it is worth performing the extraction in the entire image. A weight sharing mechanism allows neurons of the same feature map to have the same weights associated to different receptive fields. This allows the extraction of features irrespective of the position and reduces the number of parameters to be trained (LECUN et al., 2010).

The non-linear filtering consists in the application of a non-linear activation function to all components of each feature map. The reason for that is, as the same as for a regular neural network, force the network to learn a better representation for the input data, since some information is lost during the filtering process. The most common choice is the hyperbolic tangent function $tanh(m_i^l)$ (LECUN et al., 2010).

The sub-sampling mechanism involves the reduction of a feature map by a constant factor, providing the network robustness to small distortions and translations. This reduction is performed over a $f_1 \times f_2$ neighbourhood from the previous layer using a certain operation, such as addition, average and maximum, commonly called maxpooling. For example, if an input feature map has 28×28 components, and the neighbourhood size is 4×4, the output feature map will have 7×7 components, being reduced by a factor of 4 (LECUN et al., 1998; BOUREAU et al., 2010).

Some architectures use a manoeuvre to give the network some translation invariance and reduce the computation effort. Instead of applying a convolution kernel or sub-sampling operation to every possible location on the input map, leading to an overlapping region, a stride is used. This means that a step, in both horizontal and vertical directions, is used.

The last module of a CNN consists in the classifier itself. Its inputs are the outputs from the last feature maps. In general, a common linear feed-forward classifier is employed. The amount of input neurons, hidden layers and output neurons is problem-dependent.

To adjust the weights of the network, a variation of the backpropagation learning algorithm proposed by (LECUN et al., 1998) is used. Thus, the layers are trained in a supervised way, seeking to reduce the error between the predicted and the expected results, generally using the stochastic gradient descent (SGD) algorithm.

2.4.3 LENET – 5

The concepts discussed above were first applied to document recognition, more specifically, to handwritten character recognition (LECUN et al., 1998). In that paper, an architecture for a CNN, called LeNet-5, was proposed and tested. This network became the ground basis for further research about CNN.

The network, as shown in Figure 6, comprises 7 layers, not including the input. The input to the CNN is a 32×32 black and white image. Layer $C1$ is a convolutional layer with $6 \times 28 \times 28$ feature maps generated by a 5×5 trainable kernel. Layer $S2$ reduces the $C1$ feature maps to $6 \times 14 \times 14$, using a 2×2 neighbourhood. The four inputs to a unit in $S2$ are added, then multiplied by a trainable coefficient, and added to a trainable bias. The result is squashed by a sigmoid function. Layer $C3$ is a convolutional layer with 16 feature maps of size 10×10, generated by 5×5 trainable kernels. Unlike $C1$, the $C3$ layer has a special form of connection between its feature maps and the $S2$ maps. This was proposed to reduce the number of network parameters and increase its capacity. Layer $S4$ reduces the 16 feature maps from $C3$ to 5×5, using the same operation as $S2$. Layer $C5$ processes the 16 feature maps from $S4$ using 5×5 trainable kernels. Because the $S4$ maps have a size of 5×5, the output of $C5$ is 120 feature maps of size 1×1. The next layer, $F6$, is a linear, or fully-connected layer, and acts as a layer in an MLP network, computing the linear combination of its inputs with trainable weights. Output layer $F7$ is composed of Euclidean Radial Basis Function units (RBF), one for each class, with 84 inputs each. The whole LeNet-5 has a total number of 14,000 trainable parameters (LECUN et al., 1998).

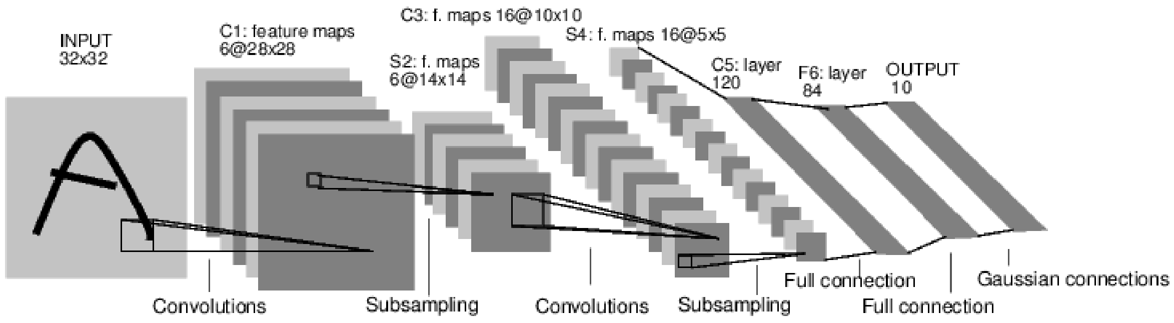

Figure 6: LeNet-5 network architecture, which was proposed to classify handwritten digits. Source: (LECUN et al., 1998)

The network was trained and tested using the MNIST handwritten digits database. The training procedure used 60,000 samples while the test set was composed of 10,000 images. After 20 passes through the training database, the network achieved an error of 0.95% on the test set. When an expanded version of the training database was employed, the test error dropped to 0.8%. The expansion was created using distortions in the images, such as horizontal and vertical translations, scaling, squeezing and horizontal shearing.

2.4.4 USAGE OF CNN

Despite the fact that the first CNN was proposed a long time ago, it attracted much attention in recent years. The community claims that two major facts reactivated research on this topic: the popularization of the use of powerful Graphical Processing Units (GPUs), along with the tools to develop algorithms using this technology, such as the Nvidia's CUDA framework; and the availability of huge annotated image databases, such as the ImageNET (DENG et al., 2009). This has led to numerous methodological proposals for different issues related to CNN, including the use of unsupervised pre-training, activation units, training procedures and architectures.

(KRIZHEVSKY et al., 2012) proposed a CNN architecture as a solution to the ImageNet Large Scale Visual Recognition Challenge 2010 (ILSVRC2010), where 1.2 million high-resolution images had to be classified into 1000 different classes. The network was composed of eight learned layers: five convolutional and three fully-connected, as shown in Figure 7. The first layer transforms a 224×244 RGB image into 96 feature maps using kernels with $11 \times 11 \times 3$ components and a 4-pixel distance (stride) between the receptive fields. The second layer expands the output of layer one to 256 feature maps, where the kernel has $5 \times 5 \times 48$ components. The third layer has 384 feature maps generated by kernels of size $3 \times 3 \times 256$. The fourth layer has 384 maps generated by kernels with a size of $3 \times 3 \times 192$. The fifth layer has 384 maps generated by kernels of size $3 \times 3 \times 192$. The fully-connected layers have 4096 units each. The max-pooling and a local normalization operation are employed only after the first and second layer. This architecture has 650,000 processing units and 60 million trainable parameters.

The training process of such a large network demands high computation power. A multi-GPU scheme was employed to adjust the network parameters in between five and six days. This optimization scheme unveils an affordable platform for training large CNNs.

During the training phase, the units in the fully-connected hidden layers of the CNN learn to detect certain regularities that enable it to make correct predictions with respect to a

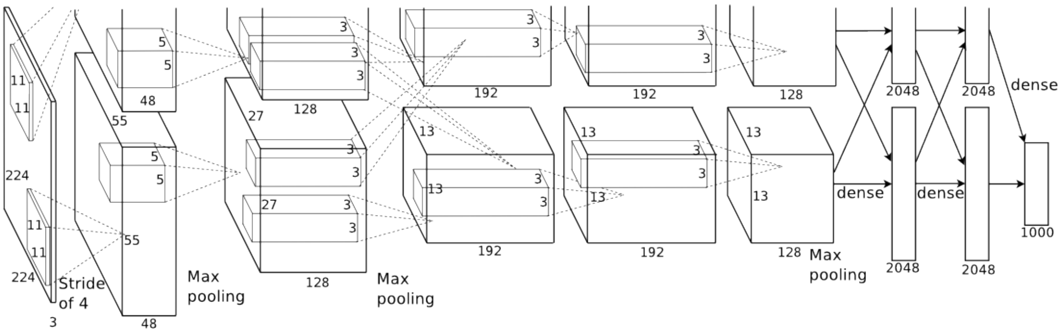

Figure 7: CNN architecture proposed by (KRIZHEVSKY et al., 2012) as solution to ILSVRC2010. Source: Adapted from (KRIZHEVSKY et al., 2012)

given input. If the number of hidden units is large, the units can learn very well how to classify the entire training data set. But when the network is used to predict the class of new input data, the results may be worst. This phenomenon is called overfitting. A possible way of reducing the overfitting, as proposed by (SRIVASTAVA et al., 2014), is to randomly turn off hidden units with a probability of 1/2 during the training phase.

The algorithm consists in, for each training example and training epoch, randomly chosen hidden units are selected and temporary deleted. The other remaining weights are normally trained using backpropagation (BALDI; SADOWSKI, 2014). Figure 2.4.4 shows a representation for this process, where light circles and dashed lines represent turned off units. This procedure is called dropout learning.

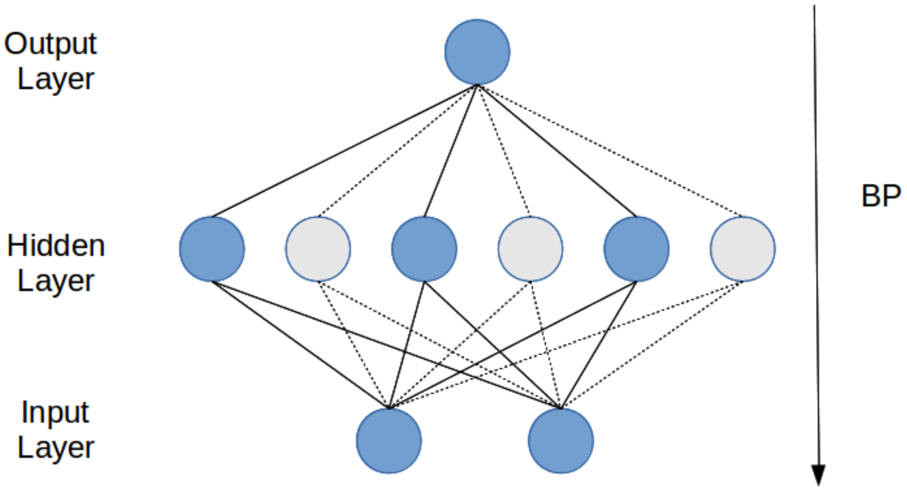

Figure 8: Representation of how dropout works. Source: Adapted from (BALDI; SADOWSKI, 2014)

This procedure can produce two main advantages in the neural network behaviour. First, it prevents the co-adaptation of units, where the units rely on the presence of others to make the prediction. Second, multiple different networks are trained at the same time, enabling model averaging. The use of the dropout strategy has become a standard in recent CNN architectures, and it has been reported as helpful to improve the network performance (SRIVASTAVA et al., 2014).

Different types of non-linearity can be used in the CNN architecture. The model proposed by (LECUN et al., 1988) was based on the hyperbolic tangent ($tanh$). Also, (JARRETT et al., 2009) found that the use of a non-linearity in the form of $relu(x) = max(0,x)$, called Rectified Linear Units (ReLU), produces better results. In (KRIZHEVSKY et al., 2012), it was stated that convolutional neural networks with ReLU are several times faster than those with $tanh$, achieving similar error rates with less training epochs. Since then, the majority of papers have used the ReLU as the non-linearity.

Great progress has been made in recent years in the field of pattern recognition, with problems like character recognition, achieving highly correct classification rates. But achieving similar visual capacity of a human using a computer poses a great challenge to the machine learning and computer vision community.

Several researchers agree that CNN has showed to be the way to achieve the desired visual capacity. Many information technology companies are investing a great deal to develop solutions for pattern recognition problems using CNN.

The results of these methods are promising, but some doubts remains as to how they are obtained. Therefore, in the coming years, a great deal of research will be conducted in an attempt to comprehend the learning mechanisms of CNN.

As discussed above, the learning process of CNN requires both large amounts of labelled data and computation power. The latter seems not to be such a big problem, since the use of GPU and large computer clusters enables the achievement of computation demanded. Annotated data is still the bottleneck of this method. Therefore, an important issue, also addressed in this work, is the use of unsupervised learning strategies for facilitate the annotation process of large image datasets.

2.5 DIFFERENTIAL EVOLUTION

As a neural network is inspired by human brain functionality, other areas of computing were inspired by nature to develop computer science methods for solving problems. This paradigm for the development of algorithms is called natural computing, or biologically inspired computing.

Bio-inspired methods can be roughly broken down into two basic lines of approach: computing inspired by nature and simulation or emulation of natural phenomena. In the first category, there are methods based on the observation of the behaviour of organisms and communities, thus generating inspiration for the development of systems aimed at solving complex problems. Evolutionary Algorithms (AE), neurocomputation, artificial immune systems and swarm intelligence fall into this class . The second line is the simulation of processes and events that occur in nature virtually trying to recreate the behaviour of organisms, thereby allowing better understanding of nature. One result of this second line is artificial creatures and life systems, that is, systems that mimic life as humans know. (CHEN, 2007). In this work, we focus our attention to evolutionary computation as an efficient tool to solve a complex problem.

In this work, the Differential Evolution (DE) proposed by (STORN; PRICE, 1997) was chosen as the optimization method, since it is robustness to solve complex problems. DE was originally designed for the optimization of continuous multidimensional mathematical functions. An important feature of DE is its independence of the function gradient, i.e., the function being optimized does not need to be differentiable. This feature allows the use of DE in a vast set of problems, from bioinformatics (KALEGARI; LOPES, 2010), combinatorial problems (KRAUSE; LOPES, 2013) to computer vision (CUEVAS et al., 2013). A survey of DE details and applications can be found in (NERI; TIRRONEN, 2010) and (DAS; SUGANTHAN, 2011).

DE is a population based approach in which a population of NP d-dimensional vectors ($\vec{x}_i^{(G)}$, $i = 1, 2, ..., NP$), referred as individuals, are manipulated during a number of iterations (denoted by G). The initial values for each candidate solution (individual) are chosen randomly and should sample the entire solutions space. Similarly to other evolutionary computation approaches, DE uses three basic mechanisms during the search: mutation, crossover (recombination) and selection (PRICE et al., 2005).

Mutation generates small perturbations in each individual, by means of a weighted difference between two vectors to a third one. Equation 6 shows the mutation operation, where $\vec{v}_i^{(G+1)}$ is the i^{th} mutant vector for the next iteration ($G+1$). Indices $r_1, r_2, r_3 \in 1, 2, ..., NP$, are mutually different random integers, i.e., $i \neq r_1 \neq r_2 \neq r_3$, such that always different individuals of the population are chosen for the operation. $F \in (0, 2]$ is a constant factor that controls the influence of the difference.

$$\vec{v}_i^{(G+1)} = \vec{x}_{r_1}^{(G)} + F \cdot (\vec{x}_{r_2}^{(G)} - \vec{x}_{r_3}^{(G)}) \tag{6}$$

Seeking to increase the population diversity, a crossover (recombination) operator is used. Trial individuals are generated using the rule described by Equation (7), where $randb(j) \in [0,1]$ is the j^{th} uniform random number generated; $CR \in [0,1]$ is the crossover probability; $rnbr(i) \in 1,2,...,d$ is a randomly chosen index, which ensures that at least one parameter of the new individual came from the mutant vector.

$$u_{ji}^{(G+1)} = \begin{cases} v_{ji}^{(G+1)} & \text{if } (randb(j) \leq CR)) \text{ or } j = rnbr(i) \\ x_{ji}^{(G)} & \text{if } (randb(j) > CR)) \text{ and } j \neq rnbr(i) \end{cases} \quad j = 1,2,...,d \tag{7}$$

After the crossover, the just-created individual can be selected or not to be used in the next iteration. The selection mechanism compares the quality of $u_i^{(G+1)}$ against x_i^G, using a simple greedy criterion, as shown in Equation(8).

$$\vec{x}_i^{(G+1)} = \begin{cases} \vec{u}_i^{(G+1)} & \text{if } (f(\vec{u}_i^{(G+1)}) < f(\vec{x}_i^{(G)}))) \\ \vec{x}_i^{(G)} & \text{otherwise} \end{cases} \tag{8}$$

The application of the two operators, mutation and crossover, followed by the selection procedure, is probabilistically done for all elements of the population, and the whole procedure is repeated for several iterations until a stopping criterion is met. For most cases, this criteria is the maximum number of iterations is reached or, else, a predefined achieved error level. Another criterion could be the mean fitness of the population, since if it is close to the minimum fitness means that the individuals have converged to a local minimum. Algorithm 1 shows a simplified pseudocode for the standard DE (PRICE et al., 2005).

The result of execution of DE is a vector with the fittest solution found during the search process. As stated, there is no guarantee that this is the optimal solution to the problem.

A great advantage in choosing DE as optimization strategy is its ability to find better solutions than other evolutionary strategies. Additionally, it is relatively easy to implement and apply to a variety of real valued problems (DAS; SUGANTHAN, 2011).

2.6 CLUSTERING

The analysis and classification of large amounts of data can be performed by means of three approaches: supervised, semi-supervised and unsupervised. The main difference between these methods is related to the previous knowledge that the user has about the data.

```
Input:$d, NP, F, CR$ Output:Best $\vec{x}_i$ G=0;
Generate initial population $\vec{x}_i^{(G)}$ $i = 1,2,...,NP$;
Evaluate the cost function $f(\vec{x}_i^{(G)})$ $i = 1,2,...,NP$;
while not stopping criterion do
    // Execute mutation
    for i=1 to NP do
        $\vec{v}_i G+1 = Mutation(\vec{x}_i^{(G)})$ // Following Equation(6)
    end
    // Execute crossover, generating the trial vectors
    for i=1 to NP do
        $\vec{u}_i^{G+1} = Crossover(\vec{x}_i^{(G)}, v_i^{(G+1)})$ // Following Equation(7)
    end
    // Execute selection, using the result of the cost
       function
    for i=1 to NP do
        if $fitness(\vec{u}_i^{(G+1)}) < fitness(\vec{x}_i^{(G)})$ then
            $\vec{x}_i^{(G+1)} = \vec{u}_i^{(G+1)}$
        else
            $\vec{x}_i^{(G+1)} = \vec{x}_i^{(G)}$
        end
    end
    G=G+1
end
```

Algorithm 1: Pseudocode of the standard Differential Evolution algorithm.

By definition, supervised learning is a machine learning task in which a mapping function is inferred by repeatedly examining annotated training data (MOHRI et al., 2012). Annotated data (or labelled data) are those in which each sample is associated to a well-defined class label. Semi-supervised methods need that the training data be partially labelled, and they try to extend the known knowledge to the unlabelled data. The unsupervised methods do not require any previous information about the data (DUDA et al., 2001).

For the particular case of image and video, annotated data implies a previous human classification of the semantic contents. Due to the high cost and complexity of this procedure the amount of unlabelled data is many times larger than the labelled one. This fact motivates the development of unsupervised methods in this area.

The main idea behind unsupervised methods is to analyse and group the input data based on one or more features inherent to the data, without any assumption about how it is partitioned or its nature. This means that methods will explore the data aiming at finding some kind of pattern that allow the formation of two or more distinct clusters. Clustering is, therefore, a basic and widely used method for unsupervised learning that allows to discover some structure in unlabelled data.

A well-known method for clustering data is known as k-means or Lloyd's algorithm (LLOYD, 1982). This algorithm is very popular, since it is simple to implement, runs fast, and it has just one parameter to be set: the number of clusters. Notwithstanding, this is the main drawback of the method, since the user may not know, a priori, the number of clusters that the data is naturally grouped into.

The goal of k-means is to partition a set of n d-dimensional samples, $\vec{x_i} \in \mathbb{R}^d,\ i = 1, 2, ...n$, into k groups, based in some similarity present in the data. This is a centroid-based method for clustering. Therefore, it is intended to find a set of centroids $\vec{C_j} \in \mathbb{R}^d,\ j = 1, ..., k$ and a designation set $A_i,\ i = 1, 2, ..., n$ for each input sample $\vec{x_i}$, in such a way that the distance between each sample and its centroid should be the smallest. Formally, C and A are found by minimizing Equation (9). The similarity metric used for computing the distance between centroids and data points is the usual Euclidean distance in the d-dimensional space.

$$\underset{C,A}{\text{minimize}} \sum_{i}^{n} \sqrt{\sum^{d} C_{(A_{(x_i)})} - x_i} \tag{9}$$

The minimization process can be split into two sub-processes, shown by Equations (10) and (11), which are sequentially repeated until convergence. The stopping criterion could be a maximum number of iterations, error threshold or stabilization of $\vec{C}$, in other words, no changes on centroids.

$$A_i := \underset{k}{\text{argmin}} |C_{(k)} - x_i|_2, \forall_i, i = 1...n \tag{10}$$

$$C_k := \frac{\sum_{j=1}^{|A_k|} x_j}{|A_k|}, \tag{11}$$

where $|A_k|$ is the number of samples assigned to cluster k. Notice that a centroid C_k is obtained by computing the mean of all elements assigned to cluster k. At the end of the k-means execution, the centroids can be interpreted as a compact representation of the input data. A pseudo-code of k-means is shown in Algorithm 2.

```
Input:set of $\vec{x}, k$ Output:$c_1, ..., c_k$ create a randomly set of $k$ vectors $\vec{c}_i$ repeat
    foreach $\vec{x}_i$ do
        assign $\vec{x}_i$ to nearest $\vec{c}$;
    end
    foreach $\vec{c}_i$ do
        recompute $\vec{c}_i$ as the means of assigned $\vec{x}_i$;
    end
until stopping criterion;
```

Algorithm 2: *k*-means pseudocode.

Two metrics for accessing the cluster quality are important here. The intra-cluster variability, called cohesion, should be as small as possible. This means that all the elements (for instance, images) assigned to a given cluster are the most similar each other as possible. The inter-cluster distance, called separation, should be the maximum, meaning that the elements of different clusters have quite different features and are well separated in the feature space. A 2D example of these two metrics is shown in Figure 9, where the area of the dashed ellipses represents the intra-cluster metric, and the dashed line between the two centroids represents the inter-cluster distance. Notice that such example is an ideal case, where the 2-D feature vector was able to capture the discriminatory characteristics of each class of objects. Usually, this is not the case in real world problems.

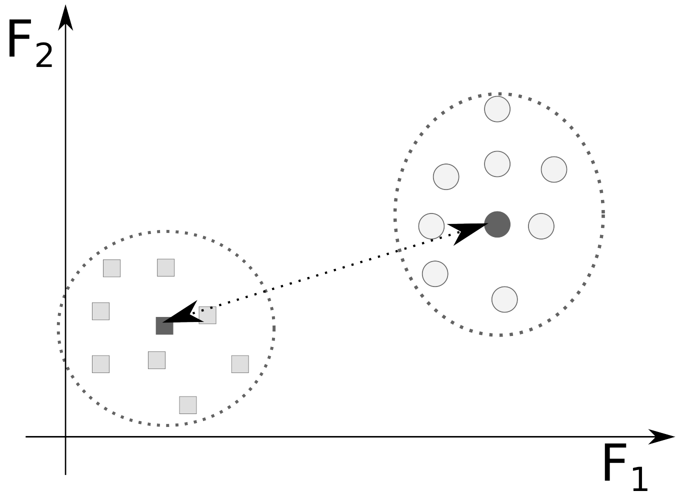

Figure 9: A hypothetical good clustering result, where similar objects are close together and there is a distance between the groups.

A method which makes a relationship between intra and inter cluster measures is called Silhouette Coefficient (SC). As stated by Tan et al. (2005), SC can be obtained by three steps:

1. For the $\vec{x}_i$ vector, calculate its average distance to all other vectors in its cluster. Call this

value a_i.

2. For the $\vec{x}_i$ vector and any cluster not containing the vector, calculate the vector's average distance to all the vectors in the given cluster. Find the minimum of such values with respect to all clusters; call this value b_i.

3. For the $\vec{x}_i$ vector, the SC is $s_i = (b_i - a_i)/max(a_i, b_i)$.

The value of SC varies between -1 and 1, where -1 represents the worst configuration and 1 the best cluster formation.

Derived from the original k-means, some improvements were proposed seeking to improve its efficiency in clustering complex data. The fuzzy k-means is a modification in the rule of assignment. Instead of restricting a sample assigned exclusively to one cluster, it is allowed to have samples assigned to several or all clusters, but with diverse degrees of membership to them.

Due to the iterative procedure of the algorithm, it is known that the clustering obtained by k-means is strongly dependent of the initialization of the centroids. To overcome this issue, (ARTHUR; VASSILVITSKII, 2007) proposed k-means^{++} that has an improved initialization. Basically, this k-means variant chooses the first centroid randomly from the data points. The remaining centroids are located in other data points, but they are chosen with a probability proportional to the square of the Euclidean distance from the current point to the nearest centroid already set. This simple modification leads to a speeds-up of two times compared with the basic algorithm and, mainly, a great improvement in error (compared with the optimal clustering)

2.7 RELATED WORK

In this section papers found in the recent literature related to our work are presented. They were divided into two subsections, one related to unsupervised feature learning and other to human classification by means of soft biometrics. This format was chosen to help the reader to understand which were the foundations and inspirations to construct this work.

2.7.1 UNSUPERVISED FEATURE LEARNING LITERATURE

The possibility to take advantage from unlabelled data in machine learning tasks has caught attention of several research groups. Different unsupervised learning methods were proposed for different tasks. The focus here was works that are concerned in employing such techniques to deal with visual problems.

Most of unsupervised learning methods proposed are based on learning a dictionary of codes, usually based on techniques such as sparse-coding, Restricted Boltzman Machines (RBM) and clustering algorithms.

The algorithm proposed by (SERMANET et al., 2013) consists in finding a dictionary learnt by sparse-coding that allows linear reconstruction of an input sample based on the combination of codes. This combination should use as little code as possible. Also, (HE et al., 2014) proposed an unsupervised feature learning framework that was constructed based on the ideas of the bag-of-visual-words, sparse coding and spatial pooling. The framework learns a hierarchy of features from a set of Scale Invariant Feature Transform (SIFT) descriptors.

Cai et al. (2010) prosed an approach called Multi-Cluster Feature Selection for selecting relevant features in an unsupervised fashion. The method tries to select features in a way that the multi-cluster structure of the data can be preserved. This is done by identifying the cluster structure from the original data, and subsequently measuring the importance of each feature for the differentiation of each cluster.

The role of single-layer networks in unsupervised feature learning is addressed by (COATES et al., 2011), where several different unsupervised learning algorithms are employed to discover features from unlabelled data. These features are then used to train a linear classifier. An important result of that paper concerns to the performance of the *k*-means algorithm.

Lee et al. (2011) proposed a method from learning invariant spatio-temporal features based on Independent Subspace Analysis which, in turn, is an extension of Independent Component Analysis. The proposed framework learn a hierarchy of features, taking advantage of two common operations of CNNs: convolution and stacking.

The unsupervised feature learning method proposed by (COATES et al., 2012) relies on two modules. The first uses the *k*-means algorithm to learn selective features, and the second uses a max-pooling strategy to combine those features. The algorithm was used in a dataset composed by 57 million 32-by-32 pixel patches.

To deal with aerial image classification, (CHERIYADAT, 2014) proposed an unsupervised feature learning technique. The main idea of the method is to generate a feature representation for each image patch. This is done by learning a set of normalized basis functions, which is computed by a variant of sparse coding, from extracted features (raw pixels, oriented filter responses and SIFT-based descriptors). Then, the features are encoded using the basis functions, and a soft threshold activation function is applied to generate sparse features.

Our proposed approach takes advantage of several pieces, which are known to be successful, such as CNN mechanisms, clustering and global optimization to construct a new framework for unsupervised feature learning. Differently from other approaches which learn features from small image patches, here we use the full image. The motivation for this approach is the idea of learning features that could be useful for the classification of high-level objects, such as faces and people.

2.7.2 SOFT BIOMETRICS LITERATURE

Some soft biometrics methods for extracting semantic information from people were developed in the last years. Different methodologies are available taking advantage of certain properties of the problem.

The work of (HANSEN et al., 2007) focused on annotating human features in surveillance videos. A person is described using the primary color of the hair, upper and lower body clothing, as well as his/her height. However, no clothes classification was done. The proposed methodology includes a background subtraction algorithm, a color descriptor based on the Hue-Saturation-Value (HSV) color model, a height estimator, and a head direction evaluator. The authors reported that the tests to clothes color description highlights difficulties when a person is wearing shirt and pants of the same color. Another problem occurred when people are wearing skin-colored clothes.

Zhang et al. (2008) proposed a methodology for clothes recognition, but restricted to t-shirts. They present a survey to evaluate which kind of detail/pattern is the most relevant for classifying t-shirts. Based on this survey, some methods were proposed to evaluate the sleeve length, recognize collar and placket, color analysis, texture recognition and shirt style recognition. More specifically, sleeve length recognition is based on face and skin detector, then, color segmentation and a one-level decision tree classifier. The images used for training and testing were collected in a controlled environment, with a constant background and people at almost on same pose. For sleeve length, they achieved almost 89% of correct classification. They reported that the errors were mainly caused by wrong skin detection.

The development of a robust color detection framework able to identify the colors of clothing in video under real illumination conditions was proposed by (D'ANGELO; DUGELAY, 2010). The objective was the identification of hooligans and the prevention of clashes in soccer matches. The proposed algorithm included stages of color constancy, color-space transformation and color matching. As classifier tem used a fuzzy k-nearest neighbours (KNN). The results showed that the proposed methodology was efficient for the specific purpose, achieving at most 94% of correct classification.

Bourdev et al. (2011) proposed a system to describe clothes of people using nine binary attributes. The detection and classification process relies on the Poselet detector, which uses a fully annotated training set and a complicated train scheme. Besides Poselets, a strategy for skin detection and segmentation was also employed. For classification a SVM with quadratic kernel was trained. The tests reported, show a mean average precision of 65.18%. Using a similar method, (WEBER et al., 2011) proposed a clothing segmentation approach. In this case, the H3D dataset was employed to construct a segmentation mask based on the Poselets detection, for later person retrieval based on their clothes appearance. They achieved reasonable good performance. It is important to notice that they were not classifying people clothes.

Bo e Fowlkes (2011) proposed a methodology for pedestrian image segmentation based on hierarchical composition of parts and sub-parts of image segments. The candidate parts and sub-parts were derived from a superpixels segmentation code. For each candidate segment a score was calculated to determine wheather it is part of a pedestrian or not. To do this, authors used a shape descriptor as well as color and texture histograms. Results suggested a promising methodology, but with some drawbacks. The concept of superpixels was also used by (YAMAGUCHI et al., 2012) for clothes labeling, based on pose estimation.

Chen et al. (2012) presented a fully automatic system capable of learning 23 binary and three multi-class attributes for human clothing. Human pose estimation was performed to find the location of upper torso and arms. Then, 40 features were extracted and quantized. This set of features was used to train an SVM classifier for each desired attribute. A Conditional Random Field (CRF) was calculated to extract the mutuality between attributes.

Employing the well-known Viola-Jones face detector, a modification of the GrabCut segmentation algorithm, the MPEG-7 color descriptor, the HOG shape descriptor and skin color detection, (CUSHEN; NIXON, 2013) presented a methodology for segmenting the upper body clothes in a mobile platform. They reported a F-score of 0.87, leading to a reasonable good performance.

In (DONG et al., 2013) authors define Parselets as a group of semantic images obtained by low-level over-segmentation, having strong and consistent semantic meaning. With this representation, a deformable mixture parsing model was proposed, based on a set of hand-designed feature descriptors and using a Gaussian Mixture Model (GMM) as a classifier. Using this method the authors managed to parse a human image, obtained by pixel segmentation, with a reasonable good performance.

A fully automated clothing suggestion approach was proposed by (KALANTIDIS et al., 2013). Authors used segmentation and hash process to extract and classify the clothes from a digital image. They used color quantization and LBP as descriptors, and claimed that the proposed method was scalable and very time-efficient.

Sharma et al. (2013) proposed a model for recognizing human attributes and actions. The model was based on a bag-of-words representation and SIFT features at multiple scales. The results related to human attributes was evaluated as good, but have shown that this problem is far from being solved.

Seeking to help the reader to comprehend the literature review, Table 1 consolidate the discussed papers based on the method main objective, pre-processing methods used, image descriptors, classification methods. It is possible to notice that most of the above-cited methodologies are based on the classical approach: some kind of segmentation method and/or hand-designed feature extractors followed by a classifier (e.g. SVM). But is not clear how to choose the suitable combination of pre-processing, feature extractor and classifier, leading to a complex solution definition. In this work we leverage the CNN capacity to learn both feature extraction and classification to propose a end-to-end soft biometrics classification framework.

Table 1: Literature consolidation for Soft Biometrics.

Authors	Objective	Pre-processing	Features	Classifier
Hansen et al. (2007)	Clothes color description	background removal	HSV color space	K-means
Zhang et al. (2008)	T-shirt recognition	skin detector and segmentation	texture descriptor	SVM
D'Angelo e Dugelay (2010)	Clothes color description	illumination normalization	RGB color space	Fuzzy k-NN
Bourdev et al. (2011)	Attributes classification	skin detection and segmentation	Poselet descriptor	SVM
Weber et al. (2011)	Clothes segmentation	skin detection and segmentation	Poselet descriptor	SVM
Bo e Fowlkes (2011)	Clothes segmentation	-	Superpixel segmentation	SVM
Chen et al. (2012)	Attributes classification	Human pose estimation	40 different features	SVM
Cushen e Nixon (2013)	Clothes segmentation	Face detection + image semantation	MPEG-7 color descriptor + HOG	K-means
Dong et al. (2013)	Human parsing	-	SIFT + HOG + color moment	GMM
Kalantidis et al. (2013)	Clothes segmentation	segmentation	color quantization + LBP	Hashing
Sharma et al. (2013)	Human attributes	-	SIFT + Bag of Words	SVM

Therefore, raw images can be used as input for the classifier, reducing the complexity of the model and computation effort. Besides that, the extraction of soft biometrics is based on the whole input image, thus the CNN must learn which part of the image is important and necessary to achieve the desired classification. The proposed method for soft biometric classification, for both image and videos, is detailed in Sections 3.2 and 3.3.

3 PROPOSED METHODS

In this section we focus our attention and discussion on proposed methods to describe visual data, images and videos, with high level information. As discussed in Section 2.1.1, a key step in pattern recognition is feature extraction. This can be done in three ways: hand-designed by an expert, unsupervised or supervised feature learning. Since one of the objectives of this work is to learn how to extract good quality features directly from data, we selected the last two lines of approach.

The first section of this chapter is dedicated to deal with unsupervised feature learning. The main objective is to learn a feature extractor which can separate images into classes, without using label information during the learning process. In essence, we are trying to verify if the low-level information contained in images is sufficient for class separation.

The second part of this chapter is dedicated to extract high level information by means of soft biometrics. Here the supervised option is selected, employing a CNN to learn an end-to-end soft biometrics classifier. In essence, we want to obtain a detailed description from raw images containing people. This part of research was published (PERLIN; LOPES, 2015). In addition to images as input, the proposed method is extended to deal with soft biometrics extraction from video frames.

3.1 METHOD FOR UNSUPERVISED FEATURE LEARNING

3.1.1 INTRODUCTION

Extracting semantic information from visual data, such as images and videos, is still an open problem in both computer vision and machine learning areas. The main idea is to develop computational methods capable of collecting, extracting and analysing data from images and videos, seeking to comprehend the visual information similarly as humans usually do.

A key point for the success of such methods is data representation, that is, the way data represents an image. Such representation is closely related to how images are manipulated and prepared (preprocessing operations and data transformations) for further use, usually a classification procedure. The representation can be considered at different levels and, after each transformation, the original data tends to be more condensed and more meaningful. This manipulation is called feature extraction, as it was discussed in Section 2.1.2. Choosing the appropriate representation level may help to improve the performance of the classification method.

Formally, feature extraction is a mapping function from $\mathbb{R}^n \to \mathbb{R}^d$, where n is the length of the original data vector and d is the length of the transformed vector. It is important to notice that, sometimes, feature extraction also leads to dimensionality reduction. But this is not always true.

The main drawback of this process is that the design of such mapping functions is a labor-intensive task. It demands a human expert to devise the suitable data transformations necessary to achieve a satisfactory efficiency of learning algorithms. Such dependence on the human capacity, exposes weakness of learning algorithms in computer vision, which are not able to manipulate image data for achieving satisfactory performance in an automatic way (BENGIO et al., 2013).

In recent years, deep learning methods have attracted the attention of the scientific community, since it has shown that it is useful for many visual recognition tasks. One of the well-known methods that uses deep learning concepts is the Convolutional Neural Network (CNN) (LECUN et al., 2010). The basic idea of deep learning is the mapping of (complex) input data throughout several layers of increasing abstraction and decreasing complexity. The major advantage of this method is its capacity to learn automatically a hierarchy of features from raw data, including images, streaming videos and sound. Since CNN is a supervised learning method, it requires labelled data during the training procedure. As a matter of fact, recent results of CNN were achieved thanks to the availability of annotated data. However, this is really laborious to obtain, since it depends on a human effort to classify data previously, according to his/her visual analysis and pattern recognition abilities. Of course, this is an important drawback of this method, constraining its application to the availability of large amounts of high-quality annotated data.

The amount of unlabelled (no tagging or annotation) visual data increases everywhere, and so the need for automatic methods that can learn how to extract useful features from such data. Some examples of unsupervised feature learning methods are clustering (k-means and Gaussian mixture model), matrix factorization (PCA, ICA and sparse coding), and non-linear embedding (Laplacian Eigenmaps, restricted Boltzman machines, etc) (LEE, 2010).

Pursuing the goal to advance in the use of unlabelled data during the process of image pattern recognition, we propose a feature extractor based on concepts of CNN, clustering and global optimization. Taking into account the ability of a CNN to learn a hierarchy of features, our starting point is the question: is it possible to adjust the weights of a CNN in an unsupervised way? The results shown in Section 4.3 suggest that this is possible. Basically, we view the training procedure of a neural network (in this case, a part of CNN) as an iterative optimization process, and we use an evolutionary computation-based method with a clustering procedure to create a cost function, so as to avoid the need of a previously annotated set of examples.

3.1.2 THE PROPOSED METHOD

In this section the proposed method for learning a feature extractor from data is presented. Seeking to help the reader to take a general view of the proposition, a block diagram is shown in Figure 10. There are three main blocks: differential evolution, convolution network and clustering. Connections are labelled with the kind of data passed between each block. The input for the process is a set of unlabelled images.

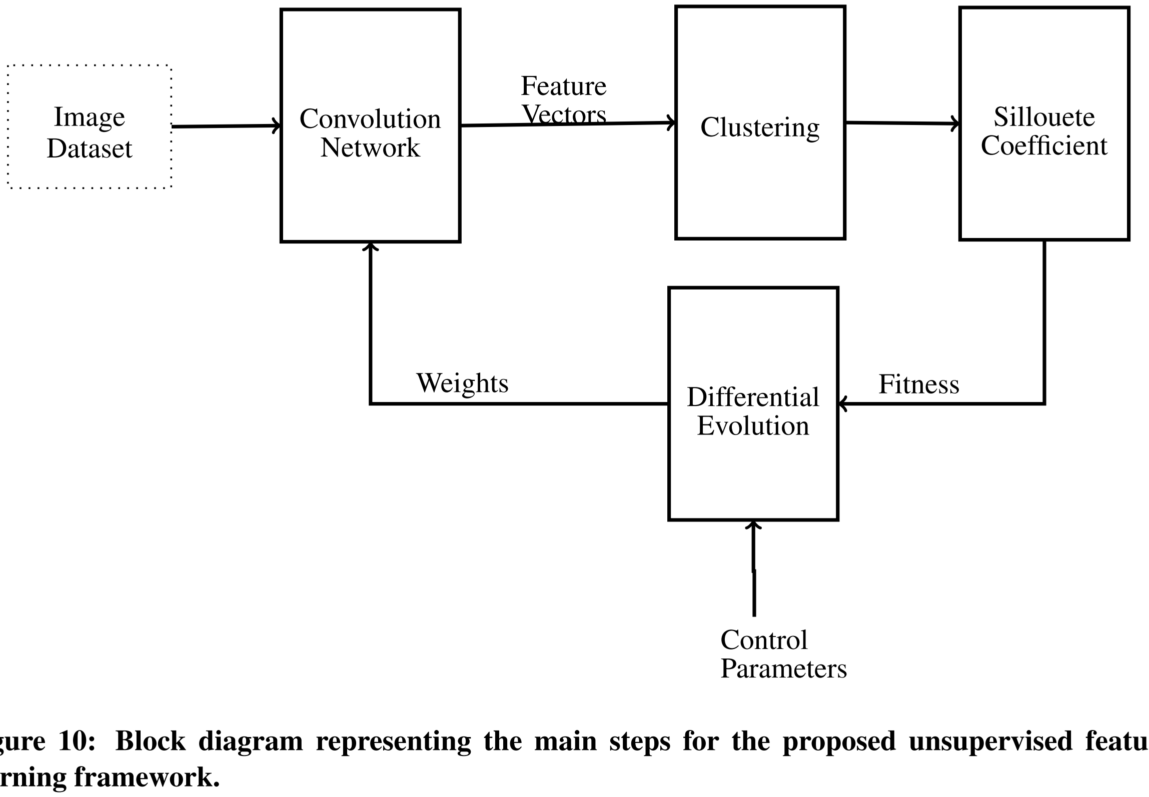

Figure 10: Block diagram representing the main steps for the proposed unsupervised feature learning framework.

As shown in Section 2.4, the CNN architecture could be divided into two modules, the feature extractor and the classifier. The output of the feature extractor module is a feature vector learnt directly from the input data. This vector is the input for the classifier. Both modules have trainable weights that determine the classification results. These weights are adjusted in a supervised way, using the well known back-propagation learning algorithm.

Formally, a CNN can be viewed as a parametrized function in the form of $y' = g(x,w)$, where y' is network prediction, x is an $\mathbb{R}^n$ input vector and w is the weight vector. During the training procedure, w is modified seeking to reduce the difference between y' and y (expected result) over the entire training set.

Since we are looking for the CNN composed by two parts, we split w into two parts w_f and w_c, which are, respectively, the weights associated to the feature extractor and to the classifier.

The main idea is to take apart the feature extractor module from a CNN and adjust their parameters in an unsupervised way. In this way, this can be formally written Equation 12, where $fe \in \mathbb{R}^d$ is the extracted feature vector. Since the label associated to a x_i sample is assumed not to be available, to find the components of w_f turns out to be a difficult task. However, considering the parameter adjustment procedure as an optimization problem, allow us to use an optimization heuristic for this task, in this case DE, as mentioned.

$$fe = g(x, w_f) \tag{12}$$

As discussed in Section 2.5, DE is based on a population of d-dimensional individuals, where each one represents a candidate solution. In the proposed method, each individual is a vector grouping all adjustable parameters of the feature extractor. In this way, during the algorithm execution, many different feature extractors will be tested.

A key point for the execution of the DE is the evaluation of a candidate solution, and this is accomplished by specifying a *fitness* function. Here, the quality of the solution implies in evaluating how good are the feature vectors generated by the extractor. A natural choice should be to verify the performance of a supervised classifier based on the feature vectors. But this is not possible since it is assumed that input data is unlabelled.

The main reason to use the output of a feature extractor as the input to a classifier, instead of using the original raw data, is because the former is supposedly much more informative than the latter. Since the idea of a classifier is to find a separation boundary between the desired classes, the more separable are the input data, the easier will be to determine that boundary.

This fact gives us a clue to evaluate the quality of a candidate solution generated by

DE. The feature extractor could be adjusted so as to produce vectors which are similar when the input data are similar, and very different when input data are different. In other words, feature vectors generated from images with similar semantics should form a compact cluster and far from semantically different clusters. Basically, this is the same definition of a clustering algorithm, as stated in Section2.6. In this way, the fitness function for a candidate solution, is the clustering quality obtained by grouping the feature vectors generated by the adjustable feature extractor.

The quality of a feature extractor will be directly related to the quality of the clusters obtained by the clustering algorithm. Feature vectors assigned to the same cluster should represent similar input data. The distance between the generated clusters should be maximum, meaning that the feature vectors from different inputs are dissimilar.

In this work, we employed k-means as clustering technique since it is fast enough to be inserted into the training loop, and it yields solutions of satisfactory quality. Additionally, an assumption to k-means functionality is the presence of cluster with cohesion and separation, the same result which is expect here. After executing k-means, an assignment vector is obtained, indicating to which cluster each vector belongs, which allows us to obtain a clustering quality metric.

Most clustering metrics, such as Mutual Information (VINH et al., 2010), Adjusted Rand Index (HUBERT; ARABIE, 1985) and V-Measures (ROSENBERG; HIRSCHBERG, 2007), are based in the ground truth data, that is, the label of annotated data. Since this is not available, the Silhouette Coefficient (SC), as detailed in Section 2.6, was chosen alternatively (B; RAWI, 2008).

Basically, the proposed method consists in the execution of the DE algorithm seeking to find a suitable set of parameters for the feature extractor. Each individual of the DE population is a d-dimensional vector, where d is the total number of adjustable parameters from the extractor. Each individual should be randomly initialized using an uniform distribution over the feature space. For each individual, its fitness should be calculated. Next, mutant vectors are generated according to Eq. 6, and they are combined with the original one using Eq. 7, thus generating the trial vectors. The fitness of each trial vector is calculated and it it used in the selection step, according to Eq. 8. Mutation, recombination, evaluation and selection are repeated until the stopping criterion is reached. At the end of the execution, the individual with minimum fitness is the solution, i.e., the set of parameters for the feature extractor. A flowchart to illustrate the framework execution is shown in Figure 11.

As stated, the input dataset for the proposed method is composed by M images with

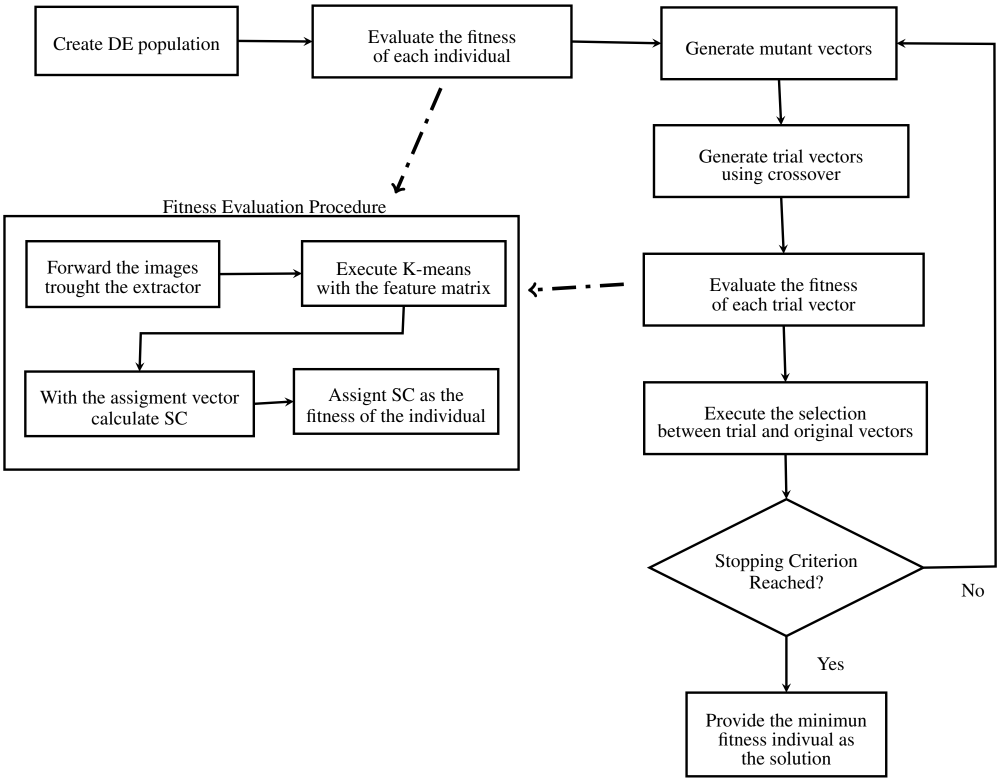
Create DE population
Evaluate the fitness of each individual
Generate mutant vectors
Generate trial vectors using crossover
Fitness Evaluation Procedure
Forward the images trought the extractor
Execute K-means with the feature matrix
With the assigment vector calculate SC
Assign SC as the fitness of the individual
Evaluate the fitness of each trial vector
Execute the selection between trial and original vectors
Stopping Criterion Reached?
No
Yes
Provide the minimun fitness indivual as the solution

Figure 11: Flowchart showing the proposed method. Dashed lines indicates a sub-process.

$C \times W \times H$ pixels, where C is the number of channels, and W is width and H is height.

The key step of the proposed method is the fitness evaluation. To evaluate the quality of an individual, the extractor is initialized with candidate solution values. All the images must be forwarded thought the feature extractor. For each image set, a d-dimensional feature vector will be generated. The d value depends upon the image size and the architecture of the extractor. At the end, an $M \times d$ matrix is generated. The k-means algorithm is then feeded with the matrix and, after its execution, an assignment vector is obtained. With this assignment vector, the SC metrics can be calculated to determine the fitness of the individual.

The extractor architecture, which is the definition of its layers, is an important piece of information, since the architecture chosen will influence directly the DE performance. Recall that the size of an individual is the total number of adjustable parameters of the feature extractor. Finally, the number of clusters that the k-means should find is a crucial parameter, and must be manually set. Since that, we assumed this action as a simplification of the proposed method.

3.1.2.1 FEATURE EXTRACTOR MODELS

Since we seek to develop a feature extractor relying on some CNN concepts, the architecture of the layers is something very important. The details of each layer has direct impact in both, the DE execution, and the produced feature vectors. The more feature maps and their kernels, the larger will be the search space for DE and, thus, the harder it will be to find a good solution at each training cycle.

Therefore, two different models were tested. The major differences between each models are the number of feature maps per convolution layers and the number of layers. The models have a 3D input, which receives the RGB/YUV image. Each convolution stage uses rectified linear units (ReLU) and Max-pooling, using a 2×2 neighborhood. The output of the last pooling layer is reshaped into a vector, and each component is filtered by a sigmoid function. These specifications are both based, on the available literature and empirically defined.

The first model, identified as Model #0, has two convolution stages, and each one generates 10 feature maps, using a 5×5 kernel. This model has a total of 3250 adjustable parameters. The second, Model #1, has three convolution stages, each one generates 10 feature maps, using a 5×5 kernel. It contains a total of 5750 adjustable parameters.

3.1.3 COMPARISON WITH OTHER FEATURE EXTRACTORS

The main objective of a feature extractor method is to transform the input data into a meaningful form, seeking to facilitate the training of a classifier method. As discussed before, there are available several different feature extractor methods in the literature.

In order to compare the proposed method, two different feature extractors were selected. The first one is HOG, since it was first designed as a solution for the people detection problem (DALAL; TRIGGS, 2005). Later it was applied to different problems, always producing good solutions. It is important to notice that HOG is a hand-designed feature extractor. Therefore, a field expert had to design it.

The second choice was PCA, which is an orthogonal transformation which generally maps a higher dimensional vector into a lower dimensional space. This mapping process is based on the eigenvalues of the dataset covariance matrix. This method is commonly employed to generate feature vectors for face detection problems. Most importantly, it is an unsupervised method.

Since the idea is to compare the feature extractor methods, a common classifier was chosen. The SVM was used to train a classifier using the three different feature vectors. Two different types of SVM were employed, linear and radial basis function.

The proposed method is validated by a series of experiments discussed in Sec 4.3.

3.2 METHOD FOR SOFT BIOMETRICS EXTRACTION USING CNN

3.2.1 INTRODUCTION

In this part of the work we aim at extracting and classifying automatically soft biometric information from images containing a person. By now, we assume that the extractor will operate after a people detector module, that is, a bounding box containing a person will be the input for the method.

The extraction includes gender, as well as the upper and lower clothes worn by the individual, based on their appearance. In other words, we are proposing to extract this kind of high level semantic concept based on the whole image. For the upper clothes (UC) there are two possible classes: short and long sleeves, and for the lower clothes (LC), pants and shorts. We are also interested in a more difficult task: determining the gender of the subject into two classes (male and female), based on the full body appearance.

3.2.2 THE METHOD

The extractor will consist in a supervised trained CNN classifier. We propose two different operation modes (OM) for training and applying the extractor. The first one (OM #1) consists of three independently trained sub-classifiers, each one dealing with a different classification problem: an UC classifier, a LC classifier, and a gender classifier. The output of each classifier is a vector with two units, and a winner-takes-all strategy is used to indicate which is the class of the image for each soft biometric.

The second operation mode (OM #2) deals with the three soft biometrics at the same time, describing each sample based on the answer of just one classifier. In this way, the classifier is trained to take into account all the semantic aspects within the input image. Here, the output of the classifier is a three continuous unit vector, where each unit represents a soft biometric with values in the range $[-1..+1]$. Therefore, the output of the classifier is a multilabel answer, allowing three different attributes to be evaluated at the same time. This is not usual for CNNs, since, in most cases, just a single class would be expected for each image.

Since the CNN allows the construction of an end-to-end classifier, the proposed approach for the semantic description of a person is based only on the raw input image, without any preprocessing. Table2 shows some details of the CNN architecture used in this work, which is the same for both OM #1 and OM #2. The difference is only in the number of outputs. The architecture parameters were adjusted based on the current literature (LECUN et al., 2010;

KRIZHEVSKY et al., 2012; SRIVASTAVA et al., 2014), as well as by preliminary experiments.

Table 2: Description of the architecture of the CNNs used in this work. Each column represents a layer. * For OM#1, the number of outputs is 2 for each classifier. In OM#2 the number of outputs is 3.

	Conv1	MaxPool1	Conv2	MaxPool2	Conv3	MaxPool3	Linear1	Linear2
Input Maps	3	32	32	64	64	128	1152	768
Input Size	128×128	60×60	30×30	24×24	12×12	6×6	-	-
Output Maps	32	32	64	64	128	128	768	*
Output Size	60×60	30×30	24×24	12×12	6×6	3×3	-	-
Kernel Size	10×10	2×2	7×7	2×2	7×7	2×2	-	-
Stride	2×2	2×2	1×1	2×2	1×1	2×2	-	-

The input for the networks is a 3D array with $3 \times 128 \times 128$ elements, defining a 128×128 RGB image.

The first convolution layer has 32 feature maps obtained by a set of 10×10 convolution kernels, using 2×2 strides. After the application of the ReLU squashing function, the next step is the sub-sampling, accomplished by maxpooling. The neighbourhood size used was 2×2. These three steps compose the first stage of the network.

The second stage is composed by a convolution layer with 64 feature maps and a 7×7 kernel. Again, after the convolution, both the ReLU function and the maxpooling operation were applied.

The third stage and the last part of the so-called feature extractor is another convolution step, which generates 128 feature maps using 7×7 kernels. Following, the convolution the ReLU and the maxpooling operation were applied.

The output generated by the last sub-sampling operation is $128 \times 3 \times 3$ large, rearranged as a 1152 unidimensional vector. This is the input to the linear stage. It is possible to observe that the convolution and sub-sampling layers work together for reducing dimensionality.

The linear stage is composed by the input layers, followed by a hidden layer containing 768 neurons. The output layer has 2 neurons for OM #1 and 3 neurons for OM #2. The number of neurons in the hidden layer was determined using the relationship $2/3$ of the size of the preceding layer. For OM #1, a winner-takes-all strategy was used to determine the classifier output. For OM #2, each output was evaluated separately

The output of the model is filtered by a softmax, or normalized exponential, function

$\sigma(y)$. This operation, denoted by Eq. 13, squashes a continuous n-dimensional vector y, in the range $(0,1)$, in such way that $\sum_{i=1}^{n}(y^i) = 1$ (BISHOP, 2006).

$$\sigma(y^i) = \frac{exp(y^i)}{1+\sum_{j=1}^{n}(exp(y^j))} \tag{13}$$

.

During the training process, the network neurons tend to co-adapt their weights based on the neighbourhood elements. This phenomenon can lead the network to lose its generalization capability. To prevent this phenomenon, some randomly chosen neurons are omitted from the network, with probability 0.5, so the elements cannot rely on other units. Basically, this is the dropout strategy proposed by (HINTON et al., 2012).

The adjustment of the network's weights was carried out by the backpropagation of the error using the mean squared error as loss metric and the SGD algorithm as the optimization strategy. The learning rate starts at 0.005 and decreases by half every 30 training epochs. The momentum was set to 0.9 so as to help the convergence of the method. The SGD optimization process was executed for a maximum of 1000 epochs. Definition of these parameters are empirically based on the available literature.

The CNN training process is very time consuming, not only because the network has many layers and neurons but, also, because many computations are required at each step of the training. Considering the algorithmic issues and the high level parallelism of the operations involved, an interesting strategy is the use of GPU as the horse power to reduce the total training time. In order to obtain efficiency from the GPU, a mini-batch scheme may be used. Here, the size of mini-batch was set to 128 as suggested by (KRIZHEVSKY et al., 2012).

3.2.3 HAND-DESIGNED IMAGE FEATURE EXTRACTOR AND CLASSIFIER

A CNN approach does not require the definition of a feature extractor or a classifier before being used. Therefore, we created a hand-designed feature extractor and classifier, shown in Figure 12, for comparing its performance with the CNN.

The first part of the diagram is the feature extractor that uses the raw image (supposedly containing the image of a person with clothes) to compute a 4200-long feature vector. This vector is composed by three groups of elements extracted from the image: 3780 features using HOG; 320 features from the top-5 interest points (each one produces a 64 float-point vector) using SURF; and 100 features using LBP.

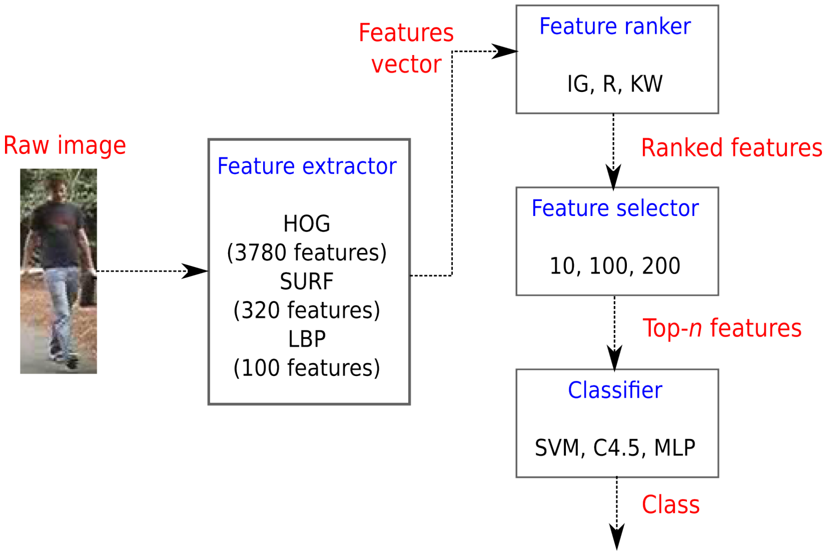

Figure 12: Hand-designed feature extraction and classification process.

HOG is a very popular method that, for years, was considered the state-of-art people detector (DALAL; TRIGGS, 2005). SURF is also a popular method (BAY et al., 2008), very efficient for many applications, from object recognition to face detection and recognition (CHIDAMBARAM et al., 2014). LBP (OJALA et al., 2002) is a simple but robust texture descriptor, and it was successfully applied to face recognition problems (AHONEN et al., 2006) and pedestrian detection (GAN; CHENG, 2011). The LBP descriptor is a histogram of the detected binary patterns in an image, such as texture. The histogram size was defined as 100 bins. As can be seen, the three chosen descriptors are complementary, in the sense of extracted images features.

Before concatenating the three feature vectors into a single one, all elements of each vector were locally normalized in the interval $[0,1]$. This was done using the min-max normalization, as shown in Equation 14.

$$minmaxNorm(x_i) = \frac{x_i - X_{max}}{X_{max} - X_{min}} \tag{14}$$

where, X_{max} is the maximum component of the vector, X_{min} is the minimum component of the vector and x_i is a single component.

Next, the predictive power of each feature regarding the class is accessed by applying some well-known statistical methods: Information Gain (IG), Spearman's rank correlation coefficient R, and the Kruskal-Wallis score (KW).

IG, also known as Mutual Information in some domains, is an entropy-based measure that evaluates the gain of information with respect to the class when a specific feature is chosen. Given the entropies of two features, $H(X)$ and $H(Y)$, $IG(X,Y)$ is given by $IG(X,Y) = H(X) - H(X|Y)$, where the last term is the conditional entropy of X given Y. Considering X as the values of a feature for the several image samples, and Y as the corresponding classes to which they belong to, therefore $IG(X,Y)$ gives a measure of how predictable a class is considering only that feature. This is the rationale for using IG for feature selection (LEE; LEE, 2006).

Spearman's rank correlation coefficient R is a nonparametric measure between two variables (BLUMAN, 2009), and it is defined by Equation 15. This measure of correlation can be used for both continuous or discrete variables and it does not require that data has normal distribution. If one of the variables is a feature and its values, and the other the classes, R can be used as a method for ranking sets of features according to their predictability.

$$R = 1 - \frac{6\sum d_i^2}{n(n^2-1)} \tag{15}$$

where, $d_i = x_i - y_i$ and, x_i and y_i are samples of the two variables being evaluated.

The KW score (also known as H test) is a nonparametric statistic frequently used for comparing means of more than two populations when they are not normally distributed and their variances are not equal (CORDER; FOREMAN, 2014). Data values are transformed into ranks and considered as a whole and separated by classes of samples. If the samples of a class differ significantly from those of other classes, this statistic will give a high value and, if the differences are small, so it is the result of the test. When applied to feature selection, the higher the value of the score, the better. For features extracted from image data, this means that a given feature has quite different values for a given class.

The result of the feature ranker are three 4200-long vectors whose elements are ranked according to the values computed by IG, R and KW, as described above.

Next, a threshold is applied to limit the length of the ranked feature vector. Three values, empirically chosen, were used for selecting the top features: 10, 100 and 200. This was done to eliminate possible noise and improve performance of the classifiers (next block).

Finally, the top-n features (n=10, 100 or 200) were applied to three different classifiers: C4.5, SVM and MLP. C4.5 is a well-known tree-based classifier based on IG that has been used for years as the baseline for comparison of classification methods (QUINLAN, 1993). SVM (CORTES; VAPNIK, 1995) is one of the most used classifiers not only in computer vision but, also, in pattern recognition and data mining. MLP is a feed-forward neural network with multiple hidden layers that is frequently used in general classification tasks, including images

(VENKATESH, 2003). The main parameters of the classifiers used are shown in Table 3. To determine SVM parameters a grid search was performed, following (HSU et al., 2003).

Table 3: Main parameters of the hand-designed classifiers.

C4.5	
Confidence factor	0.25
Minimum # objects per leaf	10
Pruning	true
MLP	
Auto build/connect hidden layers	true
Number of hidden layers	$(\#classes + \#features)/2$
Learning rate	0.3
Momentum	0.2
Training epochs	500

The proposed method is validated by a series of experiments presented in Section4.4.

3.3 METHOD FOR SOFT BIOMETRICS APPLIED TO VIDEO

Both static images and videos are used as an extension of our memory, allowing us to store relevant data. As discussed above, at the low level, an image is just a set of some primitive artifacts such as dots, edges, gradients and shapes. In general, we are interested in the high level information contained in the image or video.

In this way, the method described in Section 3.2 was extended to deal with videos, allowing the description of a frame sequence containing a person, using the same three kinds of soft biometrics used before: gender, upper and lower clothes. In essence, the method for extraction of such soft biometrics remains similar, being employed three different supervised trained CNN to deal with each of those concepts. Based on the available literature and mainly in the results from Section 4.4, for the extraction of soft biometrics in video frames, some modifications in the CNN model architecture were proposed.

The first one is related to the input size, instead of using 128×128, we employed 128×64. The main reason for this is that during the experimentation process, a framework upgrade allowed to deal with non-squared images and kernel. This opened the possibility to use the original images during the training, reducing the size of the model and the training time. As will be shown in Section 4.5, this modification did not degrade the extraction performance.

Second, as discussed before, feature extraction plays a major role in pattern recognition. Seeking to observe the influence of feature vector size in CNN performance, different configurations were proposed. Structurally, they are similar to that one employed in

Section 3.2, containing convolution layers, followed by the application of the ReLU squashing function and, finally, subsampled by a maxpooling operation. To modify feature vector size, the number of feature maps are modified, leading to four models, called Small, Medium, Large and VeryLarge. They are described in Tables 4, 5, 6 and 7.

Table 4: Description of the architecture of the Small model. Each column represents a layer.

	Conv1	MaxPool1	Conv2	MaxPool2	Conv3	Conv4	Linear1	Linear2
Input Maps	3	16	16	32	32	64	64	42
Input Size	128×64	125×60	41×20	38×16	12×5	9×1	-	-
Output Maps	16	16	32	32	64	64	42	2
Output Size	125×60	41×20	38×16	12×5	9×1	1×1	-	-
Kernel Size	5×4	3×3	5×4	3×3	5×4	1×9	-	-
Stride	1×1	3×3	1×1	3×3	1×1	1×1	-	-

Table 5: Description of the architecture of the Medium model. Each column represents a layer.

	Conv1	MaxPool1	Conv2	MaxPool2	Conv3	Conv4	Linear1	Linear2
Input Maps	3	32	32	64	64	128	128	85
Input Size	128×64	125×60	41×20	38×16	12×5	9×1	-	-
Output Maps	32	32	64	64	128	128	85	2
Output Size	125×60	41×20	38×16	12×5	9×1	1×1	-	-
Kernel Size	5×4	3×3	5×4	3×3	5×4	1×9	-	-
Stride	1×1	3×3	1×1	3×3	1×1	1×1	-	-

Table 6: Description of the architecture of the Large model. Each column represents a layer.

	Conv1	MaxPool1	Conv2	MaxPool2	Conv3	Linear1	Linear2
Input Maps	3	16	16	32	32	576	384
Input Size	128×64	125×60	41×20	38×16	12×5	-	-
Output Maps	16	16	32	32	64	384	2
Output Size	125×60	41×20	38×16	12×5	9×1	-	-
Kernel Size	5×4	3×3	5×4	3×3	5×4	-	-
Stride	1×1	3×3	1×1	3×3	1×1	-	-

Analysing those tables it is possible to notice that the four models are very similar to each other. The differences are small changes in the number of feature maps. That is, the size of the feature vector produced by the convolution and subsampling layers. The Small model produces a 64-d feature vector, Medium produces a 128-d feature vector, Large produces a 576-d feature vector and VeryLarge produces a 1152-d feature vector. The output of all models is a 2-d vector, where each dimension is used to identify the classes for each soft-biometric.

Table 7: Description of the architecture of the VeryLarge model. Each column represents a layer.

	Conv1	MaxPool1	Conv2	MaxPool2	Conv3	Linear1	Linear2
Input Maps	3	32	32	64	64	1152	768
Input Size	128×64	125×60	41×20	38×16	12×5	-	-
Output Maps	32	32	64	64	128	768	2
Output Size	125×60	41×20	38×16	12×5	9×1	-	-
Kernel Size	5×4	3×3	5×4	3×3	5×4	-	-
Stride	1×1	3×3	1×1	3×3	1×1	-	-

Since we are dealing with videos, which are a sequence of static images shown at a certain rate, we take advantage of the time dimension to verify and correct the output of the extractor. This is done by employing a sliding window to filter the output, based on the mean of a number of previous frames. The filtering operation is defined by Equation 16, where W represents the window size, t the time instant, and y the classifier output.

$$\bar{y}^t = \frac{1}{W} \sum_{i=0}^{W-1} (y^{(t-i)}) \tag{16}$$

This approach turned out to be necessary since some changes between each frame, such as pose, partial occlusion, illumination, and other effects, can influence the classifier output. If a filter is used, the output can be smoothed, thus leading to better results. An example of application of this filter using different window sizes over a random variable is shown in Figure 13.

In this example a hypothetical classifier output, varying in the interval $[0,1]$, is represented. The target output is denoted by a solid black line and, in this case it is set to 1.0. A threshold of 0.5 is set, implying that values higher than 0.5 are considered correct, and less than 0.5 incorrect prediction. Three different window sizes were applied $(5,10,15)$ leading to three new outputs. It is possible to see that a small window size has a lower influence on the actual time step. Whereas larger windows can produce smoother outputs.

This filter operation can be interpreted as a temporal supervisor, which allows the correction of the output in time step t, based on some previous outputs. Analysing Figure 13, depending on the window size applied, the desired correction could not be achieved. In this way, finding the appropriate window width requires experimentation.

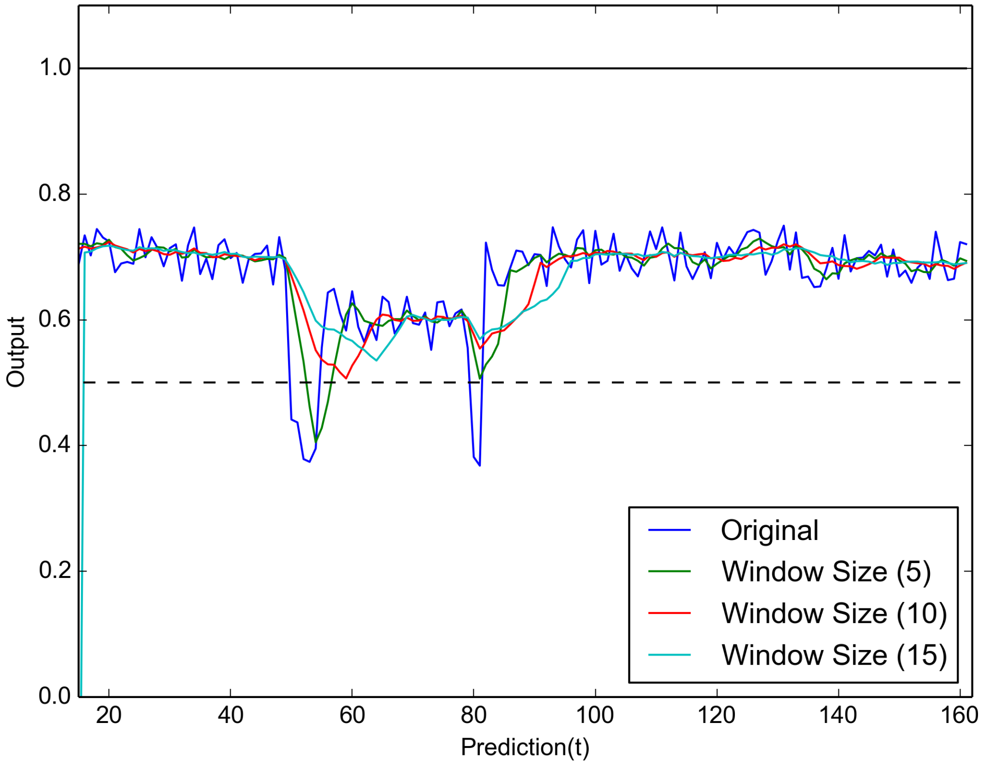

Figure 13: Example of mean filtering of an output. The blue line is the original signal, and other lines represent the signal filtered by different window sizes. The solid black line is the target, and the dashed line represents a threshold.

4 EXPERIMENTS, RESULTS AND DISCUSSION

A series of experiments were performed to test the proposed methods. Here they are presented along with discussions.

4.1 EVALUATION PROTOCOL

Since all experiments use a similar evaluation protocol, the metrics and tools employed to evaluation are here presented. When dealing with binary classifiers, one class will be the positive and the other negative. Therefore, four different measures can be computed using the outcomes of a classifier for a set of instances, in this case, input images:

- *true positive (tp):* number of positive instances that were correctly classified as positive;
- *true negative (tn):* number of negative instances that were correctly classified as negative;
- *false positive (fp):* number of negative instances that were wrongly classified as positive;
- *false negative (fn):* number of positive instances that were wrongly classified as negative;

Three metrics to evaluate the performance of the method were employed: accuracy (ACC), sensitivity (Se) or true-positive rate, and specificity (Sp) or true-negative rate, defined in Eq. 17. These metrics are based on a classifier output. Considering a two-class problem, true positive (tp) outcome is a correctly identified pattern; false positive (fp) is a incorrectly identified sample; true negative (tn) is a correctly rejected sample; and false negative (fn) is a incorrectly rejected sample.

$$ACC = \frac{(tp+tn)}{(tp+tn+fp+fn)} \qquad Se = \frac{tp}{(tp+fn)} \qquad Sp = \frac{tn}{(tn+fp)} \tag{17}$$

A graphical aid to visualize the performance of the classifiers used in this work is the Receiver Operating Characteristics (ROC) chart, which relates both true-positive and true-negative rates for a certain classifier (FAWCETT, 2006).

The ROC chart can be produced in two ways. If the output of a classifier is a continuous score, than a variation in the discrimination threshold, will produce different points in the space, leading to a curve. When the classifier produce a label, a single point in the space is obtained.

Points of a ROC chart represent the trade-off between true-positive and true-negative rates. Therefore, when plotting the performance of a classifier, the best operation point would be that at the top-left corner, representing the optimal ability of the classifier to separate classes. As the points get closer to the ascending diagonal, which represents random guessing, that indicates that the classifier has a poor ability to separate classes. Using this method, the quantitative evaluation of a classifier is given by the Area Under the Curve (AUC), which is the area in the ROC space covered by the classifier curve. The AUC varies from $[0..1]$, where 1 indicates the best possible performance.

To generate the evaluation results two different strategies were used. To experiments reported in Sections 4.3 and 4.5, the learning process was evaluated using a set of images for training and another set for testing. Just for clarify, both image sets are disjoint. To the experiments reported in Section 4.4, the classifiers were trained using a 10-fold cross-validation strategy. The procedure consists in splitting the dataset into 10 parts. Nine parts are selected to the train the model and one part is reserved to testing. Therefore, the model is trained/tested with 10 different datasets.

4.2 IMAGE DATASETS

In this section we detail the image datasets used during the experiments.

4.2.1 FACES DATASET

The Faces Dataset [1], is composed by images of faces (positive class) and random patches representing non-face images (negative class). The size of each sample was set to 32×32 pixels, encoded as a YUV image. The training set had 5000 images for both classes, and the testing set contained 1200 images. Some samples of this dataset are shown in Fig 14.

[1] Available at http://data.neuflow.org/data/faces_cut_yuv_32x32.tar.gz

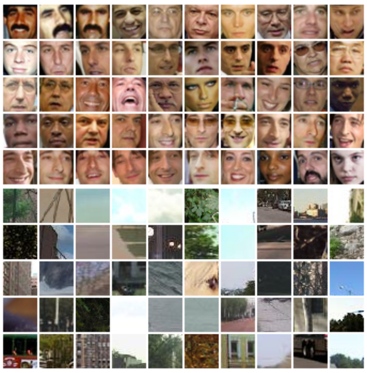

Figure 14: Samples from the Faces Dataset. The three first rows show examples for the Face class, while the last three for the Background class.

4.2.2 PEDESTRIAN DATASET

The Pedestrian Dataset, was composed by images from people (positive class) and background (negative class). It was built using selected images from three different datasets: H3D(BOURDEV; MALIK, 2009), ViPer (GRAY et al., 2007) and MIT Pedestrian[2]. A total of 3000 images were used for training the feature extractor and other 1000 for validation. Images were resized from 128 × 64 to 64 × 32 pixels. Some samples of this composite dataset are shown in Fig 15.

4.2.3 SOFT BIOMETRICS DATASET

Three datasets were used for training and evaluating the soft biometrics classifiers. The image set was composed by a mix of the H3D dataset, provided by (BOURDEV et al., 2011), the ViPer dataset created by (GRAY; TAO, 2008), and the HATdb made available by (SHARMA; JURIE, 2011). The idea behind using a mix of different image datasets is to indirectly improve the generality of the proposed method for real-world images.

The H3D includes a large variety of people, wearing different clothes, standing in different poses, sometimes subject to occlusion and cluttered background. Therefore, this dataset is challenging for pattern recognition methods. The original dataset contains 4013 images annotated with 9 binary attributes (male/female, long hair, glasses, hat, t-shirt, long sleeves, shorts, jeans, long pants). From this dataset we selected only those images with people standing up, thus remaining 1252 images. This was done by adjusting bounding boxes around a person with an aspect ratio larger than 2. Then, all examples were normalized in size to fit a frame of 128 × 64 pixels. Seeking to use just representative examples, a careful manual

[2] Available at http://cbcl.mit.edu/software-datasets/PedestrianData.html

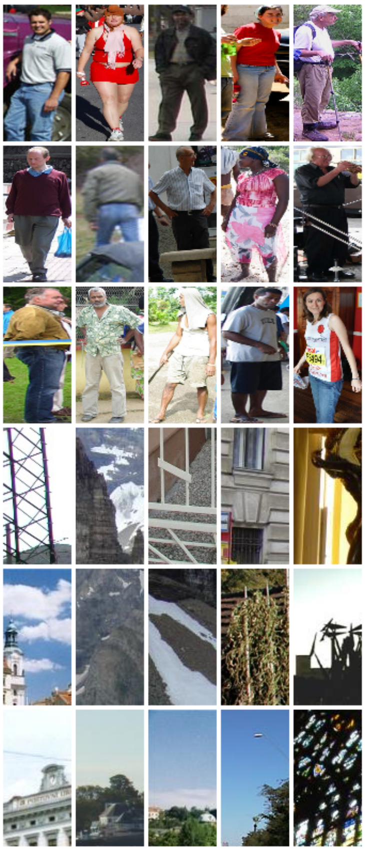

Figure 15: Samples from the Pedestrian Dataset. The three first rows show examples for the Person class, while the last three for the Background class.

selection and annotation were done.

The ViPer dataset is composed by 1188 images of 128 × 48 pixels taken from people walking in various places. Each subject has two different images taken from different positions. Both images were used to create the training set. This dataset does not have the desired labels to be used during the network training. Consequently, a manual annotation was necessary.

The original HATdb dataset is composed by 9344 images, annotated with 27 attributes. We selected a subset (at most 2615 images), only using images classified as standing people, and it is used the bounding box information to crop the person and resize it to 128 × 64 pixels. We also performed an additional revision of the annotation provided.

Considering that the classifiers were designed to deal with 128 × 128 images, the size of all the images were standardized by padding with 0's to achieve the desired dimensions. The distribution of images per class in each dataset is shown in Table 8. It is important to notice that we selected only images that have a good representation of the desired class. Additionally, we intended to maintain the class balance, so the total number of images used per attribute may be different.

Table 8: Number of original images per class used in the experiments.

	Gender		Lower Clothes		Upper Clothes	
	Male	Female	Long	Short	Long	Short
H3D	656	596	169	282	326	447
ViPer	564	624	291	178	574	453
HATdb	2615	2615	492	492	771	771
Total	3835	3835	952	952	1671	1671

Since the number of images available was small for effectively training the CNN, we expanded the dataset by using random transformations, such as translations (up to ± 10 pixels for x and y), scaling (from 0.98 to 1.1), rotation (up to $\pm 10^o$) and horizontal flip.

This data augmentation strategy is applied in an on-line way, i.e., for each epoch in the CNN training, random transformations are applied for each training sample. This is done in such a way that the total number of examples shown to the CNN at each epoch is around to 10,000 samples per class.

Similarly, the same data augmentation procedure was used for the other classifiers, reaching the same number of 10,000 samples per class.

4.2.4 VIDEO DATASET

This dataset is composed by frames taken from the SAIVT dataset (BIALKOWSKI et al., 2012), which is a set of surveillance videos recorded from 8 different camera positions. Just frames of a single camera were used in the evaluation process, since its installation position allows recording people with poses similar to those used in the training procedure. Some examples of frames used are shown in Figure 16.

There are annotation details available for 124 people who walked trough the monitored area, leading to a total of 4540 samples. The distribution among classes is detailed in Table 9. It is important to notice that, for the testing set, a total of 1000 samples were randomly selected from all subjects for each class. As validation set, all samples for each subject were used.

Table 9: Distribution images per class from SAIVT Dataset.

	Gender		Lower Clothes		Upper Clothes	
	Male	Female	Long	Short	Long	Short
SAIVT	2686	1854	3099	1441	1055	3485

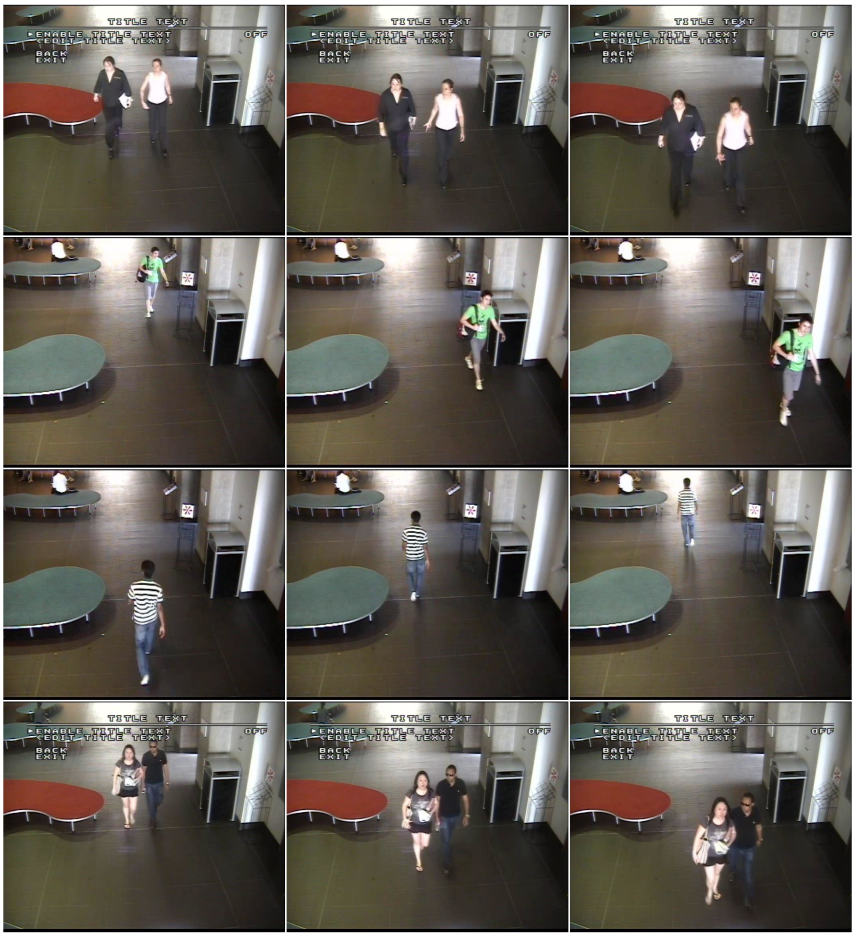

Figure 16: Examples of frames used as evaluation dataset. Annotation bounding boxes were used to extract patches with people from each frame.

4.3 UNSUPERVISED FEATURE LEARNING

In this section we present the experiments conducted to evaluate the unsupervised feature learning method for image separation as discussed in Section 3.1. The main objective of this section is to evaluate the performance of the proposed hybrid method to learn features directly from raw input images in an unsupervised way.

4.3.1 IMPLEMENTATION DETAILS

The framework described in Section 3.1 was implemented using the Python language, taking advantage of the flexibility and efficiency of existing packages, such as *numpy* and *scipy*. The last one provided the implementation for the *k*-means and the calculation of the SC metrics.

DE was implemented using the Inspyred[3] library, which was also built using Python.

[3] Available for download at https://pypi.python.org/pypi/inspyred

To model the CNN feature extractor, the Caffe (JIA et al., 2014) library was used. It is an efficient library, whose core is implemented in C++ programming language, but there is a Python API that allows a quick and flexible development and integration with other packages and codes.

4.3.2 PARAMETERS ADJUSTMENT

Since the proposed method is based on the execution of two main algorithms, some parameters need to be defined. For DE we used a population of 50 individuals, the crossover rate was set to 0.9 and the mutation scaling factor (F) was set to 0.5. The stopping criterion was set as a maximum of 100 iterations.

The second algorithm is the k-means that is used for the fitness calculation. The initialization of the centroids was done using the k-means^{++} strategy. For each DE individual, the k-means algorithm was executed for 300 iterations, which demonstrated to be necessary to the convergence.

4.3.3 RESULTS

Two different dataset, Face and Pedestrian, were used to verify the method performance. For each dataset, two models mentioned were used to train the feature extractors. Considering that DE is an stochastic optimization algorithm, multiple runs were necessary to verify the repeatability. Therefore, for each dataset and each model, 10 independent runs were performed. Results are shown with the average and standard deviation for all runs.

The results obtained for the Faces dataset are shown in Figure 17. It is observed that both models achieved a reasonably good performance, but Model #0 can be considered better, since it produced a point in the ROC graph closest to the top-left corner. Regarding accuracy, Model #0 achieved $84.83\% \pm 1.30$ of correct classification, while Model #1 achieved $78.40\% \pm 3.57$. The results obtained for this dataset can be considered good, considering that the proposed approach is unsupervised, such that it learnt to find discriminating features capable of separating input images in adequate classes.

For the Pedestrian dataset, the results are shown in Figure 18. It is possible to notice that Model #0 achieved a better performance, since it near to the top-left corner of the graph. Regarding accuracy, model #0 achieved $69.67\% \pm 5.34$ and model #1 $64.25\% \pm 3.94$ of correct classification.

It is noticeable that, when comparing the results between Faces and Pedestrian datasets,

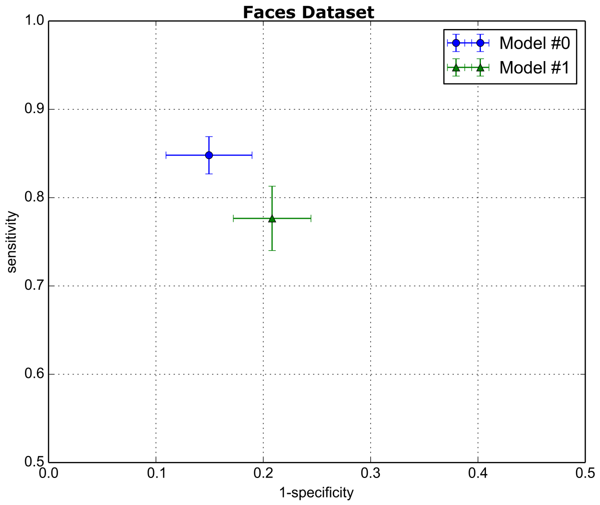

Figure 17: ROC graph comparing the performance of the two different models trained using the Face dataset.

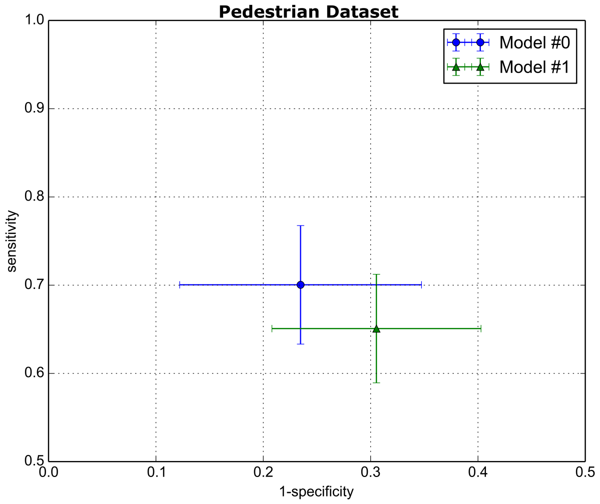

Figure 18: ROC graph comparing the performance of the two different models trained using the Pedestrian dataset.

the classification performance of both models have decreased. This can explained by the higher level of difficulty encompassed in the Pedestrian dataset when compared to the Faces dataset, since the variability of the input data is much larger, related to people appearance (clothing, accessories), poses and background. Seeking to find a quantitative explanation to support that assertion, we calculated the entropy, in the information theory sense, for a random sample of 2000 images of each class for both datasets, and the results are shown in Table 10.

Entropy can give some clues about the complexity of images, and we noticed that there is a difference of approximately 20% in the entropy obtained for both datasets, similar to the difference in the performance of the method.

Table 10: Mean and standard deviation entropy for samples of Faces and Pedestrian datasets, considering both classes (Foreground – object and Background – no object).

	Pedestrian		Faces	
	Foreground	Background	Foreground	Background
Mean±Std.dev.	8.84±0.42	7.57±1.29	7.12±0.34	5.93±1.15

We also performed a comparison between our method and two other feature extractors, to verify if the learned features could influence the performance of a supervised classifier. For both faces and pedestrian datasets, Model #0 was chosen to represent the proposed method against other feature extractors.

Since PCA is capable of producing a feature vector of arbitrary length, we decided to use the same size used for the proposed method. The vectorized images were used as input for PCA. Therefore, for the face dataset, each sample was transformed into a feature vector with 250 components. While, for pedestrian dataset, each sample is a 650 dimensional feature vector.

The parameters available in the literature were used to extract the HOG features. Therefore, we use a histogram with 9 bins to accumulate the gradients calculated using blocks with 4 cells and cells with 8×8 pixels. This generated a feature vector of 144 components for each sample of the face dataset and 720 components for the pedestrian dataset.

The feature vector generated by each extractor (our, HOG and PCA) were used as input to train two different classifiers, a linear SVM an a RBF SVM. Since there are some parameters to adjust in both kinds of SVM, a grid search strategy was used to find the suitable configuration. In this way, all combinations of feature extractor and classifiers had the same conditions during the training. Since SVM is a deterministic method, it always produce the same results given the same input. In this way, the training procedure was performed only once.

For face dataset the accuracy achieved for all six combinations of feature extractors and classifiers are shown in Table 11.

Table 11: Accuracy for face dataset in the comparison of feature extractors and classifiers.

	Acc(%)
HOG+SVM RBF	99.0
HOG+SVM LINEAR	98.15
OUR+SVM RBF	96.55
OUR+SVM Linear	94.0
PCA+SVM RBF	93.6
PCA+SVM LINEAR	88.5

For pedestrian dataset the accuracy achieved for all six combinations of feature extractors and classifiers are shown in Table 12.

Table 12: Accuracy for pedestrian dataset in the comparison of feature extractors and classifiers.

	Acc(%)
HOG+SVM RBF	90.4
OUR+SVM RBF	86.05
HOG+SVM LINEAR	81.22
OUR+SVM LINEAR	78.27
PCA+SVM RBF	72.32
PCA+SVM LINEAR	63.68

The ROC curves and the areas under the curves for face and pedestrian datasets, are shown in Figures 19 and 20, respectively.

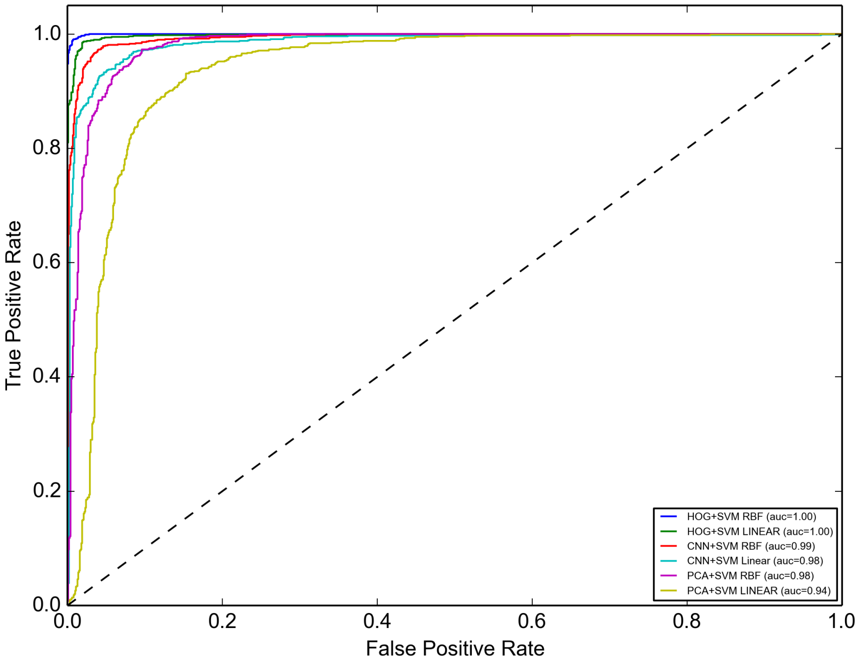

Figure 19: ROC curves and the area under the curve values for each combination of feature extractors and SVM classifiers for Faces dataset.

We also performed experiments using the soft biometric dataset, a more complex set of images. Similarly to Faces and Pedestrian dataset, we executed the unsupervised learning method using the two different models. Therefore, for each attribute (gender, upper clothes and lower clothes), two different models were learnt. For all three attributes the results obtained had a poor performance. Looking to the label associated with each sample and the separation achieved by the method, the accuracy was around 50%. This means that, for this kind of input

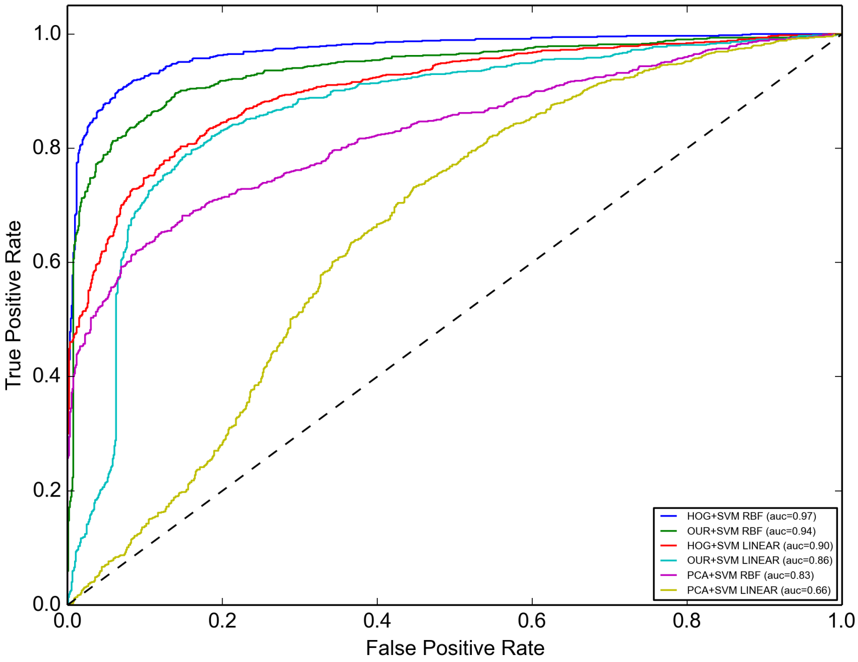

Figure 20: ROC curves and the area under the curve values for each combination of feature extractors and SVM classifiers for Pedestrian dataset.

image, the method was unable to learn good features capable to allow a good separation that could meet the human interpretation given to the images.

4.3.4 DISCUSSION

Engineering how to transform the data into a meaningful form is a complex but very important task. Automatic methods capable of extracting interesting features from input data are desired. Even more interesting, if these methods could deal with unlabelled data.

The proposed method here is based on three different pieces: an adjustable convolution network, a clustering technique and an optimization strategy. These components altogether allowed the construction of an unsupervised feature learning method. As results showed, for two datasets (Faces and Pedestrian), the method could produce promising results, achieving a reasonable performance of separation.

When compared to other feature extractors, the proposed method almost achieved the same performance of the HOG method. This is an interesting result, since our method is purely based on the raw input data. More important, the features were learned automatically by the computer. In this way, it is possible to say that the proposed method is capable of learning features in a way to facilitate the training of a classifier. It is important to notice that, since the proposed method is unsupervised, looking directly to the learned features could not make sense in the human point of view. Additional investigations should be conducted in the way to analyse and verify the learnt features.

Looking to the results of soft biometric dataset, the capability of the method were unsatisfactory. The main reason for this is related to the complexity of the separation task, and therefore to the semantic gap. It seems that extract details from images, such as gender and clothes types, demands a level of supervision, seeking to bridging the distance of low level data and higher level information.

4.4 SOFT BIOMETRICS EXTRACTION USING CNN

In this section we summarize the experiments used to validate the proposed method for person description based on soft biometrics, as discussed in Section 3.2.

4.4.1 IMPLEMENTATION

The CNNs were implemented using the Torch7 library (COLLOBERT et al., 2012) that is a very flexible and efficient package for implementing neural networks and machine learning algorithms. An important feature of this library is its capability for using General-Purpose computation on Graphics Processing Units (GPGPU). It provides an easy and almost transparent way to switch between Central Processing Unit (CPU) and GPGPU environments, allowing a significant reduction of development and training time of a CNN-based solution.

The hand-designed image descriptor was implemented using the OpenCV computer vision library, version 2.4.9.

Other classifiers mentioned in Section 3.2.3 were trained and tested using the Waikato Environment for Knowledge Analysis (WEKA) software, version 3.6.11 (HALL et al., 2009).

The running environment was a Linux-based desktop computer equipped with an Intel i7 quad-core processor, 32 GBytes of RAM and a Nvidia GTX 660 GPU board running CUDA 6.0. The CNN training process exhaustively used the GPGPU power, and the other classifiers took advantage of parallel processing by means of multi-threading.

4.4.2 HAND-DESIGNED IMAGE CLASSIFIER

Considering the methodology of the hand-designed image classifier described in Section 3.2.3, a factorial experiment was done for each classification problem: feature rankers (IG, R, KW) × number of features (10, 100, 200) × classifiers (C4.5, SVM, MLP). Considering the three problems: gender, top and bottom clothes, we performed $3 \times 3 \times 3 \times 3$ experiments. For each combination of options, a 10-fold cross-validation procedure was performed. Therefore, a total of 810 different runs were done. For the SVM classifier, a parameter grid search was conducted seeking to find a suitable parameter configuration. The mean accuracy and the standard deviation for gender, upper clothes and lower attributes are shown in Figures 21, 22 and 23.

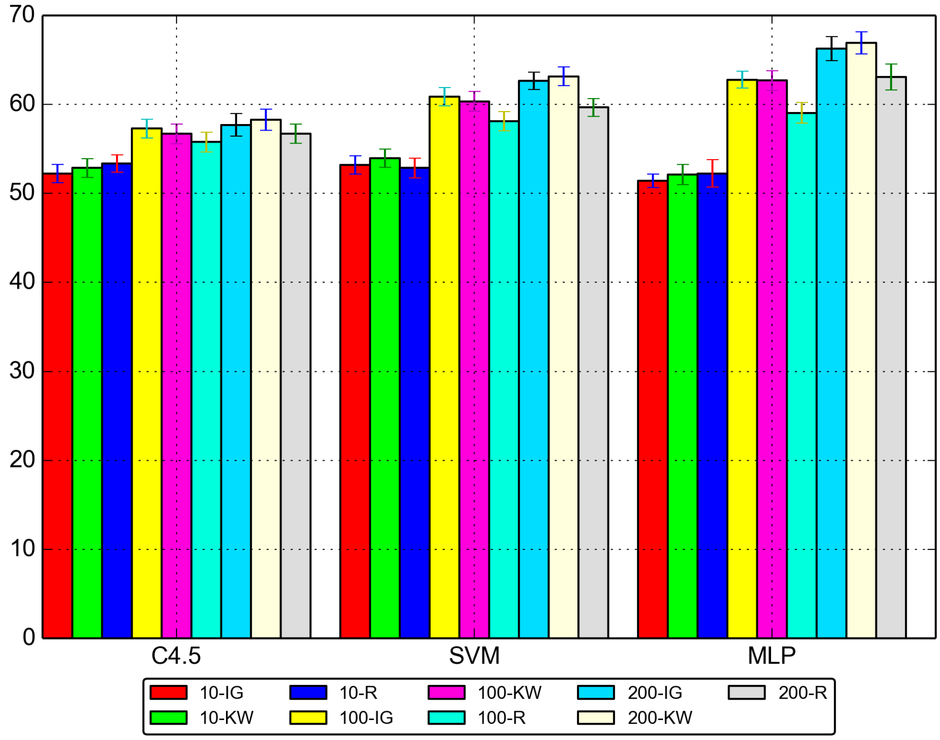

Figure 21: Comparison of hand-designed classifiers according to ranking methods and number of top features, for the Gender attribute.

It is noticed that the length of the feature vector has an important influence in the performance of the classifiers. As the number of features used increases, the overall performance also increases. For all the three classification problems, using only 10 features, results are close to random guess ($\sim 50\%$). On the other hand, the best results were achieved using the subset of 200 features. Analyzing the results from the point of view of the feature ranking methods, it is possible to see that the top features ranked by the KW method leaded to the best results. Also, the best-performing classifier was the standard MLP.

Therefore, we selected the configuration of top-200 features selected by KW classified

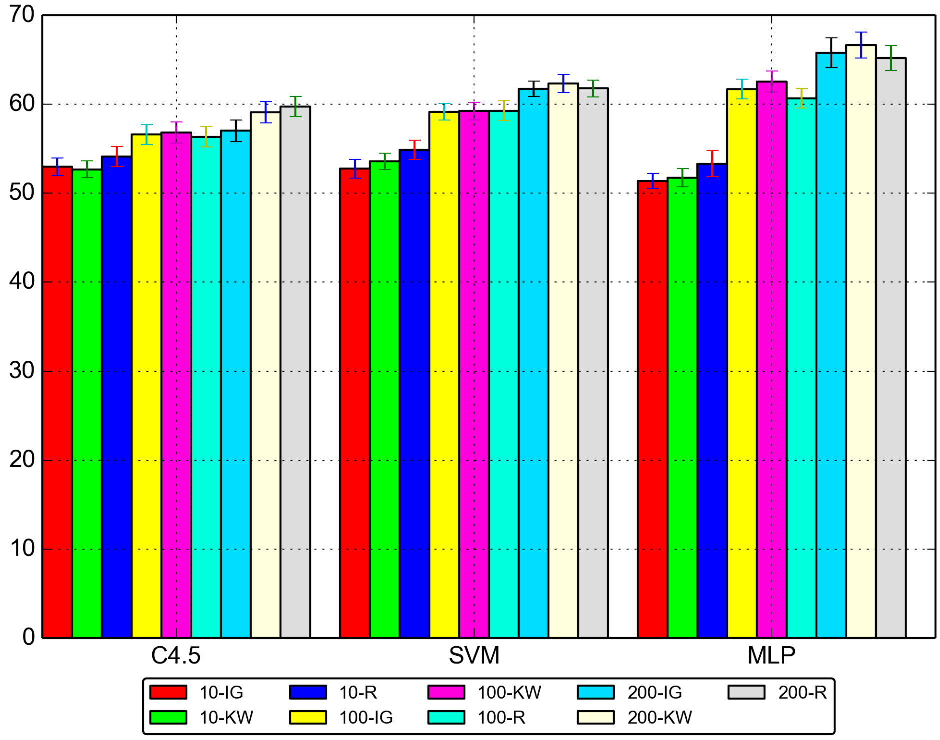

Figure 22: Comparison of hand-designed classifiers according to ranking methods and number of top features, for the Upper Clothes attribute.

by MLP to report the results in details, shown which are shown in Table 13.

4.4.3 CLASSIFICATION RESULTS FOR OPERATION MODE #1 (OM#1)

In order to visualize the evolution of the CNN's learning along the training process of the three different CNN classifiers, the average *Acc* in the 10-fold cross-validation process was plotted, for both, the training and the testing steps. The curves for gender, upper clothes and lower clothes attributes are shown in Figures 24, 25 and 26, respectively.

Observing these curves, it is noticed that the CNN has an interesting learning ability, since it was able to achieve a good classification performance. The usage of the image augmentation strategy at each training epoch introduces a challenge to the classifier, since it rarely will face the same training example multiple times. This seems to influence the behaviour of the classifier in a good manner, preventing it from overfitting. This fact can be seen since both learning curves are growing together most number of training epochs.

The final predictive mean accuracy achieved in the test by the three OM #1 CNN

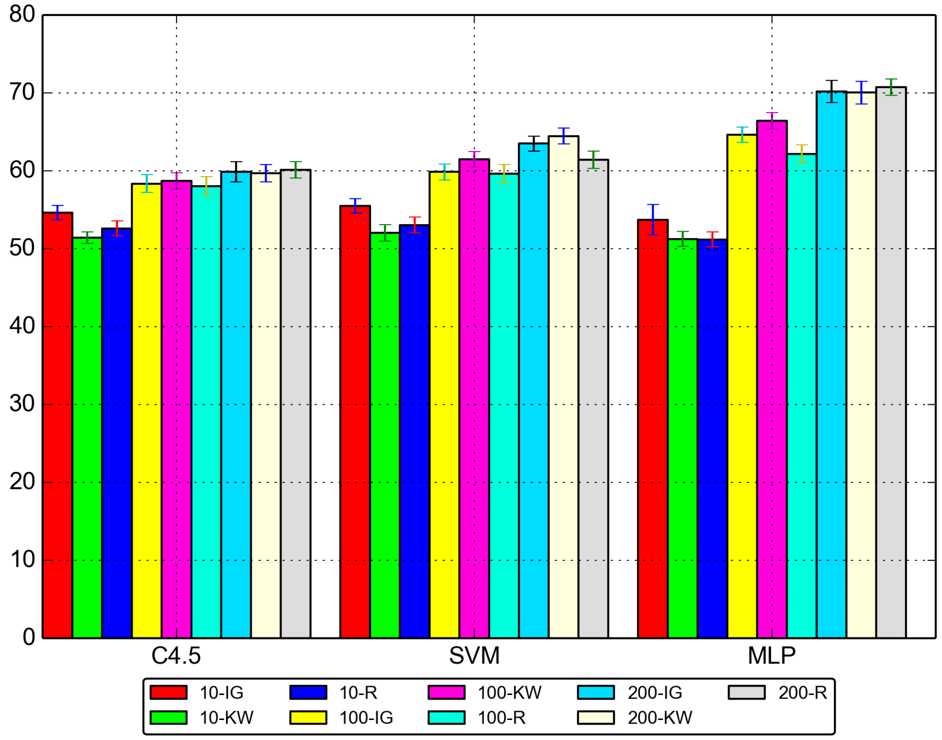

Figure 23: Comparison of hand-designed classifiers according to ranking methods and number of top features, for the Lower Clothes attribute.

classifiers is reported in Table 13. Observing the values shown, it is not clear to say which was the most suitable method for each attribute classified. Therefore, a statistical analysis was conducted.

Table 13: Mean Classification accuracy for the OM #1 and #2 CNNs and Hand-designed classifiers.

	CNN OM #1	CNN OM #2	Hand-Designed
	Acc(%) ± *std*	*Acc(%)* ± *std*	*Acc(%)* ± *std*
Gender	69.80 ± 1.16	61.04 ± 1.49	66.9 ± 1.50
Upper Clothes	80.65 ± 2.20	80.12 ± 0.63	65.87 ± 1.92
Lower Clothes	88.59 ± 1.34	85.82 ± 1.68	69.95 ± 1.71

For each method, the analysed distribution is composed by the accuracy of the 10-folds runs. In fist place, all distributions were tested to verify if they are a normal distribution. This was done by the Shapiro-Wilk test, using alpha level 5%. According to the results, all distributions were considered normal. This allows the use of the t-Student test to verify if the methods are statistically similar, using a confidence level of 5%. For each attribute classified, a paired comparison was performed.

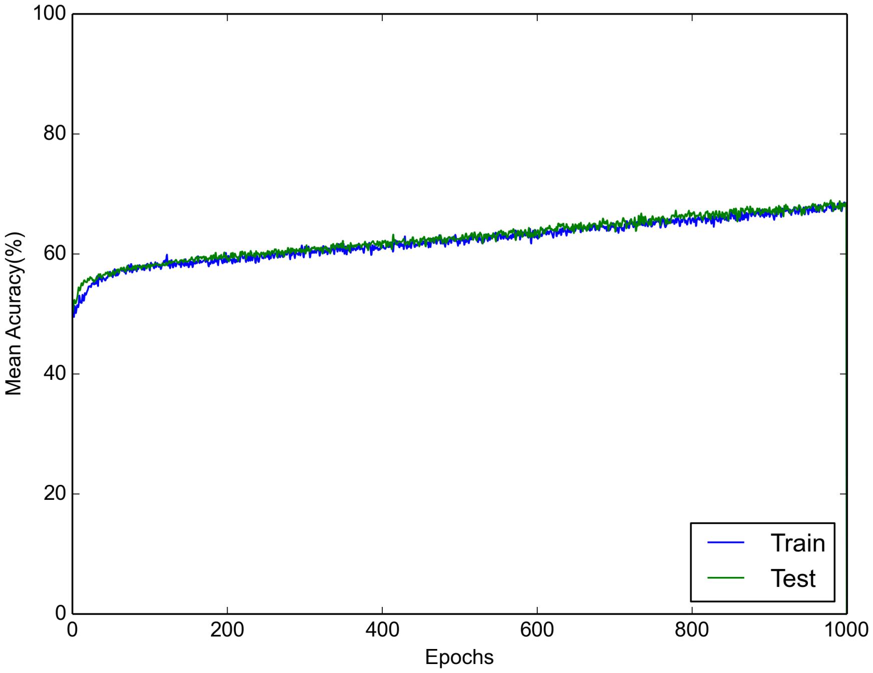

Figure 24: Accuracy curve during OM #1 CNN training for the Gender Clothes attribute.

For gender attribute and confidence level of 5%, the null hypothesis is rejected. Hence, there are differences between all three methods. To provide a visual analysis Figure 27, shows a box-plot chart of the three methods.

For upper clothes attribute and confidence level of 5% both CNN Model #1 and CNN Model #2 were considered statistically similar. To provide a visual analysis Figure 28, shows a box-plot chart of the three methods.

For lower clothes attribute and confidence level of 5%, all three methods were considered statistically different. To provide a visual analysis Figure 29, shows a box-plot chart of the three methods.

The top-10 highest output values for the best CNN trained for each class are shown in Figure 30. It is noticed that, in all classes a high variability is present in the images (poses, clothes appearance, background, etc). However, it is also noticed that the main characteristic of the classes is present. For instance, for clothes classification, gender information is not taken into account, since there are male and female individuals in the images. Another interesting detail is related to the results for short upper clothes. Among the results, three individuals are wearing long pants and short sleeves. This indicates that the CNN spotted the right information to make the decision.

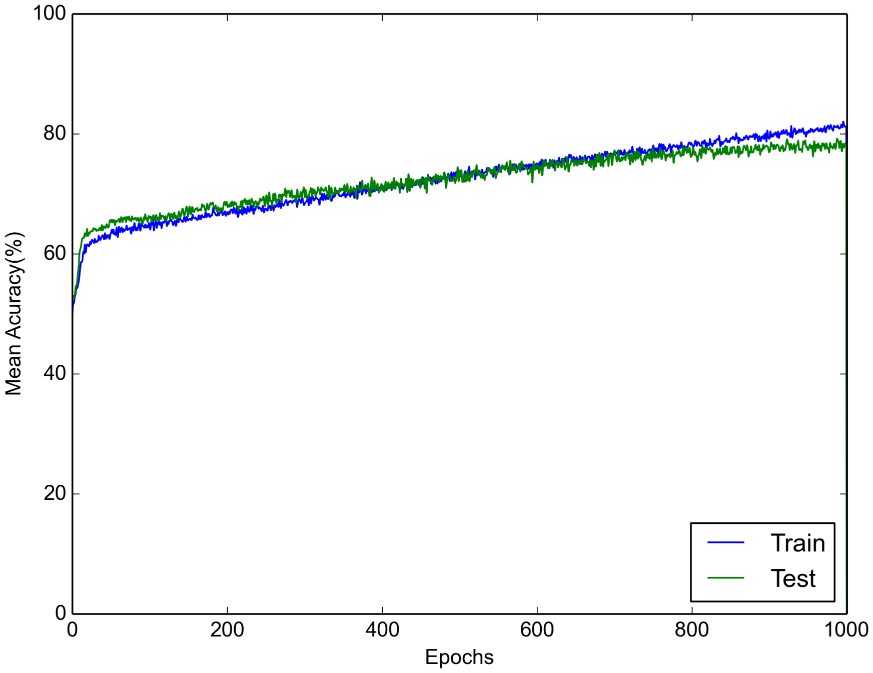

Figure 25: Accuracy curve during OM #1 CNN training for the Upper Clothes attribute.

Intend to complement the CNN performance analysis, the top-10 wrong classifications were taken and are shown in Figure 31. The first two rows shows the gender classification, been the first row men classified as women, and the second row women classified as men. It is difficult to find a common pattern that could explain the wrong results. A possible clue for woman classified as man, is that most of the individuals have short or covered hair. For clothes attributes most of the incorrect classifications are related to occlusion and changes in illumination. Additional analysis should be conducted seeking to a better understanding to the causes of incorrect classification.

The CNN classifiers were compared with other classifiers using the hand designed descriptors mentioned in Section 3.2.3. Observing the AUC values from the ROC curves of Figures 32, 33 and 34, it was found that CNN achieved better performance than the other classifiers, for the three different attributes.

Based on the values of AUC, the gender, as expected, seems to be the most difficult soft biometric to be classified, since, actually, it is also difficult in the real-world. All the classifiers tested produced poor results, even so, CNN was better than the other classifiers.

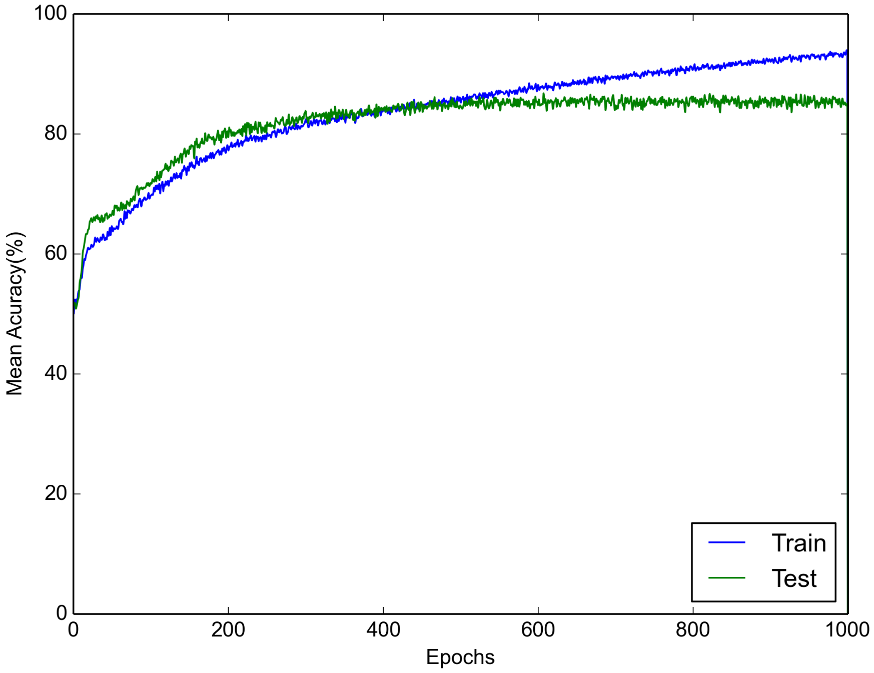

Figure 26: Accuracy curve during OM #1 CNN training for the Lower Clothes attribute.

4.4.4 CLASSIFICATION RESULTS FOR OPERATION MODE #2 (OM #2)

In OM #2, for each image instance presented, a single CNN should give values representing the classification for all three attributes (upper clothes, lower clothes and gender) at the same time. This is significantly more difficult than the previous case in OM #1 with three independent classifiers. In this experiment we seek to compare the performance of the two operation modes.

Similarly as before, the classifier was trained using a 10-fold cross-validation scheme. The results obtained, represented by the mean accuracy and the standard deviation for each attribute are shown in Table 13. We also plotted the ROC curves for each attribute obtained by both OMs in Figures 32, 33 and 34. These plots show that the results for both OMs are similar, for upper and lower clothes, suggesting that the CNN was able to learn how separate the classes simultaneously. The performance of OM#2 with gender attribute is worse than OM#1, which is an intriguing fact, and will be addressed in depth in the future.

4.4.5 DISCUSSION

Overall, the results of experiments show that the proposed CNN approach provided promising results, achieving a reasonable accuracy. For clothes classification performance,

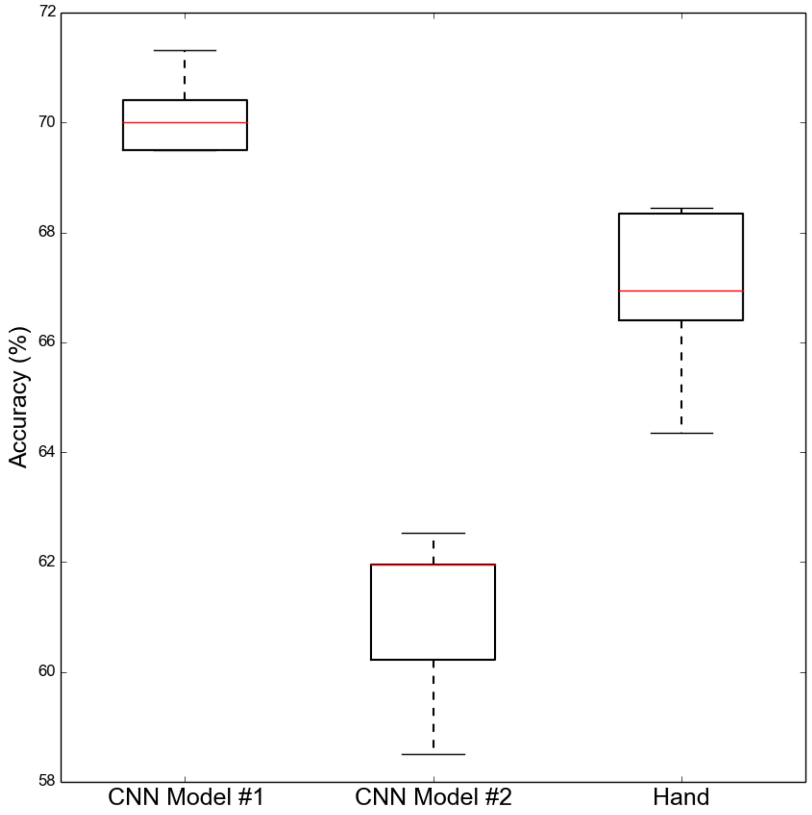

Figure 27: Box-plot chart showing the distribution of the three methods for the gender attribute.

the CNN approaches can be considered good. Notwithstanding, for gender, which is a very challenging problem, even for humans, there is still the need for further improvement.

It is impressive the CNN capacity to learn how to classify images, leading to extraction of high level semantics, based on raw images. Although it is a supervised method, and relies on the quality of the annotated data, it is able to process the data and reduce the semantic gap in the experiments performed. A possible explanation for this fact is related to the way that CNN operates. Its learning processing is based on input data and backpropagation, a relatively simple but very robust weight update rule. This data driven process allows the method to explore different possibilities and achieve better results.

Another interesting result is related to the performance of the CNN when trained in two different operation modes. Although, the final performance of both modes were similar for clothes classification, the computation effort demanded by OM#2 was smaller than by OM#1, since the classifier was trained to deal with three attributes at the same time.

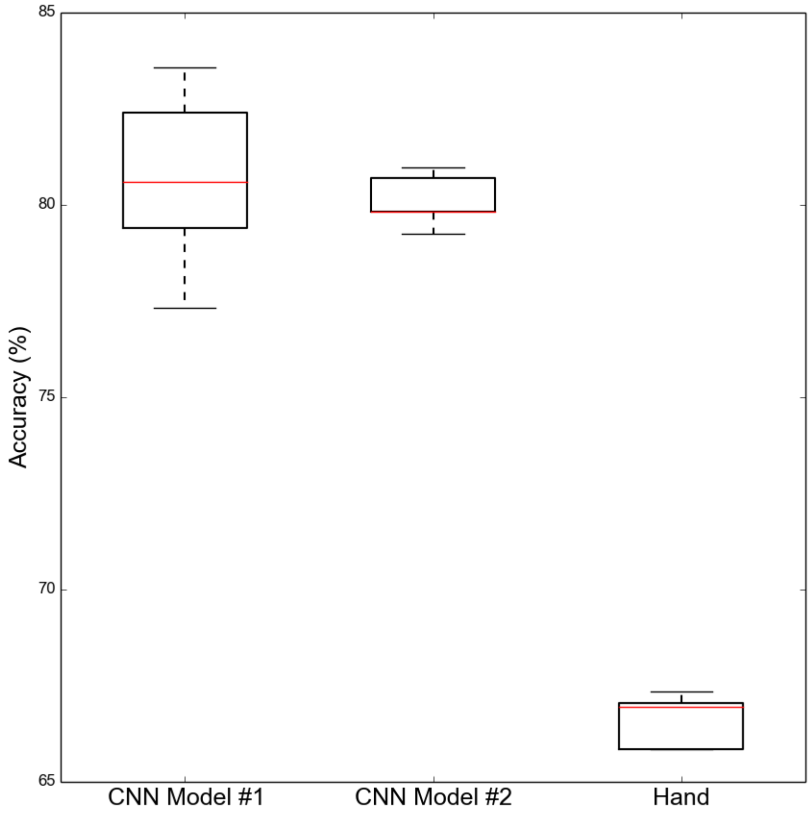

Figure 28: Box-plot chart showing the distribution of the three methods for the upper clothes attribute.

4.5 DESCRIBING VIDEOS USING SOFT BIOMETRICS

This part of the experiments is related to the description of videos contents using soft biometrics, in other words, describing people along a video using these attributes. The main objective of these experiments is to verify the performance of the proposed CNNs described in Section 3.3 to classify person attributes correctly in a sequence of frames. It is important to notice that is is assumed a pre-detected person, so person tracking was not done. To overcome this, bounding boxes provided manually in the dataset were employed.

An important aspect in this experiment is related to the CNN generalization capacity. This is done by training the CNN with a complete different dataset than that used for testing and evaluation. The soft biometric dataset was used as training set. The same procedure of data augmentation, using small random transformation in the images was applied seeking to increase the CNN learning capacity. For testing and validation the video dataset was employed.

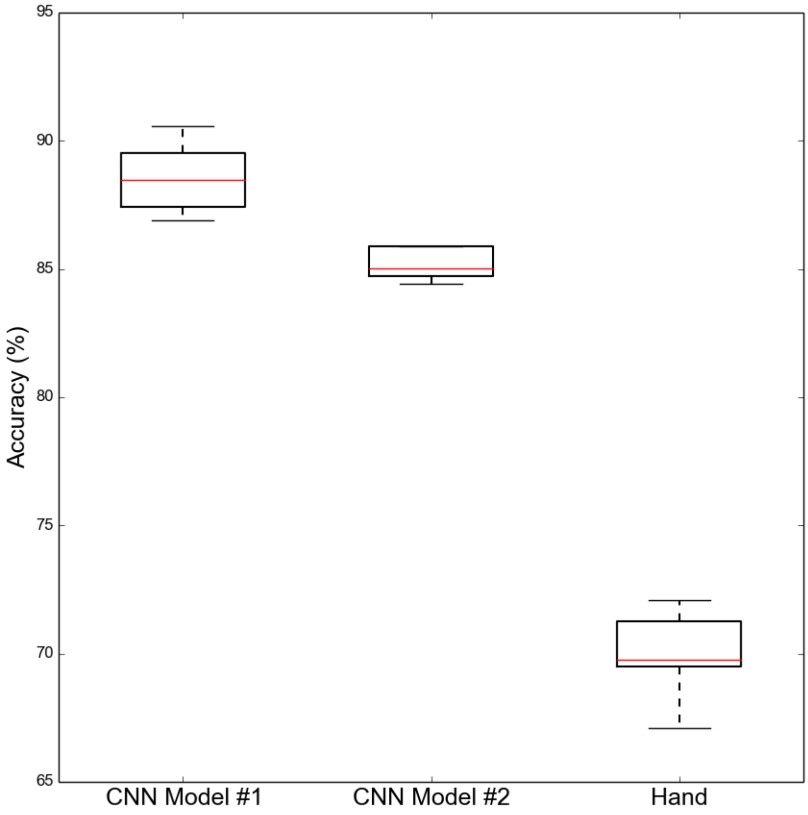

Figure 29: Box-plot chart showing the distribution of the three methods for the lower clothes attribute.

A possible way to improve results reported in Section 4.4 is to look at input data with a different perspective. Since the input data are images, this change is related to colorspace. In those experiments, images were coded using the RGB colorspace. But, color information is not an essential data piece to allow gender or clothes identification. For these experiments, all images were transformed from RGB to the YUV space.

The evaluation protocol is similar to that employed in Section 4.4, where ACC, ROC curves and AUC were used to verify the model performance.

4.5.1 RESULTS

The results obtained are reported in three different parts. The Subsection 4.5.1.1 shows the ROC curves and AUC given during the training procedure. Based on these plots, the models with higher AUC for each soft biometric were selected to be used in the validation procedure, reported in the Subsection 4.5.1.2. Results for temporal correction are shown in Subsection

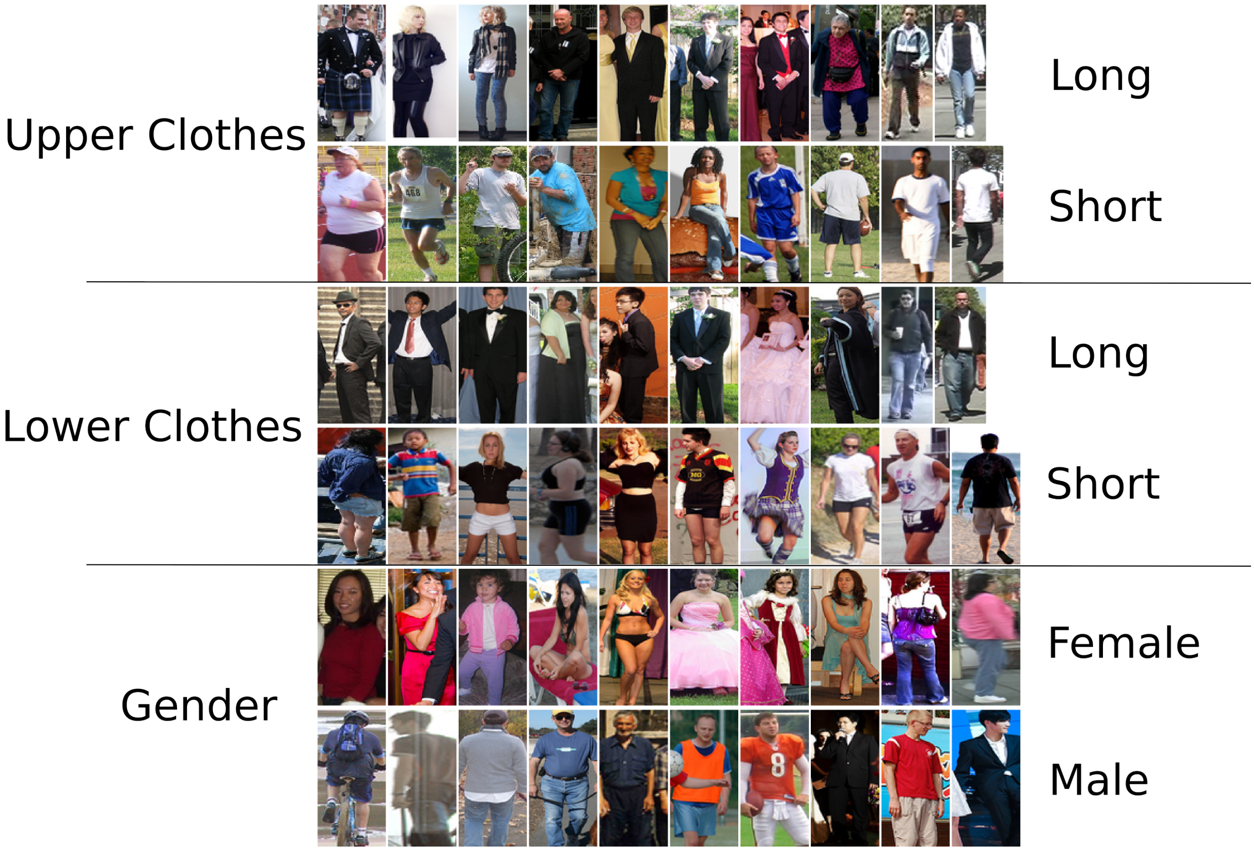

Figure 30: The top-10 OM #1 CNN results for each class. ***First and Second Rows:*** **samples classified as Long and Short for Upper Clothes.** ***Third and Fourth Rows:*** **samples classified as Long and Short for Lower Clothes.** ***Fifth and Sixth Rows:*** **samples classified as Female and Male for Gender.**

4.5.1.3.

4.5.1.1 TRAINING PROCESS

As mentioned in Section 3.3, four different models were proposed as possible classifier for each soft biometrics. For the training process, the same parameters detailed in Section 4.4 were used. Results, regarding the accuracy and standard deviation, for the testing set for each model per soft biometric are showed in Table 14.

Table 14: Mean Classification accuracy for the four different models.

	Small	Medium	Large	Very Large
	Acc(%) ± *std*	*Acc(%)* ± *std*	*Acc(%)* ± *std*	*Acc(%)* ± *std*
Gender	73.61 ± 2.02	75.48 ± 1.04	72.70 ± 3.15	76.09 ± 1.39
Upper Clothes	70.96 ± 4.38	72.04 ± 2.90	73.95 ± 2.80	70.87 ± 4.04
Lower Clothes	85.67 ± 0.73	85.58 ± 0.78	86.60 ± 0.77	85.39 ± 0.81

A statistical analysis were performed seeking to verify if there significant similarity

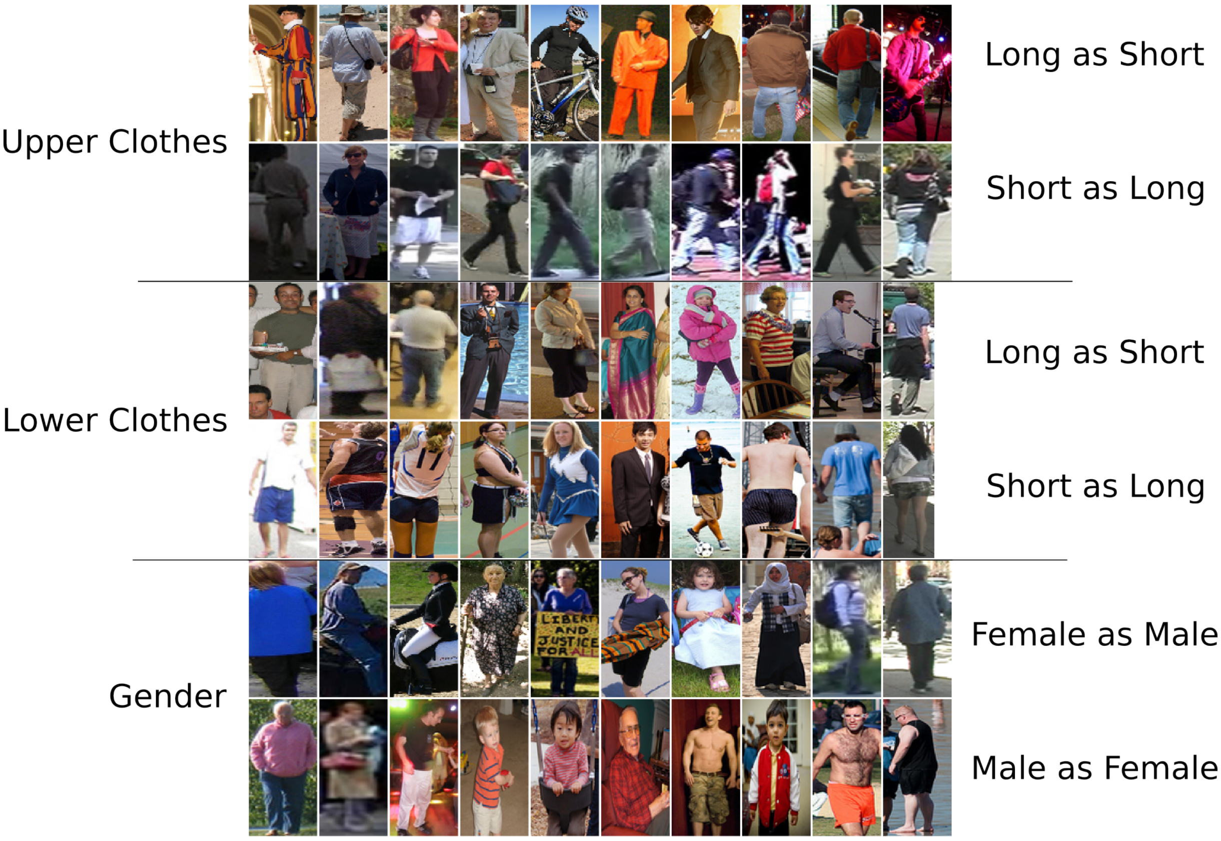

Figure 31: The top-10 OM #1 CNN wrong results for each class. ***First and Second Rows:*** **samples classified as Female and Male for Gender.** ***Third and Fourth Rows:*** **samples classified as Long and Short for Upper Clothes.** ***Fifth and Sixth Rows:*** **samples classified as Long and Short for Lower Clothes.**

or differences between each model per attribute. The achieved accuracies were used in these tests. The Shapiro-Wilk test, using alpha level 5% was used to test the distributions normality. Accordingly to the results, all distributions were considered normal. This allows the use of t-Student test to verify if the methods are statistically similar, using a confidence level of 5%. For each attribute classified, a paired comparison between each model was performed. Based on the results, all methods for both gender and lower clothes attributes were considered statistically similar. For upper clothes attributes, models Small and Large were considered statistically different, while other combinations were considered similar. For a graphical aid, the boxplot graphics for gender, upper clothes and lower clothes attributes, are presented in Figures 35, 36 and 37, respectively.

The results were also consolidated into ROC curves, seeking to analyse their performance. For gender extraction, the curves obtained are shown in Figure 38. Analysing the ROC curves and the AUC values it is possible to infer that the VeryLarge model produced the best results, using a feature vector of 1152 components. This is a coherent result, suggesting that for this instance of the problem, a larger feature vector leads to better classification.

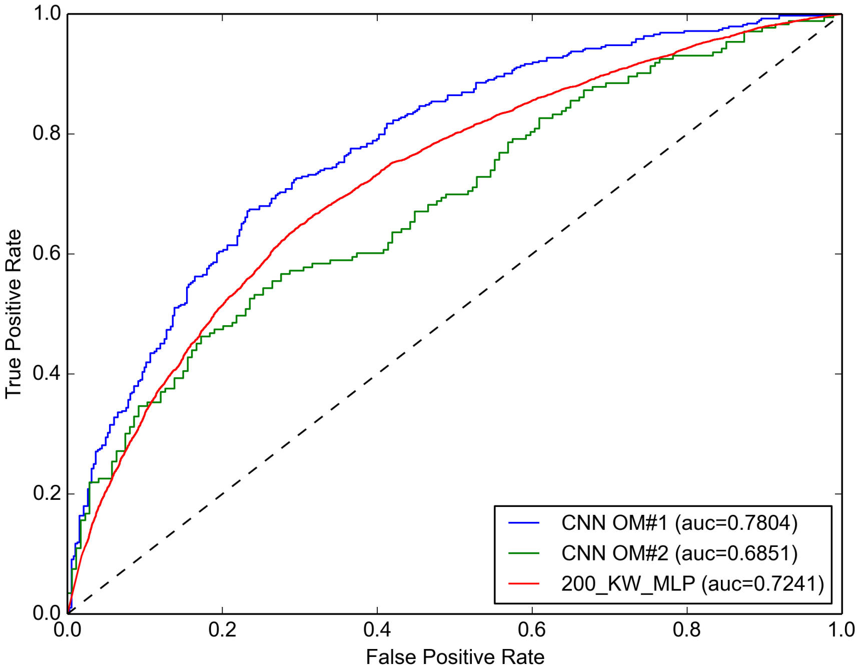

Figure 32: ROC plot comparing the OM #1 CNN, OM #2 CNN and the best Hand-designed classifier for the Gender attribute.

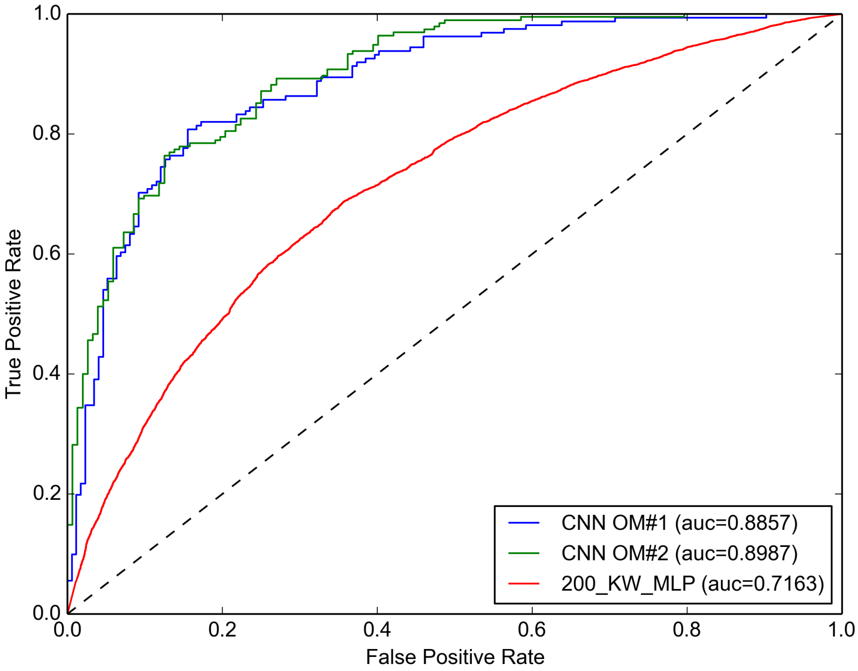

Figure 33: ROC plot comparing the OM #1 CNN, OM #2 CNN and the best Hand-designed classifier for the Upper Clothes attribute.

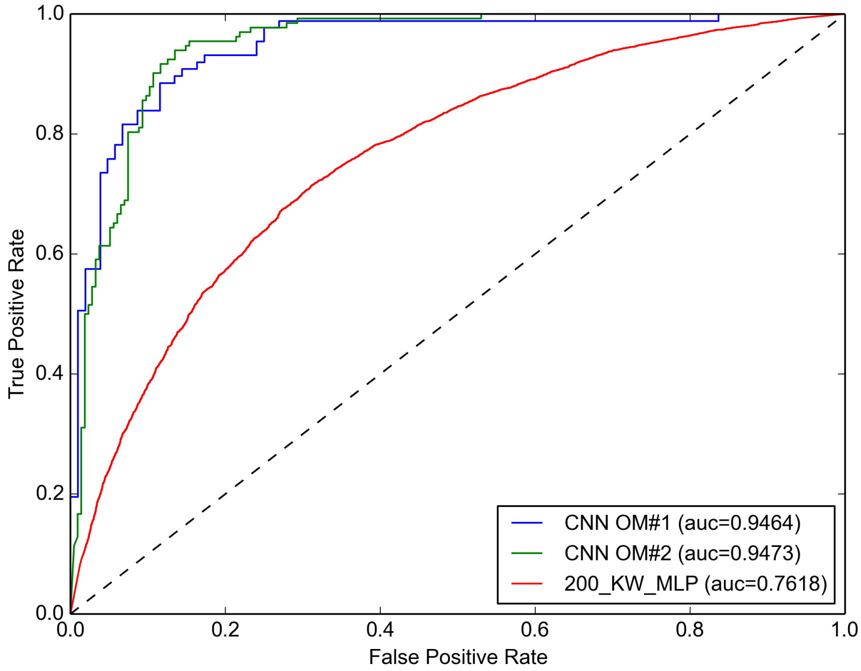

Figure 34: ROC plot comparing the OM #1 CNN, OM #2 CNN and the best Hand-designed classifier for the Lower Clothes attribute.

Performance classification for both lower and upper clothes, respectively, are shown in Figs.39 and 40. Analysing both graphs it is possible to see that, differently from gender, the size of feature vector do not seems to influence the classification results. A possible explanation for this behaviour could be related to the complexity of the problem. Classifying clothes information based on the appearance can be considered less hard than classifying gender and requires a small feature vector. A better comprehension of this behaviour requires further investigation.

4.5.1.2 VALIDATION PROCESS

Based on the ROC curves obtained during the training process, the models with the highest AUC values were chosen as classifiers for the validation process. Furthermore, ROC curves were used to choose an adequate threshold value for each soft biometric. This was done by selecting the curve point closest to the left upper corner, or point $(0, 1.0)$. Table 15 shows the thresholds chosen for each attribute. These values were used in the validation process.

All frames of each 124 subjects from SAIVT dataset were classified by the three models, one for each soft biometric. To help visualization and analysis of the validation process, accuracy for each subject per attribute were grouped into a single bar chart. A line, representing

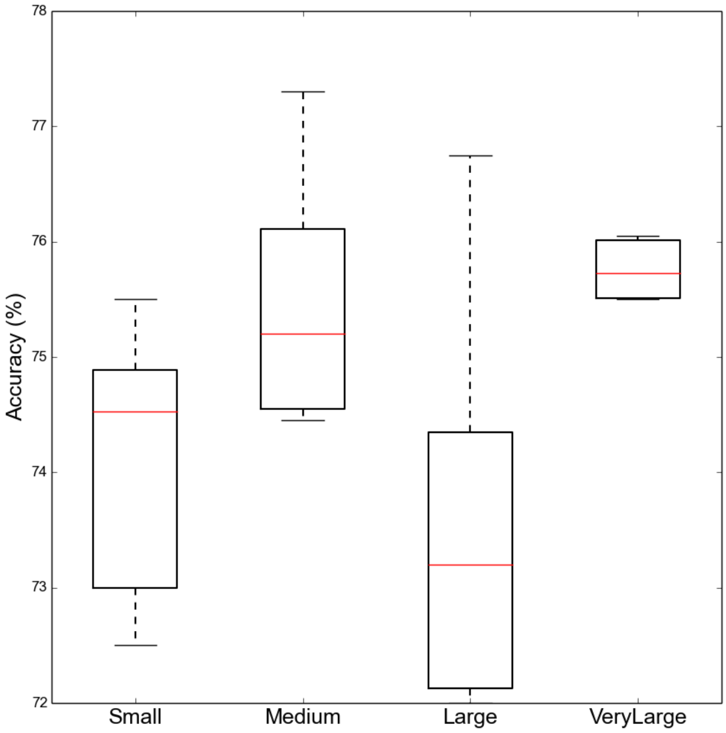

Figure 35: Boxplot graph regarding comparation of four different CNN models for gender attribute.

Table 15: Threshold values for each soft biometric.

Soft Biometric	Threshold
Gender	0.498
Lower Clothes	0.539
Upper Clothes	0.496

the mean accuracy for all subjects, also appears in each chart of Figures 41, 43 and 45.

Observing the classification results for gender attribute in Figure 41, the mean accuracy was 80.701% ± 31.891. Is possible to notice that for most cases, the model predicted its gender with an accuracy higher than the mean. This can be considered a satisfactory for gender classification. In Figure 42 shows individuals where the model was unable to correctly classify. It is possible to notice that most people is facing against the camera and have short hair. This two fact could be a possible reason to the model failure.

For upper clothes, the results obtained are shown in Figure 43. For this attribute, the classifier produced a mean accuracy of 75.529% ± 37.181. Analysing the histogram, again, the overall performance is considered satisfactory. When comparing with the other two attributes, it is clearly seen that this was the worst soft biometric classified. A possible reason could

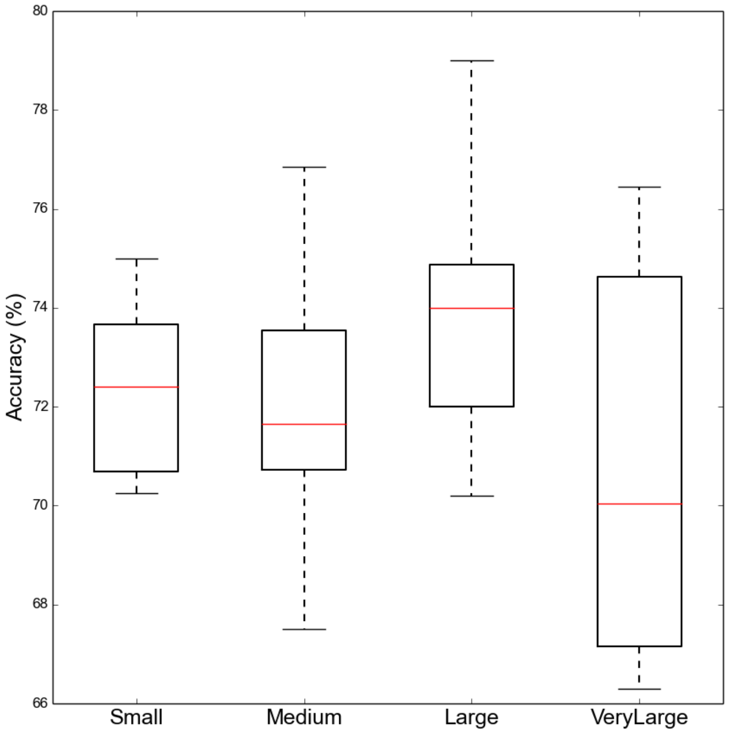

Figure 36: Boxplot graph regarding comparation of four different CNN models for upper clothes attribute.

be related to the camera position, which has a direct influence in how subject's appearance is captured. This fact can be observed in Figure 44, where samples of individuals where the model had zero accuracy. It is possible to see that most people is heading opposed to camera, which also could be a fact that influence negatively the model classification capacity. Another fact is the presence of self occlusion, which also could influence negatively.

When dealing with lower clothes, the model produced a mean accuracy rate of 87.544% ± 27.289, as shown in Figure 45. The overall performance of this model can be also considered satisfactory. In Figure 46 shows selected individuals for which the model failed to produce a correct classification. A possible reason for that is the presence of self occlusion, leading to losing information and consequently influence negatively the model capacity.

4.5.1.3 TEMPORAL CORRECTION

Dealing with classification in videos allows exploiting the temporal dimension as a way to correct classification errors. There can be changes in pose, illumination or partial occlusion, leading to a temporary incorrect response by the classifier.

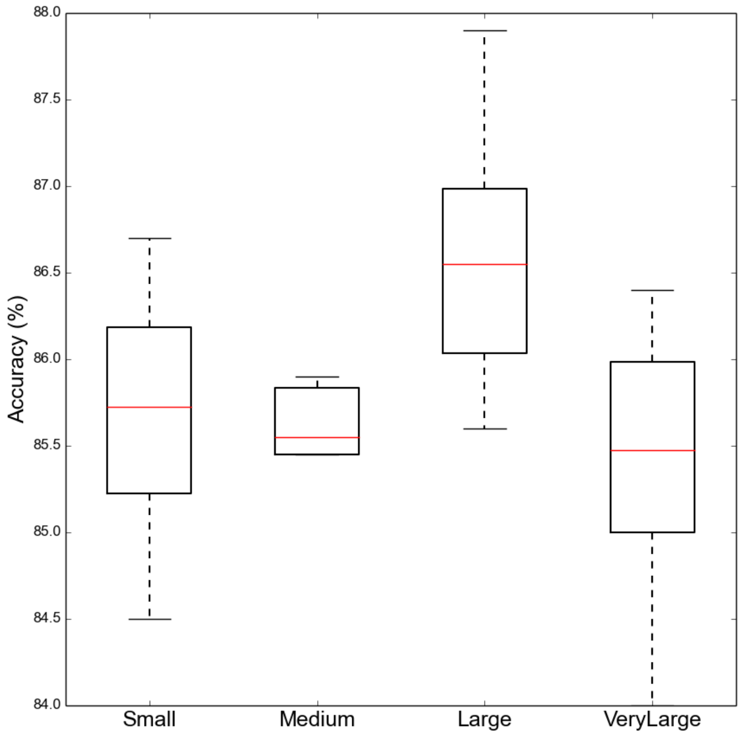

Figure 37: Boxplot graph regarding comparation of four different CNN models for lower clothes attribute.

As proposed in Section 3.3, mean filtering could be a possible solution to improve results. This is done by looking back in time, using previous outcomes to smooth the actual output. The filter is controlled by a window size, which determines how many previous times steps are used. Here, windows with $(5, 10, 15)$ time steps were used. A drawback of this operation is that it requires the storage of a buffer before applying the filter. This is needed because of the window size, since it is only possible to calculate the mean before a given number of frames.

Some experiments were done to verify if the application of the mean filtering could lead to better classification results. After the filter outputs for each subject frame sequence, a new mean accuracy for each soft biometric was calculated. These new values, along the standard deviation, are presented in Table 16. Each column represents the mean achieved using a different window size. The mean accuracy achieved without filtering is presented in the column identified as 0.

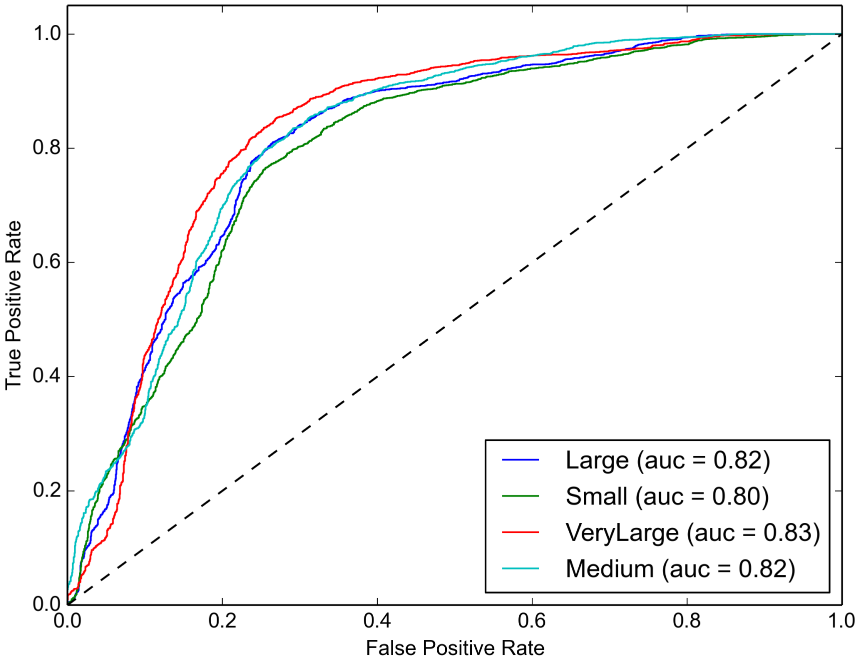

Figure 38: ROC curves and AUC values for each model proposed for gender classification.

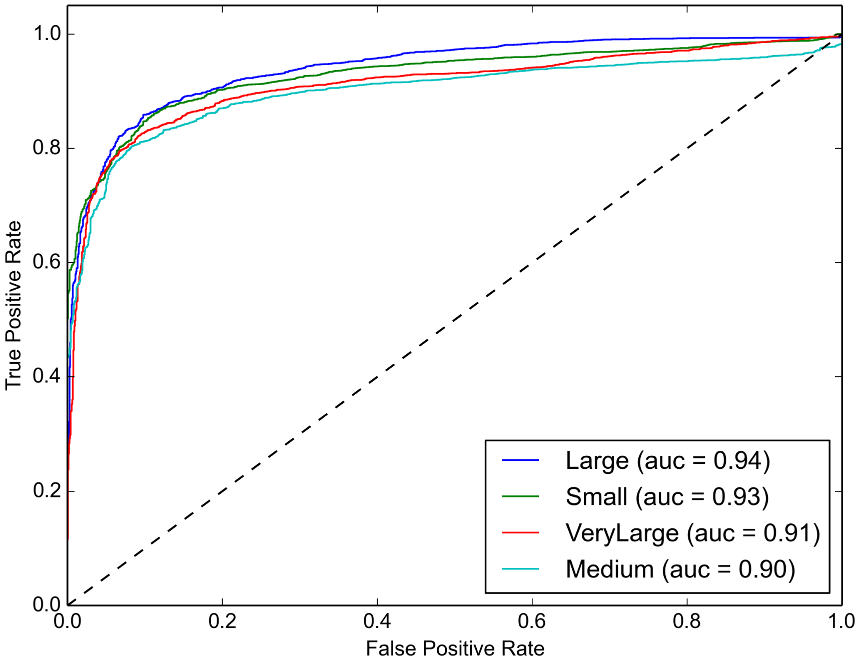

Figure 39: ROC curves and AUC values for each model proposed for lower clothes classification.

4.5.2 DISCUSSION

Describing images and videos seems to be a complex and important problem to be approached. Achieving this goal using automatic methods that can learn directly from data is

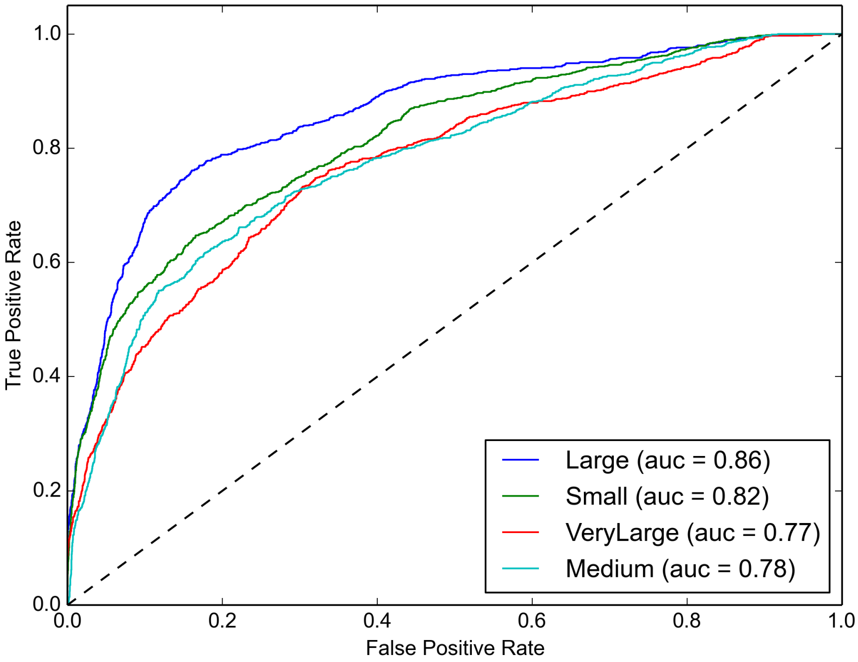

Figure 40: ROC curves and AUC values for each model proposed for upper clothes classification.

Table 16: Mean accuracy values and standard deviation for temporal mean filtering for each soft biometric.

	Window Size			
Soft Biometric	0	5	10	15
Gender	80.701±31.891	80.423±32.508	80.631±32.953	80.964±33.075
Lower Clothes	87.544±27.289	87.832±27.923	88.089±28.690	88.400±28.816
Upper Clothes	75.529±37.181	75.420±38.158	76.209±38.450	76.365±38.406

even better. In this set of experiments, CNNs were used as end-to-end classifiers to describe people extracted from video frames. Analysing the results, it is possible to say that the method provides a reasonably good performance. It was able to insert correctly into raw images, soft biometric information, with an accuracy over 70%.

When looking at the learning process the results are even more impressive. This is because training and testing data were taken from complete different sources. Training data was composed by images from three different datasets available, while testing data was taken from another available video surveillance dataset. This fact indicates that CNN has an good generalisation capability, dealing with different data distributions. When analysing the results visually, is possible to infer that the method has some limitations regarding to self occlusion and head direction. These limitations could be alleviated by using a more representative training dataset, which contains examples under such transformations.

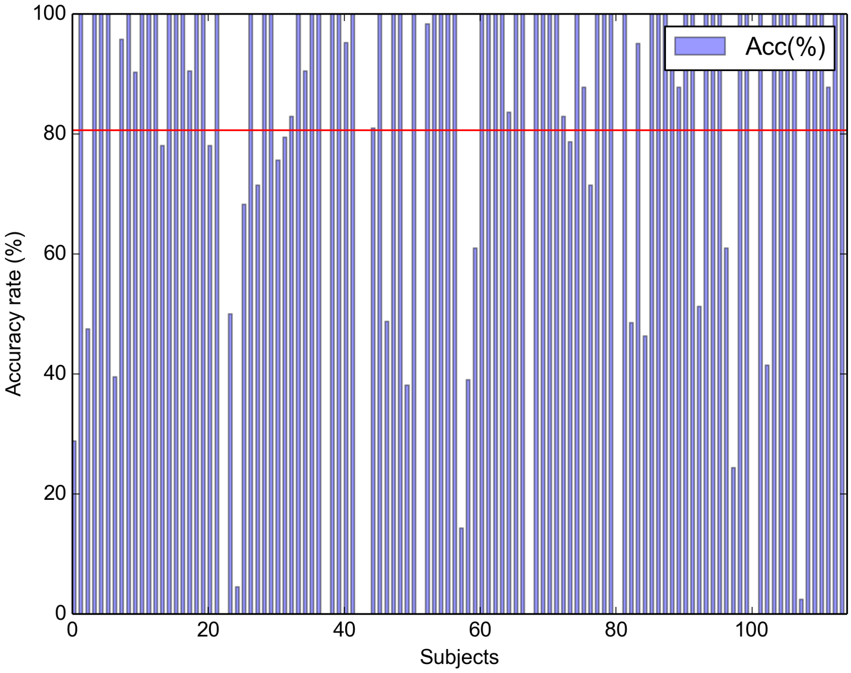

Figure 41: Accuracy per subject for gender attribute predicted by VeryLarge model. Red line represents mean accuracy among all subjects.

Temporal dimension was less explored in these experiments. The main reason for this, is that, in our opinion, temporal data is not essential to allow the description of the desired information. A tentative to use information over time leaded to insignificant improvement of the performance. A possible way to take advantage of the temporal component is to develop a better method for analysing and correcting past prediction results based on the accumulated knowledge.

As stated, the proposed method was developed using an extracted video patch containing a person. In this way, another improvement is to expand the proposed method to be able to insert soft biometric information while performing detection and tracking of people in a video.

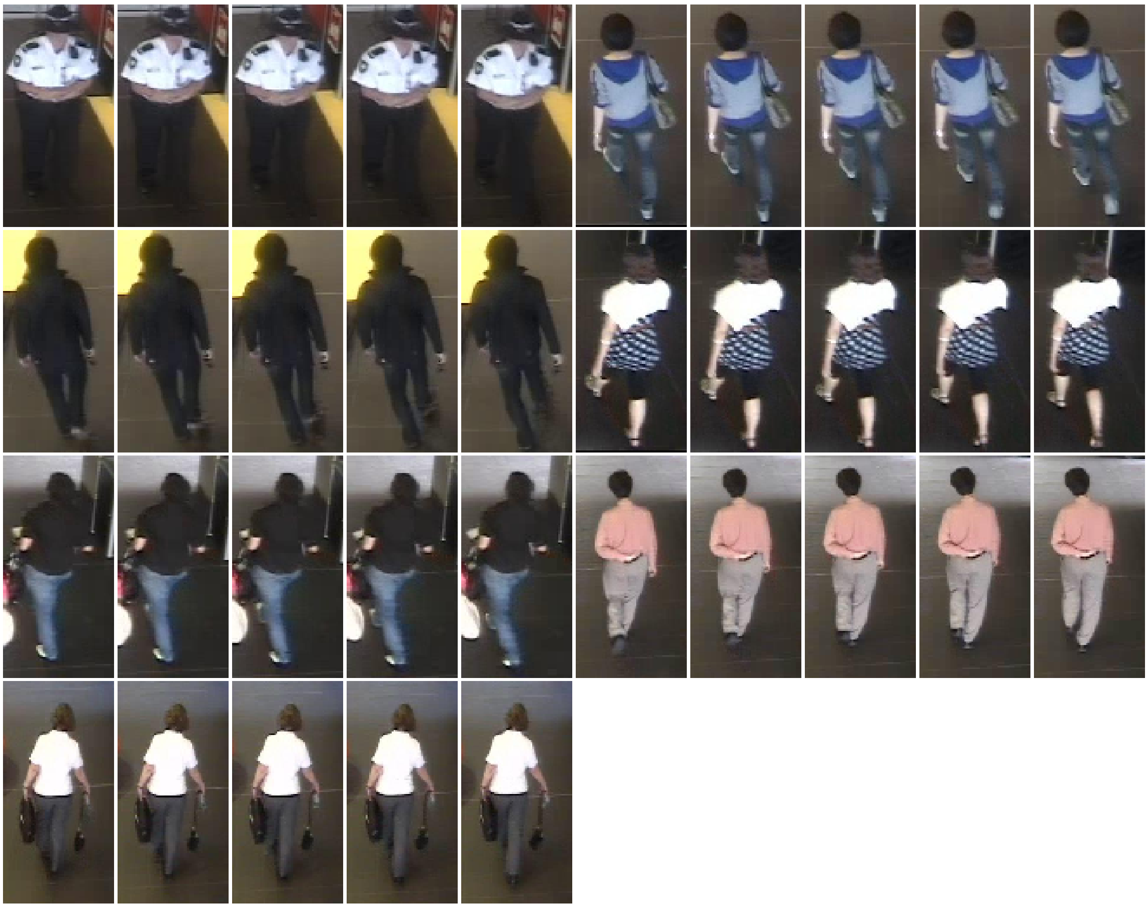

Figure 42: Individuals where the classifier produce an incorrect result for the gender attribute.

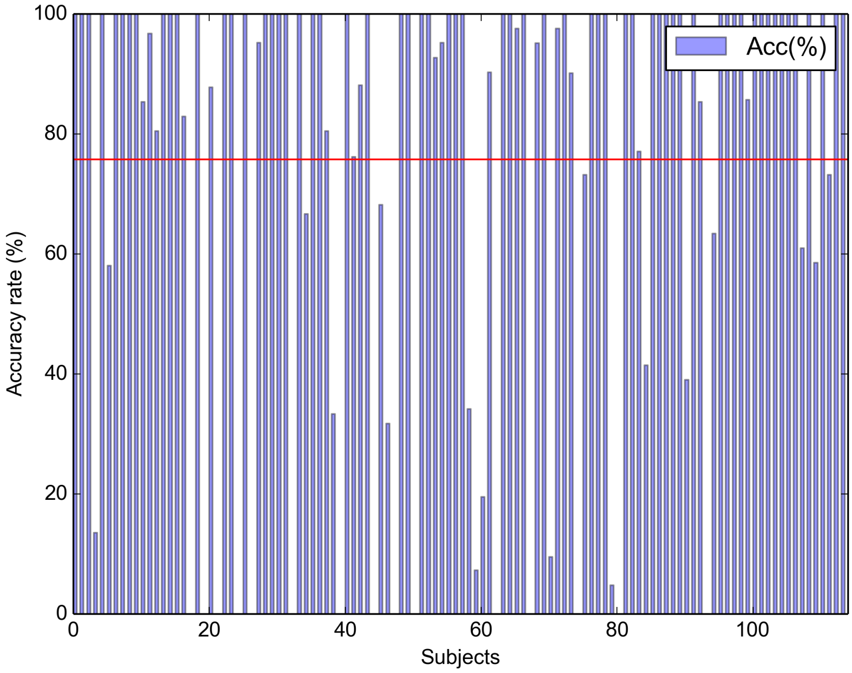

Figure 43: Accuracy per subject for upper clothes attribute by Large model. Red line represents mean accuracy among all subjects.

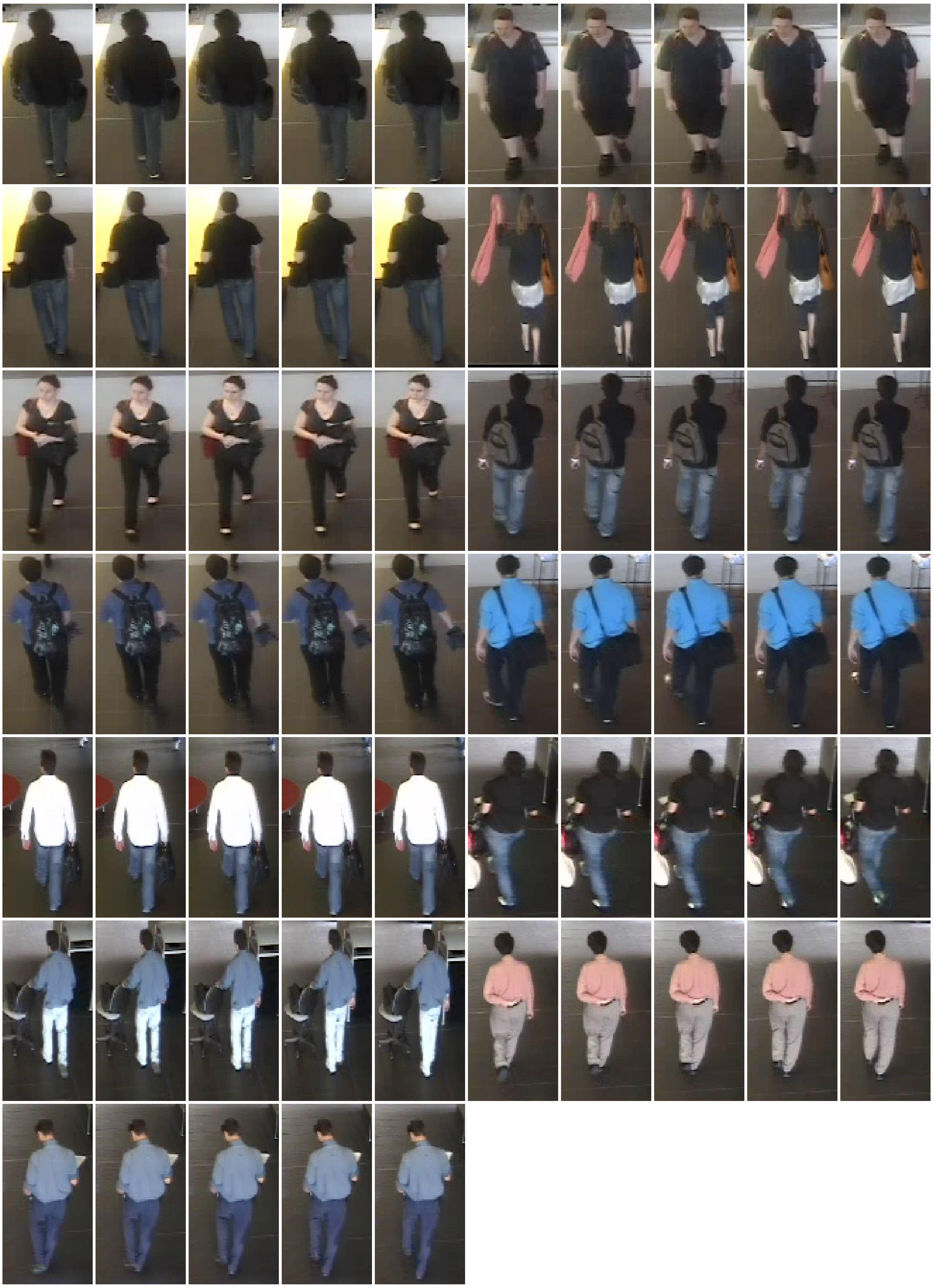

Figure 44: Individuals where the classifier produce an incorrect result for the upper clothes attribute.

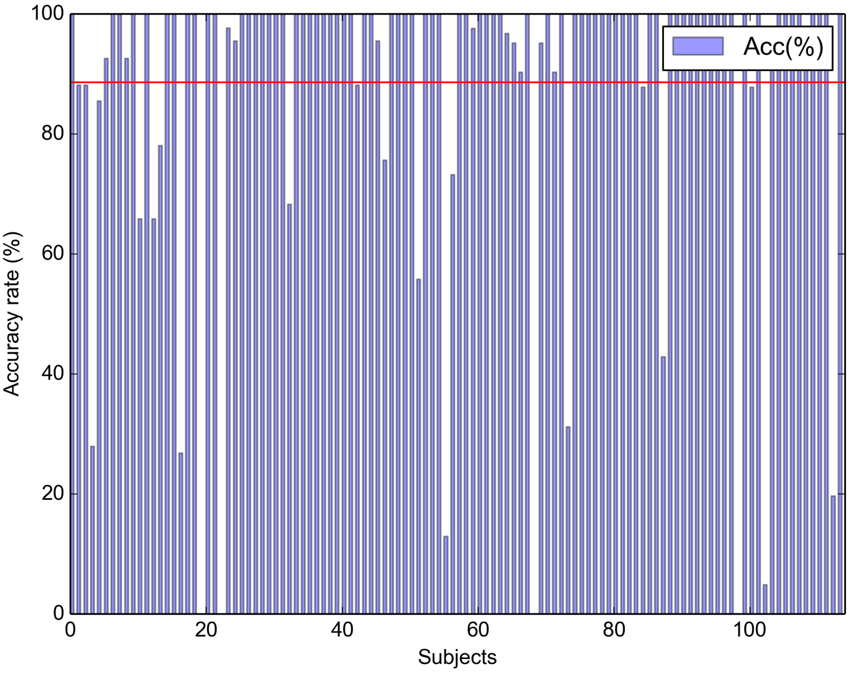

Figure 45: Accuracy per subject for lower clothes attribute predicted by Large model. Red line represents mean accuracy among all subjects.

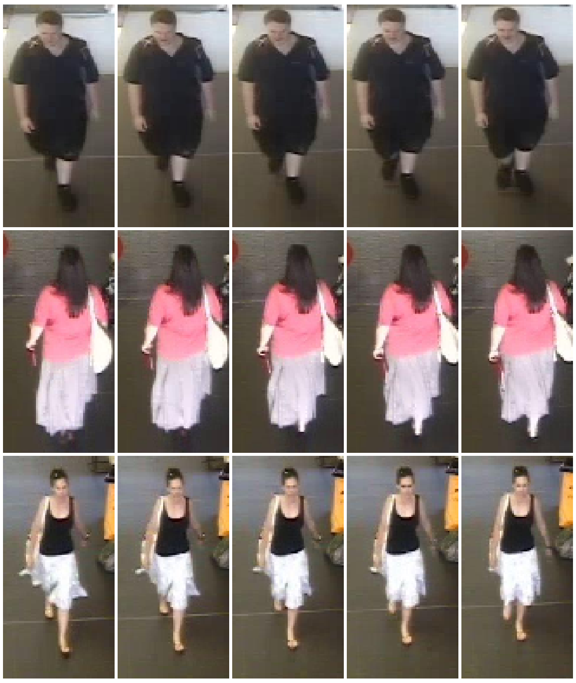

Figure 46: Individuals where the classifier produce an incorrect result for the lower clothes attribute.

5 CONCLUSION

Visual information could be seen as the main way that humans receive information from the surrounding world. As a famous saying states ”An image is worth a thousand word“. To choose an image to represent a sentence or to choose a sentence to describe an image, both can be cumbersome tasks. Frequently, an image can have different meanings to different people who observe it. And, a sentence can be interpreted differently by different readers.

Consequently, developing computational methods capable of describing an image similarly as humans do is a rather complex task. In the computer vision area, the automatic association of high-level information with visual data remains a challenge, since no method has yet reputed as a gold standard for most problems. Developing such methods could open new frontiers related to the usage of images and videos, both stored or processed in real time. Safety and surveillance are applications that could take great advantage of automatic high-level interpretation.

The big challenge to be tackled is to find a correspondence between low level image components and high level semantic meaning, reducing the so called semantic gap. Humans are very efficient to extract low-level information from images, such as borders, texture, color and shapes, and combine them into higher level elements, such as patterns, and finally, give meaning to these elements. Such automatic procedure, easily accomplished by humans, is clearly dependent on our capability to learn, categorize, store and retrieve information. However, there is still much to research on how humans interpret an image and extract meaning from it. This is particularly true from the point of view of Computer Science, since the computational methods currently in use are too shallow to emulate accurately the human behaviour in some tasks. Aiming at contributing to the development of such methods, this thesis took two different approaches.

In the first study, seeking to learn feature extraction from input data using an unsupervised way, the method proposed in this work takes the advantage of three components: an adjustable feature extractor, provided by the CNN; a feature vector quality metric, accomplished by the *k*-means clustering algorithm; and the global optimization to adjust the weights of the feature extractor, here performed by the Differential Evolution algorithm. The framework was designed to automatically discover semantic patterns from raw input images without label information, and so, it could be seen as a step towards an automatic annotation system for images.

Summarizing, what is expected from the framework is "to learn how to separate image data into classes just looking at the data, without any other external information". Once this is done, an important issue could raise: is all the information (edges, textures, shapes and higher level abstractions) required to perform the separation self-contained in the images? The results shown in Section 4.3 for the two datasets tested (Faces and Pedestrian) suggest that those images really convey the underlying information that is sufficient for separating them. The framework was able to adjust a feature extractor, which generates feature vectors that allow the separation of the images into classes coherent with the natural meaning to humans. It is important to notice that the method was able to separate the images, but not classify them, since classification requires a previously defined label, which is related to the human meaning to a set of images. Indeed, this is an important fact to be analysed. Since the method does not have any information about the semantics related to the images, it may find a separation not clearly meaningful for humans. The results of the above-mentioned experiments indicate that the relationship between the low-level information contained in images and its high-level interpretation is still an open issue. Additional investigations shall be done seeking to keep reducing the semantic gap.

The second study of this thesis was focused on supervised learning, designed to extract soft biometric information from human images. A sort of human-centred data is soft biometrics, which is related to the appearance and attributes of a person. The development of methods to extract and classify this type of data allows a better usage of the vast amount of stored images and videos, making it possible to search by the contents of the image. This kind of information is very difficult to be correctly annotated and classified, not only due to its large variability, but also, due to its multiple semantic meaning.

In recent years, several methods were proposed in the literature using many preprocessing tools, feature extraction and classifiers. Choosing the right combination of these methods and the correct tuning of parameters is a hard task, and it is usually done by a trial-and-error process. This procedure demands intensive computation and the ability of experts to adjust the parameters of the system, as we showed in Chapter 4. On the other hand, results of our experiments suggest that CNN is adequate do classify soft biometrics. The performance of a CNN to build an end-to-end classifier, in a data-driven way was very interesting. Basically, it learns how to extract features from the raw images, so that they can be useful for the subsequent classification of the images. There is no need for preprocessing, meaning that the raw images can be used directly as inputs. This capability of the CNN allows the construction of soft biometric classifiers with a reduced complexity.

In the third study of this thesis, we also extended the CNN to deal with a stream of video frames. This provides a method to describe detected people in each frame. The results for this test case suggest that the CNN has a great generalization capability, since it was trained and tested with different datasets. In this work, three different attributes were approached: gender, upper clothes and lower clothes. This decision was based on the relatively large amount of labelled data required to train the CNN. However, it would be possible to classify other attributes if annotated data is available. Training additional classifiers to deal with other type of attributes, such as glasses, hat, or other clothes details, could allow a better description of a person in the images. Therefore, this refined information could be used to further improve the results of a content-based image/video retrieval system.

The results of this experiment were very promising, suggesting that further improvements can lead to a system capable of automatically annotating interesting human attributes in video streams, especially in surveillance videos. Based on these annotations, a content-based video search framework could be constructed, allowing a better usage of such data. Overall, the methods developed in this thesis, applied to real-world image/video datasets, leaded to results that strongly encourages further research in the area.

5.1 CONTRIBUTIONS

The research conducted during the construction of this thesis leaded, in our opinion, to the following contributions:

- Enhancement of computer vision and machine learning areas through the proposed methods and results;
- Proposition and development of an unsupervised feature learning method, capable of image separation into high level classes;
- Proposition and development of a method to describe images of people using soft biometrics;
- Proposition of the basis to the construction of a content-based image/video search system capable of dealing with high-level information, in the form of soft biometrics.

5.2 FUTURE WORK

As discussed before, there is still plenty of work to be done in the computer vision field. Therefore, we propose some possible extensions of this work.

Future work should focus on extending the proposed unsupervised framework to handle several classes simultaneously. In addition, other clustering methods could be tested, such as fuzzy *k*-means, Self Organizing Maps, Growing When Required Neural Networks, and others. Some effort should be done to eliminate a many control parameters needed as possible, thus increasing the level of independence of the proposed method.

Regarding the soft biometrics, we intend to improve the results using other image datasets, as well as to study the tuning of the CNN parameters. Since the proposed method uses the annotated raw data as input, it could be easily extended to other soft biometrics. Therefore, it is intended to obtain detailed information about people in images using other attributes.

Concerning the videos, it would be important to better explout the temporal information to improve the CNN performance. A clue to this could be the use of Hidden Mark Models or other method capable of dealing with temporal data. Incorporating the capacity of detecting and tracking people in video frames along with their description using soft biometrics is a desired objective. This could allow even more the simplification of a system to automatically annotate data of interest in videos, more specifically surveillance videos.

REFERENCES

AHONEN, T.; HADID, A.; PIETIKAINEN, M. Face description with local binary patterns: Application to face recognition. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 28, n. 12, p. 2037–2041, Dec 2006.

ARTHUR, D.; VASSILVITSKII, S. K-means++: The advantages of careful seeding. In: **Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms**. Philadelphia, PA: SIAM, 2007. (SODA '07), p. 1027–1035.

B, A.-Z. M.; RAWI, M. al. An efficient approach for computing silhouette coefficients. **Journal of Computer Science**, Science Publications, v. 4, n. 3, p. 252–255, 2008. ISSN 1549-3636.

BAI, L.; LAO, S.; GUO, J. Video semantic concept detection using ontology. In: **Proceedings of the Third International Conference on Internet Multimedia Computing and Service**. New York, NY, USA: ACM, 2011. (ICIMCS '11), p. 158–163.

BALDI, P.; SADOWSKI, P. The dropout learning algorithm. **Artificial Intelligence**, v. 210, p. 78–122, 2014.

BAY, H. et al. Speeded-up robust features (SURF). **Computer Vision and Image Understanding**, v. 110, n. 3, p. 346–359, jun. 2008. ISSN 1077-3142.

BENGIO, Y.; COURVILLE, A.; VINCENT, P. Representation learning: A review and new perspectives. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, IEEE Computer Society, Los Alamitos, CA, USA, v. 35, n. 8, p. 1798–1828, 2013. ISSN 0162-8828.

BIALKOWSKI, A. et al. A database for person re-identification in multi-camera surveillance networks. In: **Digital Image Computing : Techniques and Applications (DICTA 2012)**. Esplanade Hotel, Fremantle, WA: IEEE, 2012. p. 1–8. In this paper, we present a new challenging multi-camera surveillance database (available from https://wiki.qut.edu.au/display/saivt/SAIVT-SoftBio+Database). Disponível em: <http://eprints.qut.edu.au/53437/>.

BISHOP, C. M. **Pattern Recognition and Machine Learning (Information Science and Statistics)**. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

BLUMAN, A. **Elementary Statistics**. 7th ed.. ed. New York, USA: McGraw-Hill Higher Education, 2009.

BO, Y.; FOWLKES, C. C. Shape-based pedestrian parsing. In: **Proc. of IEEE Conference on Computer Vision and Pattern Recognition.** Washington, DC, USA: IEEE Computer Society, 2011. (CVPR '11), p. 2265–2272. ISBN 978-1-4577-0394-2.

BOURDEV, L.; MAJI, S.; MALIK, J. Describing people: Poselet-based attribute classification. In: **International Conference on Computer Vision**. [S.l.: s.n.], 2011.

BOURDEV, L.; MALIK, J. Poselets: Body part detectors trained using 3D human pose annotations. In: **Proc. of IEEE International Conference on Computer Vision**. Piscataway, NJ, USA: IEEE Press, 2009. p. 1365–1372.

BOUREAU, Y.-L.; PONCE, J.; LECUN, Y. A theoretical analysis of feature pooling in visual recognition. In: FüRNKRANZ, J.; JOACHIMS, T. (Ed.). **Proceedings of 27th International Conference on Machine Learning**. Madison, WI, USA: Omnipress, 2010. p. 111–118.

CAI, D.; ZHANG, C.; HE, X. Unsupervised feature selection for multi-cluster data. In: **Proceedings of 16th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'10)**. [S.l.: s.n.], 2010. p. 333–342.

CALONDER, M. et al. BRIEF: Computing a local binary descriptor very fast. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 34, n. 7, p. 1281–1298, 2012.

CHAPELLE, O.; SCHLKOPF, B.; ZIEN, A. **Semi-Supervised Learning**. 1st. ed. Cambridge,MA: The MIT Press, 2010.

CHEN, H.; GALLAGHER, A.; GIROD, B. Describing clothing by semantic attributes. In: **Proceedings of the 12th European Conference on Computer Vision - Volume Part III**. Berlin, Heidelberg: Springer-Verlag, 2012. (ECCV'12), p. 609–623.

CHEN, Z. Learning from nature: Natural computing meets virtual learning. In: **Proceedins of 2nd Internation Conference on Virtual Learning**. [S.l.: s.n.], 2007. p. 117–124.

CHERIYADAT, A. Unsupervised feature learning for aerial scene classification. **IEEE Transactions on Geoscience and Remote Sensing**, v. 52, n. 1, p. 439–451, 2014.

CHIDAMBARAM, C. et al. Multiple face recognition using local features and swarm intelligence. **IEICE TRANSACTIONS on Information and Systems**, The Institute of Electronics, Information and Communication Engineers, v. 97, n. 6, p. 1614–1623, 2014.

COATES, A.; KARPATHY, A.; NG, A. Y. Emergence of object-selective features in unsupervised feature learning. In: BARTLETT, P. L. et al. (Ed.). **NIPS**. [S.l.: s.n.], 2012. p. 2690–2698.

COATES, A.; LEE, H.; NG, A. An analysis of single-layer networks in unsupervised feature learning. In: GORDON, G.; DUNSON, D.; DUDIK, M. (Ed.). **Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics**. [S.l.: s.n.], 2011. v. 15, p. 215–223.

COLLOBERT, R.; KAVUKCUOGLU, K.; FARABET, C. Implementing neural networks efficiently. In: MONTAVON, G.; ORR, G.; MULLER, K.-R. (Ed.). **Neural Networks: Tricks of the Trade**. [S.l.]: Springer, 2012, (Lecture Notes in Computer Science, v. 7700). p. 537–557.

CORDER, G. W.; FOREMAN, D. I. **Nonparametric Statistics: a step-by-step approach**. New York, NY, USA: J. Wiley & Sons, 2014.

CORTES, C.; VAPNIK, V. Support-vector networks. **Machine Learning**, v. 20, n. 3, p. 273–297, 1995.

CUEVAS, E. et al. An improved computer vision method for white blood cells detection. **Computational and Mathematical Methods in Medicine**, v. 2013, n. art. no. 137392, 2013.

CUSHEN, G.; NIXON, M. S. Mobile visual clothing search. In: **IEEE International Conference on Multimedia and Expo Workshops.** Piscataway, NJ: IEEE Press, 2013. p. 1–6.

DALAL, N.; TRIGGS, B. Histograms of oriented gradients for human detection. In: **Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition**. Piscataway, NJ: IEEE Press, 2005. (CVPR '05), p. 886–893.

D'ANGELO, A.; DUGELAY, J.-L. Color based soft biometry for hooligans detection. In: **Proceedings of IEEE International Symposium on Circuits and Systems**. Piscataway, NJ: IEEE Press, 2010. p. 1691 –1694.

DAS, S.; SUGANTHAN, P. Differential evolution: A survey of the state-of-the-art. **IEEE Transactions on Evolutionary Computation**, v. 15, n. 1, p. 4–31, Feb 2011.

DELAC, K.; GRGIC, M. A survey of biometric recognition methods. In: **Proceedings of 46th International Symposium Electronics**. Zadar, Croatia: Croatian Society Electronics in Marine, 2004. p. 184–193.

DENG, J. et al. ImageNet: A Large-Scale Hierarchical Image Database. In: **CVPR09**. [S.l.: s.n.], 2009.

DONG, J. et al. A deformable mixture parsing model with parselets. In: **Proceedings of IEEE International Conference on Computer Vision**. Piscataway, NJ, USA: IEEE Press, 2013. p. 3408–3415.

DUDA, R. O.; HART, P. E.; STORK, D. G. **Pattern Classification**. 2. ed. New York: Wiley, 2001.

EROZEL, G.; CICEKLI, N. K.; CICEKLI, I. Natural language querying for video databases. **Information Sciences**, v. 178, n. 12, p. 2534 – 2552, 2008.

FAWCETT, T. An introduction to roc analysis. **Pattern Recogn. Lett.**, Elsevier Science Inc., New York, NY, USA, v. 27, n. 8, p. 861–874, jun. 2006. ISSN 0167-8655. Disponível em: <http://dx.doi.org/10.1016/j.patrec.2005.10.010>.

FUKUSHIMA, K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. **Biological Cybernetics**, v. 36, p. 193–202, 1980.

GAN, G.; CHENG, J. Pedestrian detection based on HOG-LBP feature. In: **Seventh International Conference on Computational Intelligence and Security.** [S.l.: s.n.], 2011. p. 1184–1187.

GONZALEZ, R. C.; WOODS, R. E. **Digital Image Processing (3rd Edition)**. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2006.

GRAY, D.; BRENNAN, S.; TAO, H. Evaluating appearance models for recognition, reacquisition, and tracking. In: **In IEEE International Workshop on Performance Evaluation for Tracking and Surveillance, Rio de Janeiro**. [S.l.: s.n.], 2007.

GRAY, D.; TAO, H. Viewpoint invariant pedestrian recognition with an ensemble of localized features. In: **Proc. of the 10th European Conference on Computer Vision: Part I**. Berlin, Heidelberg: Springer-Verlag, 2008. (Lecture Notes in Computer Science Volume., v. 5203), p. 262–275.

HALL, M. et al. The weka data mining software: An update. **SIGKDD Explorations Newsletter**, ACM, New York, NY, USA, v. 11, n. 1, p. 10–18, nov. 2009. ISSN 1931-0145.

HANSEN, D. M. et al. Automatic annotation of humans in surveillance video. In: **Proceedings of Fourth Canadian Conference on Computer and Robot Vision.** Piscataway, NJ: IEEE Press, 2007. p. 473–480.

HE, Y. et al. Unsupervised feature learning by deep sparse coding. In: **Proceedings of SIAM International Conference on Data Mining (SDM)**. [S.l.: s.n.], 2014. p. 902–910.

HINTON, G. E. et al. Improving neural networks by preventing co-adaptation of feature detectors. **CoRR**, abs/1207.0580, 2012.

HOFFMAN, K. L.; LOGOTHETIS, N. K. Cortical mechanisms of sensory learning and object recognition. **Philosophical Transactions of the Royal Society B: Biological Sciences**, v. 364, n. 1515, p. 321–329, fev. 2009.

HSU, C.-W.; CHANG, C.-C.; LIN, C.-J. **A Practical Guide to Support Vector Classification**. [S.l.], 2003.

HUBERT, L.; ARABIE, P. Comparing partitions. **Journal of Classification**, Springer-Verlag, v. 2, n. 1, p. 193–218, 1985. ISSN 0176-4268.

JAIN, A.; HONG, L.; PANKANTI, S. Biometric identification. **Communications of the ACM**, v. 43, p. 90–98, February 2000.

JARRETT, K. et al. What is the best multi-stage architecture for object recognition? In: **Proceedings of IEEE 12th International Conference on Computer Vision**. Piscataway, NJ: IEEE Press, 2009. p. 2146–2153.

JIA, Y. et al. CAFFE: Convolutional architecture for fast feature embedding. **CoRR**, abs/1408.5093, 2014. Disponível em: <http://arxiv.org/abs/1408.5093>.

KALANTIDIS, Y.; KENNEDY, L.; LI, L.-J. Getting the look: Clothing recognition and segmentation for automatic product suggestions in everyday photos. In: **Proceedings of the 3rd ACM Conference on International Conference on Multimedia Retrieval**. New York, NY, USA: ACM, 2013. p. 105–112.

KALEGARI, D. H.; LOPES, H. S. A differential evolution approach for protein structure optimisation using a 2D lattice model. **International Journal of Bio-Inspired Computing**, Inderscience Publishers, v. 2, n. 3/4, p. 242–250, 2010.

KIM, M.-G. et al. A survey and proposed framework on the soft biometrics technique for human identification in intelligent video surveillance system. **Journal of Biomedicine and Biotechnology**, vol. 2012, article id. 614146, 2012.

KOSKELA, M.; SJöBERG, M.; LAAKSONEN, J. Improving automatic video retrieval with semantic concept detection. In: **Proceedings of the 16th Scandinavian Conference on Image Analysis**. Berlin, Heidelberg: Springer-Verlag, 2009. (SCIA ’09), p. 480–489.

KRAUSE, J.; LOPES, H. A comparison of differential evolution algorithm with binary and continuous encoding for the mkp. In: **Proceedings of BRICS on Computational Intelligence and 11th Brazilian Congress on Computational Intelligence (BRICS-CCI CBIC)**. Piscataway, NJ: IEEE Press, 2013. p. 381–387.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. Imagenet classification with deep convolutional neural networks. In: BARTLETT, P. et al. (Ed.). **Advances in Neural Information Processing Systems 25**. Nevada, USA: NIPS, 2012. p. 1106–1114.

LECUN, Y. et al. Gradient-based learning applied to document recognition. **Proceedings of the IEEE**, v. 86, n. 11, p. 2278–2324, nov. 1998.

LECUN, Y.; KAVUKCUOGLU, K.; FARABET, C. Convolutional networks and applications in vision. In: **Proceedings of 2010 IEEE International Symposium on Circuits and Systems**. Piscataway, NJ: IEEE Press, 2010. p. 253–256.

LECUN, Y. et al. A theoretical framework for back-propagation. In: **The Connectionist Models Summer School**. [S.l.: s.n.], 1988. v. 1, p. 21–28.

LEE, C.; LEE, G. G. Information gain and divergence-based feature selection for machine learning-based text categorization. **Information Processing and Management**, v. 42, n. 1, p. 155–165, 2006.

LEE, H. **Unsupervised Feature Learning via Sparse Hierarchical Representations**. Tese (PhD Thesis) — Department of Computer Science, Stanford University, 2010.

LEE, H. et al. Unsupervised learning of hierarchical representations with convolutional deep belief networks. **Communications of ACM**, v. 54, n. 10, p. 95–103, out. 2011.

LEUTENEGGER, S.; CHLI, M.; SIEGWART, R. BRISK: Binary robust invariant scalable keypoints. In: **Proceedings of the IEEE International Conference on Computer Vision**. Piscataway, NJ: IEEE Press, 2011. p. 2548–2555.

LLOYD, S. Least squares quantization in PCM. **IEEE Transactions on Information Theory**, v. 28, n. 2, p. 129–137, 1982.

LOWE, D. G. Distinctive image features from scale-invariant keypoints. **International Journal of Computer Vision**, v. 60, n. 2, p. 91–110, 2004.

LU, B. et al. Semantic concept detection for video based on extreme learning machine. **Neurocomputing**, v. 102, p. 176 – 183, 2013.

MIKOLAJCZYK, K. et al. A comparison of affine region detectors. **International Journal of Computer Vision**, v. 65, n. 1-2, p. 43–72, nov. 2005.

MILNER, A. D.; GOODALE, M. A. **The visual brain in action**. [S.l.]: Oxford University Press, 1995.

MOHRI, M.; ROSTAMIZADEH, A.; TALWALKAR, A. **Foundations of Machine Learning**. Cambridge, MA: The MIT Press, 2012.

NERI, F.; TIRRONEN, V. Recent advances in differential evolution: a survey and experimental analysis. **Artificial Intelligence Review**, v. 33, n. 1-2, p. 61–106, 2010.

OJALA, T.; PIETIKäINEN, M.; MäENPää, T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 24, n. 7, p. 971–987, jul. 2002.

PATEL, B. V.; MESHRAM, B. B. Content based video retrieval systems. **CoRR**, abs/1205.1641, 2012.

PENATTI, O. A.; VALLE, E.; TORRES, R. da S. Comparative study of global color and texture descriptors for web image retrieval. **Journal of Visual Communication and Image Representation**, v. 23, n. 2, p. 359 – 380, 2012.

PERLIN, H. A.; LOPES, H. S. Extracting human attributes using a convolutional neural network approach. **Pattern Recognition Letters**, p. –, 2015. ISSN 0167-8655. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0167865515002159>.

PRICE, K.; STORN, R. M.; LAMPINEN, J. A. **Differential Evolution: A Practical Approach to Global Optimization (Natural Computing Series)**. Secaucus, NJ, USA: Springer-Verlag, 2005.

QUINLAN, J. R. **C4.5: Programs for Machine Learning**. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993. ISBN 1-55860-238-0.

REID, D. et al. Soft biometrics for surveillance: An overview. In: RAO, C.; GOVINDARAJU, V. (Ed.). **Handbook of Statistics Machine Learning: Theory and Applications**. [S.l.]: Elsevier, 2013, (Handbook of Statistics, v. 31). p. 327 – 352.

ROSENBERG, A.; HIRSCHBERG, J. V-measure: A conditional entropy-based external cluster evaluation measure. In: **Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning**. [S.l.: s.n.], 2007. p. 410–420.

SERMANET, P. et al. Overfeat: Integrated recognition, localization and detection using convolutional networks. **CoRR**, abs/1312.6229, 2013.

SHANDILYA, S. K.; SINGHAI, N. A survey on: Content based image retrieval systems. **International Journal of Computer Applications**, v. 4, n. 2, p. 22–26, July 2010.

SHARMA, G.; JURIE, F. Learning discriminative spatial representation for image classification. In: **Proc. of the British Machine Vision Conference**. [S.l.]: BMVA Press, 2011. p. 6.1–6.11.

SHARMA, G.; JURIE, F.; SCHMID, C. Expanded parts model for human attribute and action recognition in still images. In: **Proceedings of International Conference on Computer Vision and Pattern Recognition**. Piscataway, NJ: IEEE, 2013. p. 652–659.

SNOEK, C. G.; SMEULDERS, A. W. Visual-concept search solved? **Computer**, v. 43, n. 6, p. 76 –78, june 2010.

SRIVASTAVA, N. et al. Dropout: A simple way to prevent neural networks from overfitting. **Journal of Machine Learning Research**, v. 15, n. 1, p. 1929–1958, jan. 2014.

STORN, R.; PRICE, K. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. **Journal of Global Optimization**, v. 11, n. 4, p. 341–359, dez. 1997.

TAN, P.-N.; STEINBACH, M.; KUMAR, V. **Introduction to Data Mining, (First Edition)**. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2005.

TANG, J. et al. Video semantic analysis based on structure-sensitive anisotropic manifold ranking. **Signal Processing**, v. 89, n. 12, p. 2313–2323, 2009.

TUYTELAARS, T.; MIKOLAJCZYK, K. Local invariant feature detectors: A survey. **Foundation and Trends Computer Graphics and Vision**, v. 3, n. 3, p. 177–280, jul. 2008.

VENKATESH, Y. V. On the classificaton of multispectral satellite images using the multilayer perceptron. **Pattern Recognition**, v. 36, n. 9, p. 2161–2175, 2003.

VINH, N. X.; EPPS, J.; BAILEY, J. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. **Journal of Machine Learning Research**, JMLR.org, v. 11, p. 2837–2854, dez. 2010. ISSN 1532-4435.

WEBER, M.; BAUML, M.; STIEFELHAGEN, R. Part-based clothing segmentation for person retrieval. In: **Proceedings of IEEE International Conference on Advanced Video and Signal-Based Surveillance**. Piscataway, NJ: IEEE Press, 2011. p. 361–366.

XIE, Z. et al. Brief survey on image semantic analysis and understanding. In: **Proceedings of International Conference of Soft Computing and Pattern Recognition (SoCPaR)**. [S.l.: s.n.], 2011. p. 179–183.

YAMAGUCHI, K. et al. Parsing clothing in fashion photographs. In: **Proceedings of IEEE Conference on Computer Vision and Pattern Recognition**. Piscataway, NJ: IEEE Press, 2012. p. 3570–3577.

ZHA, Z.-J. et al. A comprehensive representation scheme for video semantic ontology and its applications in semantic concept detection. **Neurocomputing**, v. 95, p. 29–39, out. 2012.

ZHANG, C. et al. Semantic retrieval of events from indoor surveillance video databases. **Pattern Recognition Letters**, v. 30, n. 12, p. 1067–1076, set. 2009.

ZHANG, W. et al. Real-time clothes comparison based on multi-view vision. In: **Proceedings of Second ACM/IEEE International Conference on Distributed Smart Cameras.** Piscataway, NJ: IEEE Press, 2008. p. 1–10.