DISSERTATION

A KERNEL MATCHING APPROACH FOR EYE DETECTION IN SURVEILLANCE IMAGES

Diego Armando Benavides Vidal

Brasília, 23 November de 2016

UNIVERSIDADE DE BRASÍLIA

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASILIA
Faculdade de Tecnologia

DISSERTATION

A KERNEL MATCHING APPROACH FOR EYE DETECTION IN SURVEILLANCE IMAGES

Diego Armando Benavides Vidal

Report submitted to the Department of Mechanical Engineering as a partial requirement for obtaining the title of Master in Mechatronic Systems

Examination board

Díbio Leandro Borges, UnB
Advisor

José Maurício S. T. da Motta, UnB
Chair member

Hugo Vieira Neto, UTFPR
Chair member

Benavides Vidal, Diego Armando

A kernel matching approach for eye detection in surveillance images / Diego Armando Benavides Vidal. –Brasil, 2016.

68 p.

Orientador: Díbio Leandro Borges

Dissertação (Mestrado) – Universidade de Brasília – UnB
Faculdade de Tecnologia – FT
Programa de Pós-Graduação em Sistemas Mecatrônicos – PPMEC, 2016.

1. Machine learning. 2. Kernel methods. 3. Eye detection. 4. Computer vision. 5. Artificial intelligence. I. Díbio Leandro Borges, orientador. II. Universidade de Brasília. III. Faculdade de Tecnologia.

To

My life partner, Eloisa, for her support while I was following my passion, my madness. Thank you very much for being there.

Diego Armando Benavides Vidal

Acknowledgments

Here, I am using the opportunity to express my sincere gratitude to each the persons that gave me their support through of these years. In the first place I want to thank my wife and my son to being there and support me everyday I love you so much. Also, I would also like to thank my parents for the support they provided me through my entire life. Thank Pedro and Olga, my success will always be their successes. In Brasilia, I would like to thank to my "cousin" Max Vizcarra who believed in me for achieve this challenge and received me in his house when I arrived in Brasília. Also, I would like to thank to my new family that I met the first year of the program, especially to my Colombian brother Sebastian. In fact with all of you I had feel as home. I have to say that the main experience also have been meet all of you and their cultures that despite being so different from mine, we could found some similar things about our brother countries.

I would like to thank to PPMEC for the opportunity that they got me and I sincere hope that the program achieve all of its targets. In general I have to thank to UnB and CAPES for opening me the door to a new universe so different to mine, in fact, was an adventure to know each aspects about the student and research life in Brasília.

Finally, I want to thank to the professors that were guide me through of program road including to Professor José Maurício Motta for his advices to start the dissertation, to profesor Li Weigang to complement my knowledge in the IA world and especially I would like to thank my advisor Díbio Borges for trust and interest in my work since the first email that I sent him and he will definitely does it until my defence. His teachings has strengthened my skills as researcher and I hope that we can work together participating in more conference, publications and projects in the future. I must emphasize the human side because there are few persons that look beyond their own interests. He could keep me in calm when the situation seemed into chaos. This was great help to achieve the goal. Thank very much Professor.

Diego Armando Benavides Vidal

ABSTRACT

Eye detection is a open research problem to be solved efficiently by face detection and human surveillance systems. Features such as accuracy and computational cost are to be considered for a successful approach. We describe an integrated approach that takes the outputted ROI by a Viola and Jones detector, construct HOGs features on those and learn an special function to mapping these to a higher dimension space where the detection achieve a better accuracy. This mapping follows the efficient kernels match approach which was shown possible but had not been done for this problem before. Linear SVM is then used as classifier for eye detection using those mapped features. Extensive experiments are shown with different databases and the proposed method achieve higher accuracy with low added computational cost than Viola and Jones detector. The approach can also be extended to deal with other appearance models.

RESUMO

A detecção ocular é um problema aberto em pesquisa a ser resolvido eficientemente por detecção facial em sistemas de segurança. Características como precisão e custo computacional são considerados para uma abordagem de sucesso. Nós descrevemos uma abordagem integrada que segmenta os ROI emitidos por um detector Viola e Jones, constrói características HOGs e aprende uma função especial para mapear essas características para um espaço dimensional elevado onde a detecção alcança uma melhor precisão. Esse mapeamento segue a eficiente abordagem de funções Kernel, que se mostrou possível mas não foi feita para esse problema antes. Um classificador SVM linear é usado para detecção ocular através dessas características mapeadas. Experimentos extensivos são mostrados com diferentes bancos de dados e o método proposto alcança uma precisão elevada com baixo custo computacional adicional do que o detector Viola e Jones. O método também podem ser estendido para lidar com outros modelos equivalentes.

Table of Contents

List of Figures

List of Tables

List of Symbols

Symbols

R	Region in a image
R_i	The i-th rectangle in a image
P	Patch or block defined as a subregion of a image
tp	Normalized dimension of one size of a output generate for a detector
S	Dimension of one size of a block. For example, a $S \times S$ block
p	Size step using in the HOG feature structure
z	Pixel of a image
c	Amount of HOG features per block in a image
$\tilde{m}$	Magnitude of normalized gradient in a HOG framework
G_x	Matrix of x axis variations in a HOG framework
G_y	Matrix of y axis variations in a HOG framework
$\lfloor a \rfloor$	Major integer less than or equal to a used to characterized the descriptor vector of orientations per pixel z in a HOG framework
ϵ_p	Small constant to avoids zero division
P	Performance measure
K_p	Kernel descriptor
Pr	Probability measure
$\mathcal{K}$	Integral operator
Kc	Cluster number used in K-Means method
l	The cardinality of a samples set
$\mathcal{D}$	The dimension of the mapping $\Phi(x)$

Symbols Greek

λ	Parameter in Gaussian kernel
Δ	Variations between two values
α	KKT coefficient

Sets

$\mathbb{R}$	1-dimensional Euclidean space
$\mathbb{R}^n$	n-dimensional Euclidean space
$\mathcal{H}$	Hilbert space
E	Samples set
T	Tasks set
D	Training set
D_+	Training set samples labeled with y_+
D_-	Training set samples labeled with y_-
Y	Labels set
Z	Basis vector set

Funtions

$i(x, y)$	Origin image function
$ii(x, y)$	Integral image function
$m(z)$	Magnitude function in HOG framework
$\theta(z)$	Orientation function in HOG framework
$\delta(z)$	HOG indicator vector
$\delta_i(z)$	HOG i-th bin of the indicator vector
$\vartheta(P)$	HOG feature vector for the patch P
f	Decision function
$\hat{f}$	Decision function in Perceptron Model
g	Linear function
$\hat{g}$	Linear function in Perceptron Model
Φ	Map function
$k(x, y)$	Kernel function
$\mu(x)$	Binary feature vector

Acronyms

$2D$	Two dimensional
$3D$	Three dimensional
MTC	Modified Census Transform
SVM	Support Vector Machine
HOG	Histogram of oriented gradient
DPM's	Deformable part-based models
ERM	Empirical risk minimization
KKT	Karush Kuhn Tucker
BOW	Bag of words
EMK	Efficient match kernel
PCA	Principal component analysis
DP	Dual program
ROI	Region of interest
VJD	Viola and Jones detector

Uma abordagem de funções kernel para detecção de olhos em imagens de vigilância

Capítulo 1: Introdução

A detecção ocular vem sendo uma parte importante no desenvolvimento de sistemas de detecção pessoal e de ações do corpo humano em geral. É muito comum em áreas de aplicação assim como segurança por imagens e interação entre humano e computador, os quais frequentemente requerem alta precisão, exatidão e velocidade. Uma resposta rápida e precisa é a preocupação principal das pessoas em segurança visual, e a detecção ocular pode ser considerada a primeira e uma parte essencial de qualquer sistema de segurança facial e também para aplicações de rastreamento ou identificação [Yang, Kriegman e Ahuja 2002].

São muitos os desafios que devem ser citados quando lidamos com problemas de detecção ocular em sistemas de segurança de vídeo. Desafios como deformação, ponto de vista, estruturas variáveis e oclusão são os mais frequentemente achados no contexto de segurança por vídeo porque isso é feito num ambiente não controlado. Nesse sentido, sistemas de segurança atuais devem ser capazes de lidar com esses desafios sem um impacto significante em sua performance.

Muitos trabalhos [Awais, Badruddin e Drieberg 2013, Chen et al. 2014, Choi, Han e Kim 2011, Hyunjun, Jinsu e Jaihie 2014, Oliveira et al. 2012] que são feitos com a melhor tecnologia possível citaram os problemas com a detecção ocular, os quais obtiveram resultados em pesquisas conhecidas. Uma das abordagens mais influentes para a detecção ocular foi proposta por Viola e Jones no [Viola e Jones 2004]: Eles propuseram um método que incluía a ideia da imagem integral para computar características Haar-like em várias escalas diferentes para descrever uma imagem, e um algoritmo de apredizagem AdaBoost para seleção e classificação das características. Embora principalmente usado para detecção facial baseado em partes da face assim como nariz, boca e olhos, a abordagem de Viola e Jones é útil para a detecção de vários objetos como carros e pedestres por exemplo. Pesquisas tem se baseado no [Viola e Jones 2004] para gerar sistemas de detecção melhores em tempo real. Isso ocorre pela velocidade de resposta e o baixo custo computacional. De qualquer modo, as características Haar-like para detecção em ambientes não controlados geram um grande número de falsos positivos, portanto outros métodos devem ser usados para melhorar seu resultados.

Aqui nós propusemos um método para lidar com o problema de detecção ocular combinando os resulados do detector de Viola e Jones e melhorando sua precisão numa eficiente estrutura de funções Kernel. Um conjunto particular de características são obtidas pelos ROI (Regions of Interest) do detector de Viola e Jones e um algoritmo para aprender e classificar elas num modelo de funções Kernel é descrito e testado. Os eficientes descritores Kernel são propostos em [Bo e Sminchisescu 2009] como uma ferramenta fundamental para o reconhecimento visual. Aqui, é usado para reduzir o montante de falsos positivos do detector Viola e Jones sem comprometer os verdadeiros positivos. A principal contribucion de este trabalho é mostrar que esses descritores também podem ser usados para resolver o problema de dados não linearmente separáveis num espaço de entrada mais complexo. Assim, poderemos obter um detector ocular com melhor precisão sem um aumento considerável do custo computacional. Resultados experimentais obtidos usando os conjuntos de dados FDDB [Jain e Learned-Miller 2010], BioID [Jesorsky e Kirchberg Klaus e Frischholz 2001, CK+ [Lucey et al. 2010] e FERET [Phillips et al. 2000] mostram que nosso método alcança uma melhor precisão.

Capítulo 2: Revisão da literatura

Este capítulo familiariza o leitor com os trabalhos mais importantes citando os problemas da detecção ocular em imagens coletadas em ambientes espontâneos e como os métodos kernel são usados no reconhecimento visual. Existem trabalhos fora do contexto de segurança que precisamos mencionar, portanto dividimos esse capítulo em três partes importantes. A detecção ocular baseada na estrutura de Viola e Jones, a detecção ocular com diferentes métodos e detecção de objetos com os métodos kernel.

A seção 2.2 resume a estrutura Viola e Jones e os métodos baseados nela. A seção 2.3 apresenta os trabalhos mais importantes fora da estrutura Viola e Jones. Finalmente, a seção 2.4 apresenta um breve resumo das principais noções sobre os métodos Kernel que foram usados na detecção de objetos.

Capítulo 3: A metodologia proposta

Neste capítulo é apresentada a metodologia usada para desenvolver o detector de olhos proposto neste trabalho. Primeiramente definimos os conceitos de ambiente controlado e não controlado e suas principais diferenças são discutidas. Estas diferenças são detalhadas utilizando amostras dos bancos de dados usadas para os experimentos. Em seguida, apresentamos o detector de Viola e Jones como estágio de entrada (front end) para a detecção de objetos em especial para detecção de olhos e as suas principais vantagens e desvantagens presentes nos ambientes não controlados. Logo, uma solução para diminiur o número de falsos positivos dos resultados do detector [Viola e Jones 2004] baseada em descritores Kernel [Bo e Sminchisescu 2009] é proposta com o qual um detector com três estágios é proposto: O estágio de extração, que constrói um conjunto de vetores base (Visual Vocabulary) que definem bem as características de um olho. O estágio de treinamento, onde o conjunto de vectores base é usado para construir descritores Kernel os quais são usados para treinar um classificador SVM linear. Por fim, a fase de classificação, onde uma entrada não rotulada é classificada com o classificador linear gerado na fase de treinamento o qual define sem um ROI é um falso positivo ou não.

Capítulo 4: Experimentos

Neste capítulo nós descrevemos os resultados obtidos pelo detector proposto nesse trabalho. Primeiro, é detalhado os bancos de dados usados para os testes e os parâmetros para usar em cada caso. Nós apresentamos dois tipos de experimentos. O primeiro está relacionado com a quantidade de vetores base usado para definir as características de um olho. O objetivo deste experimento é validar se há melhorias nos resultados quando a quantidade de características de base aumenta ou diminui. O segundo experimento está relacionado com o parâmetro λ do Kernel Gaussiano usado para construir os descritores Kernel. É recomendado analisar os resultados do detector proposto quanto este parâmetro muda porque não existe um valor padrão para garantir que os resultados para um determinado problema serão os melhores. É informado anteriormente que os resultados de saída obtidos pelo detector Viola e Jones serão utilizados para comparações.

A seção 4.2 detalha todos os bancos de dados a se usar. Na seção 4.3 nós explicamos os resultados obtidos pelo detector proposto destacando dois tipos de testes em particular, quando a quantidade de vetores base muda e quando o valor do parâmetro λ muda.

Capítulo 5: Conclusões

Nós descrevemos um método de detecção ocular que trabalha com configurações sem restrições e melhora a detecção em vários bancos de dados de imagens. Uma aproximação de um mapeamento foi construída baseada em características HOG num subespaço de Hilbert provendo novas características conhecidas como descritores kernel eficientes [Bo e Sminchisescu 2009].

O detector de olho melhora os resultados do conhecido detector Viola e Jones usando os descritores kernel de características locais, com pelo menos 19% (FERET dvd1) mais precisão no pior caso, e 47% no melhor caso (CK+).

Esses descritores são projeções de vetores de características mapeados num espaço Hilbert de alta dimensão e são construídos baseados num conjunto finito de vetores base e operações com um kernel Gaussiano. Nós construímos esses descritores kernel para melhorar os resultados obtidos pelo detector Viola e Jones para olhos e obtivemos uma redução significativa em falsos positivos sem uma redução relevante dos verdadeiros positivos. Isso mostra que as características kernel levam em conta características de baixo nível que complementam a cada uma onde se detecta através da estrutura Viola e Jones sem custo adicional computacional durante todo processo de detecção. Como nós focamos apenas no resultado do detector Viola e Jones, as regiões de interesse foram reduzidas. Isso reduz o espectro de características falhas encontradas na projeção de um olho numa imagem.

Nesse capítulo, nós detalhamos as principais contribuições do detector proposto e as futuras linhas de investigação que devem ser exploradas quando a teoria kernel é utilizada nos problemas de detecção do objetos. Finalmente, nós concluímos esse capítulo mencionando detalhes do artigo gerado por esse trabalho e que foi aceito em uma conferência internacional.

Chapter 1

Introduction

1.1 Presentation

Eye detection has been a major part in the development of people detection systems and actions of the human body in general. It is very common in application areas such as security, video surveillance and human-computer interaction, which often require high precision, accuracy and speed. A fast and precise response is a major concern in visual surveillance of people, and eye detection can be considered a first and an essential part of any face surveillance system, either for tracking or for identification applications [Yang, Kriegman and Ahuja 2002].

Many are the challenges that must be addressed when we deal with eye detection problem in video surveillance applications. Challenges as deformation, viewpoint, variable structure and occlusion are most often found in context of video surveillance because in that is treated with uncontrolled environments. In that sense, current surveillance systems must be able to deal with them without significantly impacting its performance.

Many works [Awais, Badruddin and Drieberg 2013, Chen et al. 2014, Choi, Han and Kim 2011, Hyunjun, Jinsu and Jaihie 2014, Oliveira et al. 2012] inside state of the art have addressed the problem of eye detection which have resulted in well known investigations. One of the most influential approaches for eye detection was proposed by Viola and Jones in [Viola and Jones 2004]. They proposed a method including the idea of an integral image to compute Haar-like features in many different scales to describe an image, and an AdaBoost learning algorithm for features selection and classification. Although mainly used for face detection, and based on parts of faces such as nose, mouth and eyes, Viola and Jones approach is useful for many object detection tasks such as cars and pedestrians for example. Researchers have been based in Viola and Jones work to generate better detection systems. That is because its fast responses and low computational cost. However, Haar-like features for detections in uncontrolled environments generate a large number of false positives so other methods should be used.

Multi-vew approaches and features integration are also being used for generic 3D object recognition. In [Yu et al. 2014], a multi-view stochastic classification was presented where different features from multiple views were integrated in a probabilistic way describing high order distances between points. A multi-view manifold regularization learning and hypergraph matching was presented in [Hong et al. 2015] for 3D generic object recognition from 2D images with state of the art results.

Here, we propose a method to address the eye detection problem by combining the outputs of a Viola and Jones detector (VJD) and improving its precision in an efficient match kernel framework. A particular set of features are derived over the outputs of the Viola and Jones detector and an algorithm for learning and classifying those in a kernel matching paradigm is described and tested. Efficient kernel descriptors were proposed in [Bo and Sminchisescu 2009] as a general tool for visual recognition. Here, in order to reduce the amount of false positives of the Viola and Jones detector without significantly reduce the amount of true positives, we used the efficient kernel framework to construct a new class of features based on HOG (Histogram of oriented gradient) features and train a linear SVM (Support Vector Machine) using two class, eye or non-eye. The main contribution of our work is to show that these descriptors can also be used to solve the non linearly separable problem in a challenging input space. Thus, we can obtain an eye detector with better accuracy without significantly increase of the computational cost. Experimental results obtained using FDDB [Jain and Learned-Miller 2010], BioID [Jesorsky, Kirchberg and Frischholz 2001], CK+ [Lucey et al. 2010] and FERET datasets [Phillips et al. 2000] show that our method achieve the state of the art accuracy.

1.2 Objectives

The main objective of this work is to design, construct and test an efficient eye detector in environments not controlled. It is desired that this detector has fast responses and be able to confront the usual challenges of detection.

The specific objectives of this work are listed below:

1. Show that the kernel descriptors are a good complement to the Haar-like features.
2. Construct an algorithm to extract and train kernel features for eye detection.
3. Construct an application to test the proposed eye detector.
4. Test the proposed detector with known databases and generate indicators that show that the results obtained by Viola and Jones detector are improved significantly.
5. Validate that the kernel descriptors are an elegant way to address the challenges present in eye detection.

1.3 Structure of the document

The second chapter contain a review of the most important literature and the theoretical background necessary for the full understanding of the work developed. The third chapter presents the proposed methodology for the eye detector. In the fourth chapter, the most relevant experiments are described and, finally, the fifth chapter present our conclusions and lists some future lines of investigation.

Chapter 2

Review of the literature

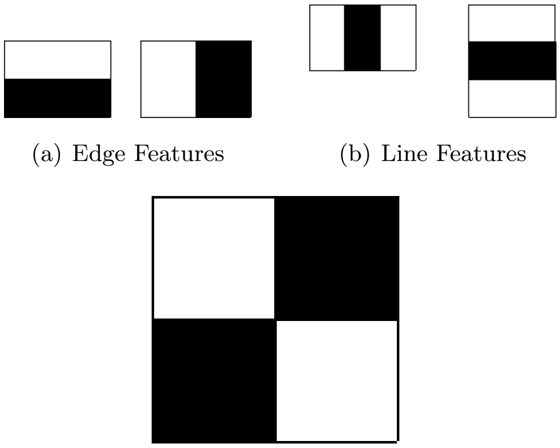
2.1 Introduction

This chapter familiarizes the reader with the most important works addressing the eye detection problem on images in unconstrained environments and how the kernel methods are used in visual recognition. There are important works outside of the surveillance context that we need to mention, so we divided this chapter in three main parts. Eye detection based in the Viola and Jones framework, eye detection with different methods and object detection using kernel methods.

Section 2.2 summarizes the Viola and Jones framework and the methods based on it. Section 2.3 present the most important works whose do not use the Viola and Jones framework. Finally, Section 2.4 present a brief review of the main notions about kernel methods that have been used in object detection.

2.2 Eye detection based in Viola and Jones framework

There are many challenges that we must consider when we dealing with the eye detection problem in visual recognition. Problems as deformation, viewpoint, variable structural or occlusion are currently the center of the interest for many researchers. However, the precision and speed ratio of the detection are always taken into account for real time applications as in the case of eye detection in surveillance context. There are many research addressing both precision and speed problem but one of the most influential approaches for eye detection was proposed by Viola and Jones in [Viola and Jones 2004]. They proposed a method including Haar-like features in many different scales to describe an image patch or ROI (Region of interest). Haar-like features are composed of rectangular regions such as the ones shown in Figure 2.1 and their value is the sum of pixels within the clear rectangles subtracted by the sum of pixels within the shaded rectangles. Rectangular features can be calculated in a fast way using a representation of image patch that they called integral image. This is basically an image where each pixel (x, y) equals the sum of every pixel above and to the left of (x, y) in the original image. That is



(a) Edge Features

(b) Line Features

(c) Four-rectangle Features

Figure 2.1: The Haar-like features used in Viola and Jones framework and the same features are used in many current investigations.

$$ii(x,y) = \sum_{x' \leq x, y' \leq y} i(x^{'}, y^{'}), \tag{2.1}$$

where $ii(x,y)$ is the integral image and $i(x,y)$ is the original image.

Figure 2.2 shows how the integral image works to calculate the rectangle pixel sum. If we need the pixels sum of the rectangle R_4 only the operation $ii(x_4,y_4)+ii(x_1,y_1)-(ii(x_2,y_2)+ii(x_3,y_3))$ is required.

The integral image is a strong contribution but a second contribution was made. A classifier by the selection of critical visual features through an AdaBoost learning algorithm [Freund and Schapire 1997] was constructed to improve de classification rate. More about this cascade architecture is explained in the Chapter 3 for our proposed method.

Many researchers have been follow the work of Viola and Jones [Awais, Badruddin and Drieberg 2013, Chen et al. 2014, Choi, Han and Kim 2011, Hyunjun, Jinsu and Jaihie 2014, Oliveira et al. 2012]. For instance, a boosting classification using Haar-like features was used to detect faces and eyes in [Chen et al. 2014] for real-time eye detection and identifying events. Similar strategies to reduce the computational cost of the detection were used in [Awais, Badruddin and Drieberg 2013] for detecting both eyes. Based on the correct face detection, a golden ratio was used to detect the eye pair in that work. By first detecting one eye and then applying symmetry of the face, the other eye could be detected. Then this detection was used to track the eye blinking. Also, in [Choi, Han and Kim 2011], AdaBoost approach was used with Modified Census Transform (MCT) [Froba and Ernst 2004] to construct a linear combination of weak classifiers with MCT-based eye features for eyes blinking detector.

In [Oliveira et al. 2012] instead of using MCT-based eye features, weak statistical features were proposed to build a heuristic algorithm taking into account length and variance to especially

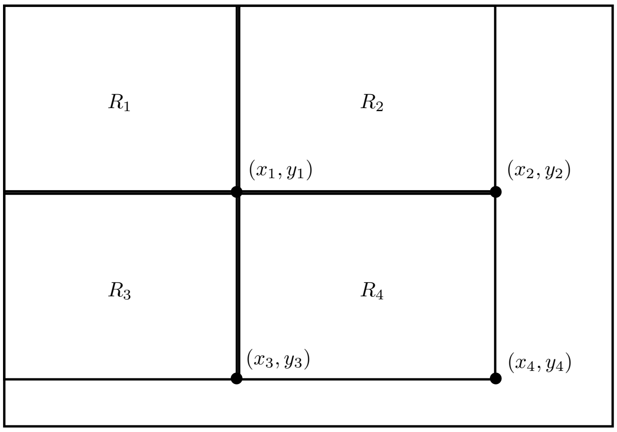


Figure 2.2: Calculation of rectangle sum using four array reference. (x_1, y_1), (x_2, y_2), (x_3, y_3) and (x_4, y_4) are the reference in the original image and R_1, R_2, R_3 and R_4 are regions in the original image.

locate features over regions of interest. Features as are shown in Figure 2.3 were found in the resulting outputs of Viola and Jones detector and with these was built an algorithm used in a cascade classification to decrease the number of false positives.

All the works detailed in this section have a good speed ratio as their better characteristic but the accuracy for eye detection in complex environments could still be improved. In Section 2.3 we will review another types of methods that focusing in the precision instead of speed.

2.3 Eye detection with different methods

The major contribution of the methods referred in Section 2.2 is their ability to perform the detection with an acceptable speed for real-time applications. However, their major disadvantage is that they do not have a good precision in uncontrolled environments and sometimes in controlled environments too. Taking into account that for a good detection in surveillance context accuracy is an important feature, it is necessary make a review to the methods focusing in optimize that feature in the state of the art even leaving aside the detection speed.

There are many works [Pandey and Lazebnik 2011, Dalal and Triggs 2005, Felzenszwalb et al. 2010, Kumar, Raja and Ramakrishnan 2002, Pedersoli and Leuven 2014] have dealt with the accuracy problem for objects detection with different methodologies and usually these start defining a form to characterize the object of interest (car, human, face or eye) through a set of local features. It is not always easy to extract local features of some object of interest in an image but there are some popular approaches that have been used in current researches that are important to mention in this work. For example, in [Dalal and Triggs 2005] were used local features and a linear SVM for human detect. The major contribution of that work was extract local features based on evaluating normalized local histograms of image gradient orientations in a dense grid called Histogram of Oriented Gradient or HOG. To understand that approach is necessary to remember that the image of an object, for example of a human, is the $2D$ expression of a moment in $3D$ world, so it is

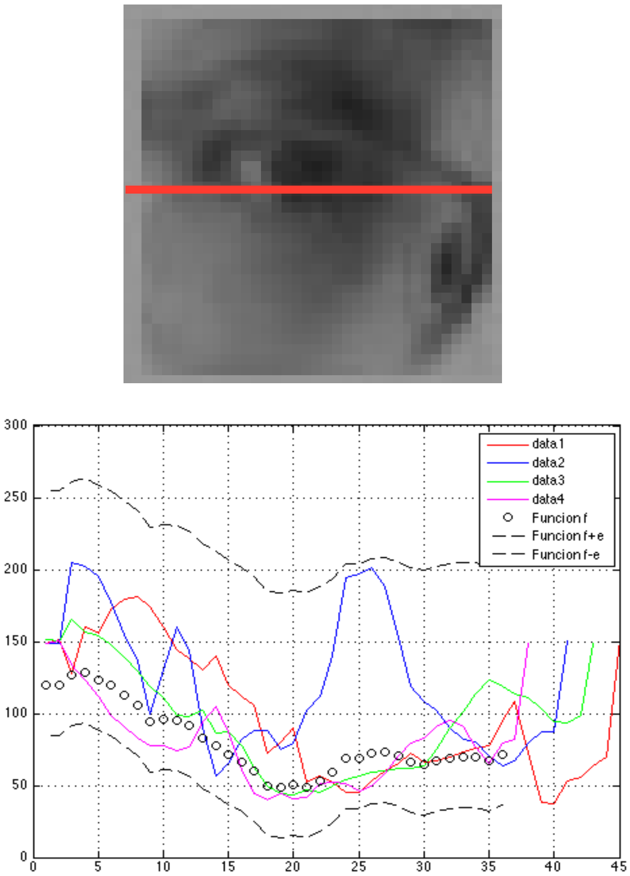


Figure 2.3: Intensity pixel pattern in eye image on middle row of pixels and the graph of intensity of many eye samples.

possible find a way to extract human features in an image as shown in Figure 2.4. The human shoulder in Figure 2.4 has some patterns in the image which can be expressed with variations of the pixels intensity. This variations can be used to construct a features vector or descriptor vector to train a classifier.

Formally, Histogram of Oriented Gradients or HOG is a model for extraction of low level features. Basically, it consists in build a histogram of magnitudes $m(z)$ and orientations $\theta(z)$ of variation vectors of each pixel z of image. This construction is done within of a set-patch of sub-regions of the image or ROI (Region of interest). Given that in image processing an image is considered a scalar matrix (e.g, image in grayscale) where each scalar represent a pixel, we can calculate the variation of each pixel in two directions using the rows and columns of the matrix. For example, given z pixel inside 3×3 image as shown in Figure 2.5. The resulting variation vector consist in the variations in the axis x and y of the neighbors pixels close to z, that is, in axis x there would be a variation of $\Delta x = 94 - 54 = 40$ and in axis y, $\Delta y = 101 - 45 = 56$. So, the variation vector for the pixel z is $(\Delta x, \Delta y) = (40, 56)$. Repeating this process for all the pixels at the image patch, two matrix G_x and G_y are obtained. Those contains the variations in axes x and y respectively.

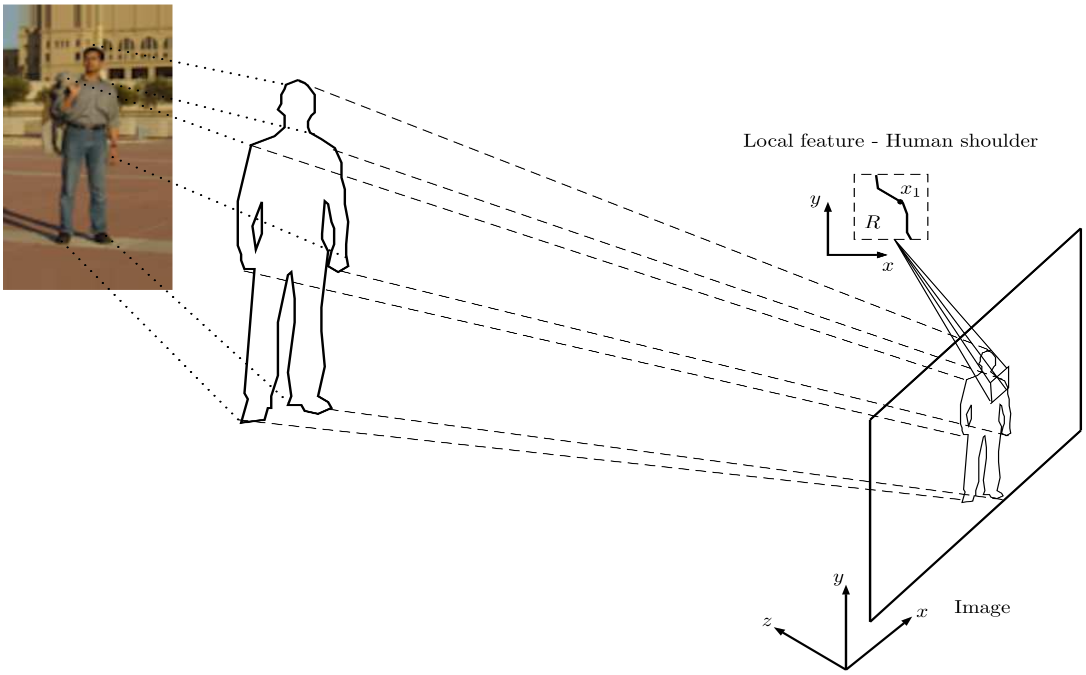
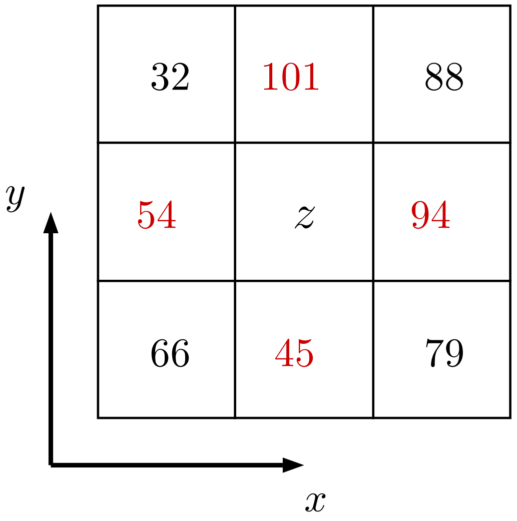


Figure 2.4: Representation of a human in the image. Here R is the area of a local features.



32	101	88
54	z	94
66	45	79

Figure 2.5: Local variation of pixel at 3 × 3 image on the construct of HOG descriptors.

After calculating the matrix G_x and G_y is straightforward to get magnitude $m(z)$ and orientation $\theta(z)$ of the resulting gradient vector for each pixel z in an image using the equations

$$m(z) = \sqrt{G_x^2(z) + G_y^2(z)}, \tag{2.2}$$

$$\theta(z) = \arctan\left(\frac{G_y(z)}{G_x(z)}\right), \tag{2.3}$$

where $G_x(z)$ y $G_y(z)$ represent the variations of pixel z in axis x and y respectively.

Now to finish the construction of the HOG descriptor is necessary characterize somehow the orientations and the magnitudes for all pixels at image. In HOG (also in SIFT [Lowe 2004]) the intensity variation expressed as orientation of each pixel is discretized into a d-dimensional indicator vector $\delta(z) = [\delta_1(z), \cdots, \delta_d(z)]$ with

$$\delta_i(z) = \begin{cases} 1, & \lfloor \frac{d\theta(z)}{\pi} \rfloor = i - 1 \\ 0, & \text{otherwise.} \end{cases} \tag{2.4}$$

where $i = 1, \cdots, d$ and $\lfloor a \rfloor$ takes the major integer less than or equal to a. Therefore, the resulting vector descriptor of orientations for a pixel z, taking into account d bins, is $\delta(z) = [\delta_1(z), \cdots, \delta_d(z)]$. Now, as we already have a categorization of orientations and we can calculate the magnitudes $m(z)$ for each pixel z, it is possible build the features vector of the form $\vartheta_p(z) = m(z)\delta(z)$. Adding this idea in a image patch P (or ROI) we obtain the histogram of oriented gradients (HOG)

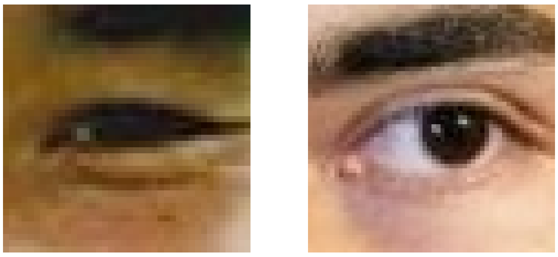
$$\vartheta(P) = \sum_{z \in P} \tilde{m}(z)\delta(z) \tag{2.5}$$

where $\tilde{m}(z) = m(z)/\sqrt{\sum_{z \in P} m(z)^2 + \epsilon_p}$ is the magnitude of normalized gradient, with ϵ_p a small constant that avoids zero division.

The functional relationship (2.4) that induces the bins is a strong expression of the underlying classification that was taken for the simple presentation of the problem. For practical purposes, it is common to use a soft expression as is used in [Bo and Sminchisescu 2009].

Many works have been inspired by [Dalal and Triggs 2005] but the most important work for object detection that deals with the problem of accuracy was developed in [Felzenszwalb et al. 2010]. Even though it was not necessarily developed for eye detection but for objects as cars or persons, that approach can be adapted for objects detection in general. In that work the authors focused in many challenges that appear in the problem of object detection. For instance, many variations arise in illumination and viewpoint as in the case of eye detection in unconstrained environments, but they are not necessarily unique variations. There are non-rigid deformations, and intraclass variability in shape and other visual properties too. A sample of this can be seen when we look within different classes of races that exists between people in the world. For example, Asian people have a different eye shape than people of south America as shown Figure 2.6. To address these challenges in [Felzenszwalb et al. 2010] was proposed an object detection system that represents highly variable objects using mixtures of multi-scale deformable part models. This system is based on the framework extension of the pictorial structures [Fischler and Elschlager 1973], [Felzenszwalb and Huttenlocher 2005] using visual grammars [Felzenszwalb and McAllester 2007], [Jin and Geman 2006], [Zhu and Mumfordt 2007] to represent objects with variable hierarchical structures called deformable part-based models or DPM's. In addition, and because of the nature of grammar based models, it is necessary use a latent variable formulation called latent SVM [Yu and Joachims 2009]. It is because that object representation is composed of unlabeled parts (sub-regions of the image) of the object of interest such that this information must be considered as latent information.

The detection system developed in [Felzenszwalb et al. 2010] can be used for detection tasks and obtain a good precision in its results but with high computation cost for some types of objects and using images with unconstrained environment. This is because there is an underlying match



(a) Asian people.

(b) South American people.

Figure 2.6: Racial differences. Shape and color.

based in two types of filter in multiple scales. Based in the HOG features used in [Dalal and Triggs 2005] a response through a root filter plus part filters is calculated to make a match. Using that match approach is possible obtain the localization of root box that contain the object of interest and its parts. Later, these box and part boxes are used to construct a total feature to training a latent SVM.

The same approach was used in [Pandey and Lazebnik 2011] to scene recognition as shown Figure 2.7. In that work the authors demonstrated that DPM's are capable to recognize disordered structures taking into account latent variable for the training.

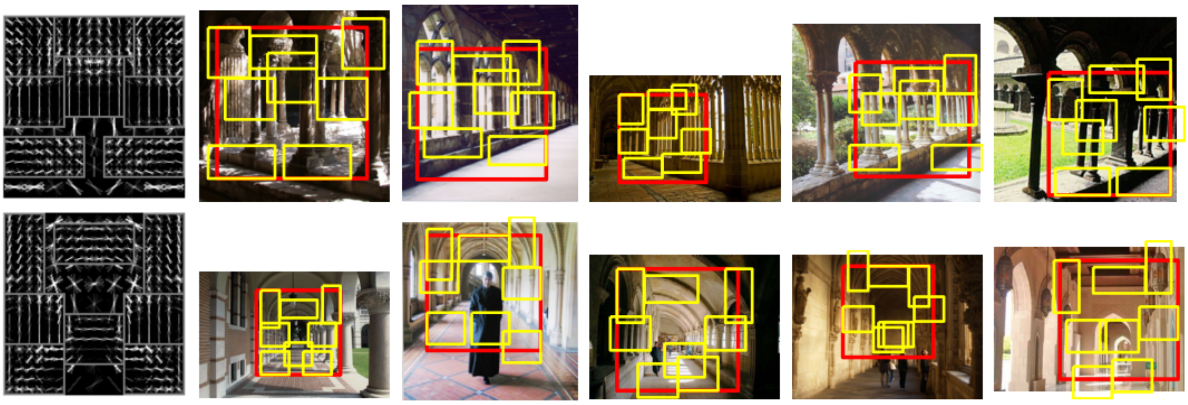


Figure 2.7: A sample of scene recognition made in [Pandey and Lazebnik 2011]. Left column: root and part filters based HOG features. Right five columns: root and parts box position of the scene.

2.4 Kernel methods for object recognition

In this section it is explained the main notions about kernel methods that will help the reader to understand many definitions and theories related with them in an object recognition context. As the kernel methods are embedded within the concept of machine learning, it is necessary to start defining that and how it is used in object recognition. Later, it is explained how machine learning are related with the kernel methods, what is the motivation to use them and the main methods known. Finally, we present a brief review about how kernel methods are used in object recognition.

2.4.1 Machine learning

Learning is a word difficult to define and it is possible to find a different definition about that in many research areas as Psychology, Neuroscience, Medicine, Computer Science or Mathematics. In this section we do not have as a target finding a correct definition about learning but explain existing models about machine learning that are known inside the area of artificial intelligence. In this sense, it is presented below a machine learning definition made by Tom Mitchell in his book [Mitchell 1997] that will be used along this work.

Definition 2.4.1 (Machine learning) *A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.*

To understand this definition we can use the context of eye detection and setting T as recognizing and classifying eye images, P as percent of eyes correctly classified and E as database of eyes with given classifications. Throughout this work is considered T as a tasks set, E as a samples set and P as a performance measure.

A model about machine learning made according Definition 2.4.1 was the Perceptron model by Frank Rosenblantt in [Rosenblatt 1958]. It was presented as a hypothetical nervous system designed to mimic some of the organized systems of the brain for human learning. Many researches about object recognition has been inspired in Rosenblatt's work so it is good explain main keys about it.

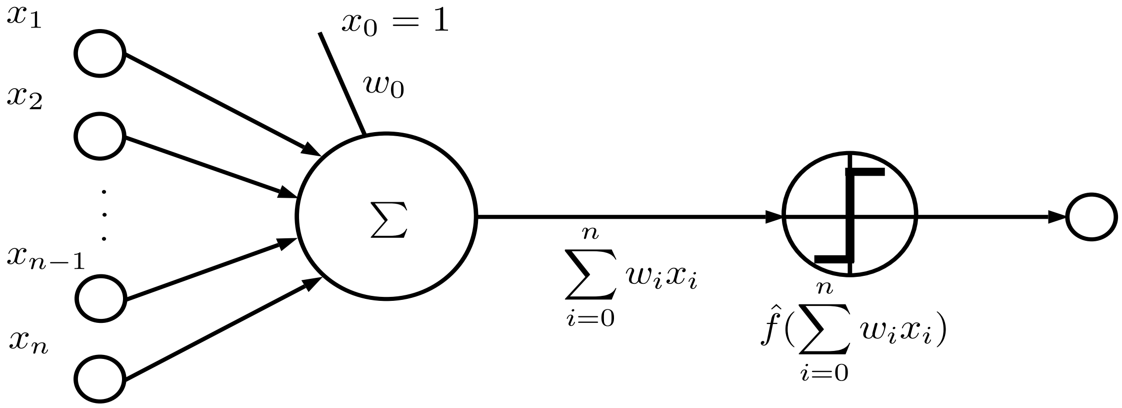


Figure 2.8: Perceptron model present by Frank Rosenblatt in 1958. Here x_1, x_2, $\cdots$, x_n are components of a vector that represent a sample of data in E. This vector is used to estimate weight parameters w_i in a linear function. Then, this function is used to calculate an output response $\hat{f}(\cdot)$ usually called decision function.

So according to Definition 2.4.1 suppose we have a set of samples $E \subset \mathbb{R}^n$ where $\vec{x}_m = (x_{m1}, \cdots, x_{mn})$ is a sample, and a set of labels $Y = \{y_-, y_+\} \subset \mathbb{R}$ that we can associate with each sample in E such that it is defined a training set D with the form

$$D = \{(\vec{x}_m, y_m) : \vec{x}_m \in E \wedge y_m \in Y\}. \tag{2.6}$$

Rosenblantt proposed a learning model where is build a linear function $\hat{g}$ such that

$$\hat{g}(\vec{x}_m) = \sum_{i=0}^{n} w_i x_i, \tag{2.7}$$

where $\vec{x}_m \in E$, $x_0 = 1$ and $\{w_0, \cdots w_1\} \subset \mathbb{R}$ is a weight set which should be calculated through an iterative algorithm based on E in a process called training. This is made by a performance measure P such that

$$P(\vec{x}_m) = ||\hat{f}(\sum_{i=0}^{n} w_i x_i) - y_m||, \tag{2.8}$$

where y_m is the label associated with the sample $\vec{x}_m$ and $\hat{f} : \mathbb{R}^n \to Y \subset \mathbb{R}$ is a function that return the estimated label for sample $\vec{x}_m$ called decision function. Here, P allow us to estimate the correct amounts of each element in $\{w_i\}$ using E for a given T. Figure 2.8 shown this basic learning model.

The approach made by Frank Rosenblatt inspired models such as classical neural networks and more elegant models as SVM. In this work we are interested in explain more about SVM because it is used as learning method for the proposed eye detection system. But first, it is necessary translate the idea of the perceptron approach as a binary classification problem.

The binary classification problem is defined as the task of distributing the elements of a not empty training set D such that $D = D_+ \cup D_-$ and $D_+ \cap D_- = \emptyset$ where

$$\begin{aligned} D_+ &= \{(\vec{x}_m, y_m) : y_m = y_+\} \text{ and} \\ D_- &= \{(\vec{x}_m, y_m) : y_m = y_-\}. \end{aligned} \tag{2.9}$$

In practice is possible to perform this type of classification through a function $g : \mathbb{R}^n \to \mathbb{R}$ but that is not always easy. A basic sample can be shown if a training set D is linearly separable.

Definition 2.4.2 (Linearly separable set) *A training set D is linearly separable if there is a linear function $g : \mathbb{R}^n \to \mathbb{R}$ such as*

$$\begin{aligned} g(\vec{x}) - b > 0, &\quad \textit{if } (x, y) \in D_+, \\ g(\vec{x}) - b < 0, &\quad \textit{if } (x, y) \in D_-. \end{aligned} \tag{2.10}$$

where $b \in \mathbb{R}$.

It is possible represent a Definition 2.4.2 in a geometrically example inside eye detection context as shown Figure 2.9. Here, we used the resulting output of Viola and Jones detector as a training set D and g classifies its elements in D_+ and D_-.

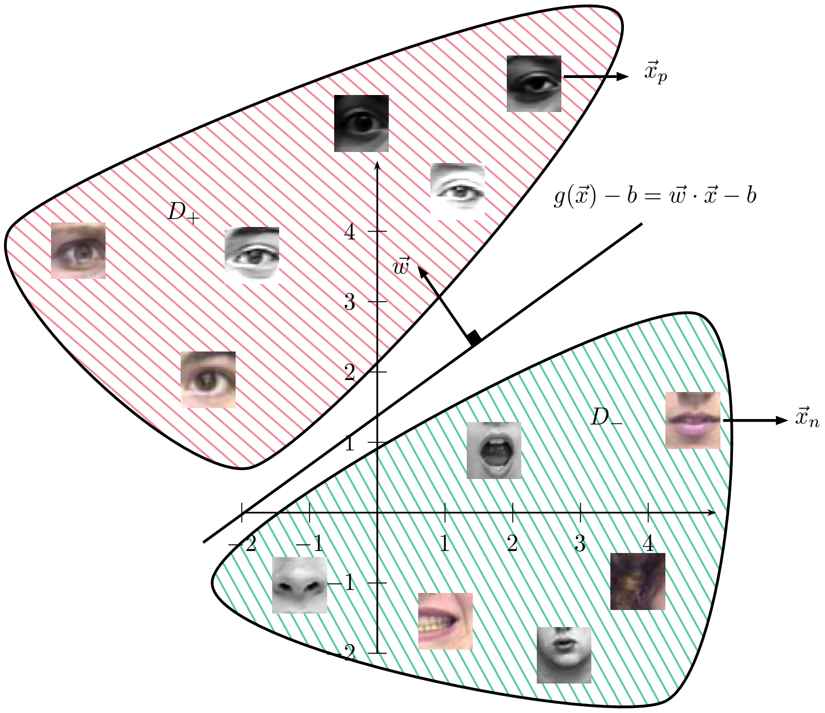


Figure 2.9: Linearly separable training set where $\vec{x}_n$ and $\vec{x}_p$ are false positive and true positive sample respectively resulting of the Viola and Jones detector. A linear function $g(\vec{x})-b = \vec{w}\cdot\vec{x}-b = 0$ classifies these samples eye regions inside D_+ and D_-.

It is not difficult to notice that linear function $\hat{g}$, in perceptron model, is a good alternative for function $g - b$ (with $w_0 = -b$) but the problem is how to construct function g and find a suitable b to solve a binary classification problem in an optimal way. A better approach to find both g and b was present in 1992 by Vladimir Vapnik and his team of AT&T Labs in [Vapnik, Boser and Guyon 1992] called Support Vector Machine or simply SVM. Based in the empirical risk minimization (ERM) [Vapnik 1992] and formalized in probability theory [Vapnik and Chervonenkis 1971], Vapnik concluded that the problem of finding a linear function $g(\vec{x}) = \vec{w} \cdot \vec{x}$ and b to solve the binary classification problem taking a linearly separable training set D can be solved through the solution of a convex optimization problem (DP) [Hamel 2009] of the form

$$(DP)\begin{cases} \arg\min\limits_{\vec{\alpha}} h(\vec{\alpha}) & = \arg\max\limits_{\vec{\alpha}}(\sum\limits_{i=1}^{l} \alpha_i - \dfrac{1}{2}\sum\limits_{i=1}^{l}\sum\limits_{j=1}^{l} \alpha_i\alpha_j y_i y_j \vec{x_i}\cdot\vec{x_j}) \\ \text{subject to} & \sum\limits_{i=1}^{l} \alpha_i y_i = 0, \\ & \alpha_i \geq 0 \\ \text{for} & i = 1, ..., l. \end{cases} \tag{2.11}$$

where l is the cardinality of the set E, $\vec{w}^* = \sum_{i=1}^{l} \alpha_i^* y_i \vec{x}_i$ and $b^* = \sum_{i=1}^{l} \alpha^* y_i \vec{x}_i \cdot \vec{x}_{sv} - 1$ are the optimum values for $g - b$, and $\vec{x}_{sv}$ is an element in E called support vector calculated using the KKT conditions [Peressini, Sullivan and Uhl 1988] related with the problem (DP). Geometrically find the optimum values w^* and b^* for $g(\vec{x}) - b$ can be understood like finding a linear function that separates the sets D_+ and $D-$ better. This separation is associated with a concept called maximum margin associated in turn with two parallel functions as shown Figure 2.10. This means that the greater the margin or the greater the distance separating the functions $g(\vec{x}) - b + 1 = 0$ and $g(\vec{x}) - b - 1 = 0$, classification is better. Finally, optimal linear function $\vec{w}^* \cdot \vec{x} - b^* = 0$ can be used to classify an element $\vec{x}_s \notin E$ with a decision function (like $\hat{f}$ in Perceptron model) of the form

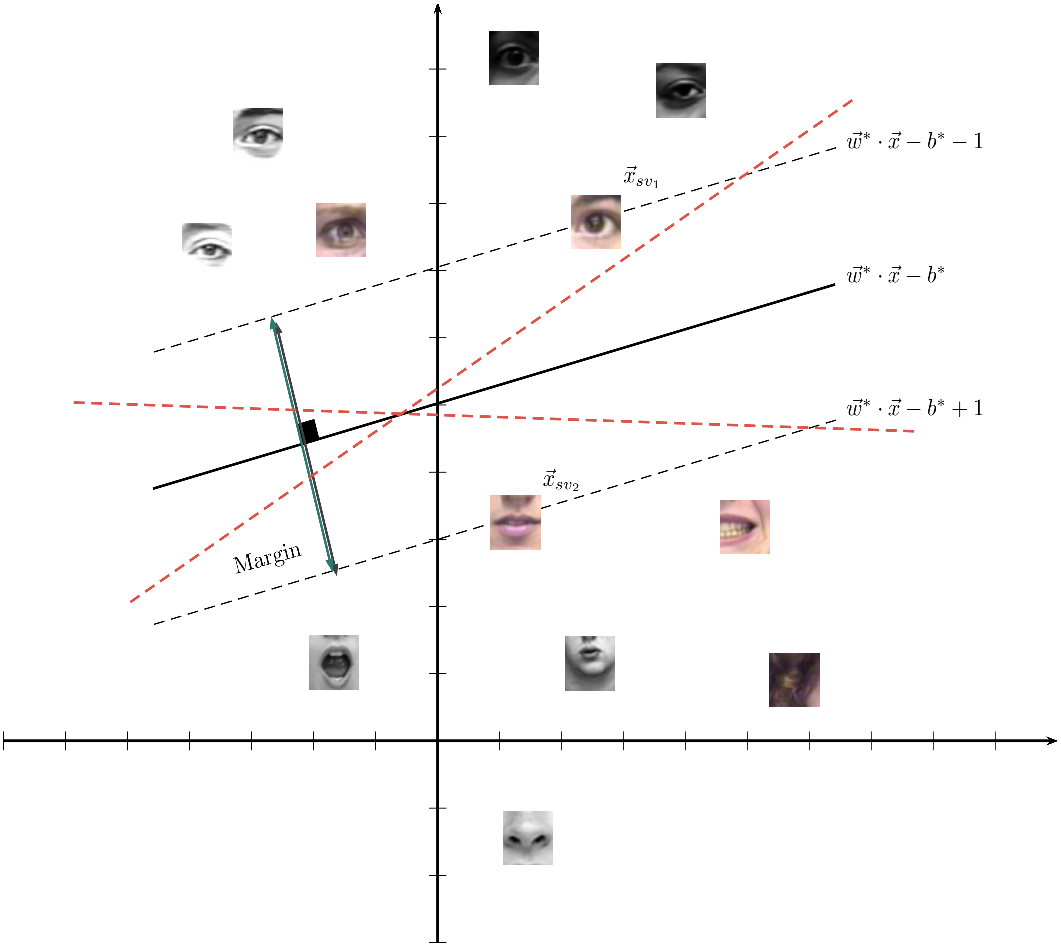

$\vec{w}^* \cdot \vec{x} - b^* - 1$
$\vec{x}_{sv_1}$
$\vec{w}^* \cdot \vec{x} - b^*$
$\vec{w}^* \cdot \vec{x} - b^* + 1$
$\vec{x}_{sv_2}$
Margin

Figure 2.10: A geometric sample of SVM approach. Solving problem (DP) and based in an support vector, in this case $\vec{x}_{sv_1}$ or $\vec{x}_{sv_2}$, the optimum values $\vec{w}^*$ and b^* for $g(\vec{x}) - b = \vec{w} \cdot \vec{x} - b = 0$ are calculated. The red dotted lines are not optimal linear functions.

$$f(\vec{x}_s) = \begin{cases} y_+, & \text{if } \vec{w}^* \cdot \vec{x}_s - b^* > 0, \\ y_-, & \text{if } \vec{w}^* \cdot \vec{x}_s - b^* < 0, \end{cases} \tag{2.12}$$

where $y_+, y_- \in Y$ and $\vec{x}_s \in \mathbb{R}^n$. With decision function f is possible solve the binary classification problem based in a linear separable set D better than in Perceptron model. However, there still is a challenge related with D to solve in a real-life problems.

In practice there are few real-life problems that can be express as a binary classification problem or decision problem related with a training set D linearly separable. For this reason the SVM version presented above can not be used for most real world problems such as eye detection. However, there is a way to deal with non-linearly separable training set using an approach called Kernel Trick.

In 1964 information theorist Thomas M. Cover in his research about pattern recognition [Cover 1965] proposed one way to map the elements in a non-linearly separable set D where E is in some input space (usually in Euclidean space $\mathbb{R}^n$) to an space $\mathcal{H}$ with high dimension, possibly infinite, where the mapped element are possibly linearly separable. Formally, $\mathcal{H}$ is an Hilbert space called feature space but the notion about this concept will be more important later. To understand well the linearly separability of a training set D in this new space it is necessary define the concept of ϕ-separable.

Definition 2.4.3 (ϕ-separable) *Let $E \subset \mathbb{R}^{m_0}$ and $\mathcal{H}$ is an m_1-dimensional space. A mapped training set $\bar{D} \subset \mathcal{H}$ is ϕ-separable if there is an m_1-dimensional vector $\vec{w} \in \mathcal{H}$ such that*

$$\begin{aligned} \vec{w} \cdot \phi(x) - b > 0, &\ \ \textit{if } (\phi(x), y) \in \bar{D}_+, \\ \vec{w} \cdot \phi(x) - b < 0, &\ \ \textit{if } (\phi(x), y) \in \bar{D}_-, \end{aligned} \tag{2.13}$$

where $\bar{D} = \{(\phi(\vec{x}), y) : \vec{x} \in E \wedge y \in Y\}$, $\phi : E \subset \mathbb{R}^{m_0} \to \mathbb{H}$ is a map that take the elements of E to $\mathcal{H}$ when $m_1 > m_0$.

Here, $\bar{D}_+ = \{(\phi(x), y) : y = y_+\}$ and $\bar{D}_- = \{(\phi(x), y) : y = y_-\}$ satisfy $\bar{D} = \bar{D}_+ \cup \bar{D}_-$ and $\bar{D}_+ \cap \bar{D}_- = \emptyset$ according to Definition 2.4.2.

Notice that if $\bar{D}$ is linearly separable in $\mathcal{H}$ it is possible to use SVM approach to classify its elements in that space. But, how to guarantee that bringing the elements of $E \in \mathbb{R}^{m_0}$ to $\mathcal{H}$, its mapped elements will be linearly separable. The response to this question is in Cover theorem.

Theorem 2.4.1 (Cover theorem) *If $Pr(N, m_1)$ is the probability of separation in D randomly selected be ϕ-separable, then*

$$Pr(N, m_1) = (\frac{1}{2})^{N-1} \sum_{k=0}^{m_1-1} \binom{N-1}{k}, \tag{2.14}$$

where N is the cardinality of $E \subset \mathbb{R}^{m_0}$ and m_1 is the dimension of $\mathcal{H}$.

According to Theorem 2.4.1 is possible guarantee that when the dimension of $\mathcal{H}$ increases, the probability of linear separability of $\bar{D}$ based on E and some suitable ϕ, increases too. So, it can use SVM approach in $\bar{D}$ to solve the binary classifier problem in $E \subset \mathbb{R}^{m_0}$ as shown Figure 2.11. Now, the non-linearly separability problem of D with $E \subset \mathbb{R}^{m_0}$ and $Y = \{y_+, y_-\}$ was reduced to construct a suitable map ϕ such that $\bar{D}$ is linearly separable and how to operate problem (DP) inside $\mathcal{H}$. The answer to these two unknowns can be drawn from functional analysis and statistical learning theories, specifically from the results obtained by James Mercer [Mercer 1909], Nachman Aronszajn [Aronszajn 1950] and Vladimir Vapnik [Vapnik and Chervonenkis 1971, Vapnik and Cortes 1995]. According to Mercer theorem it is possible to have the conditions upon which we

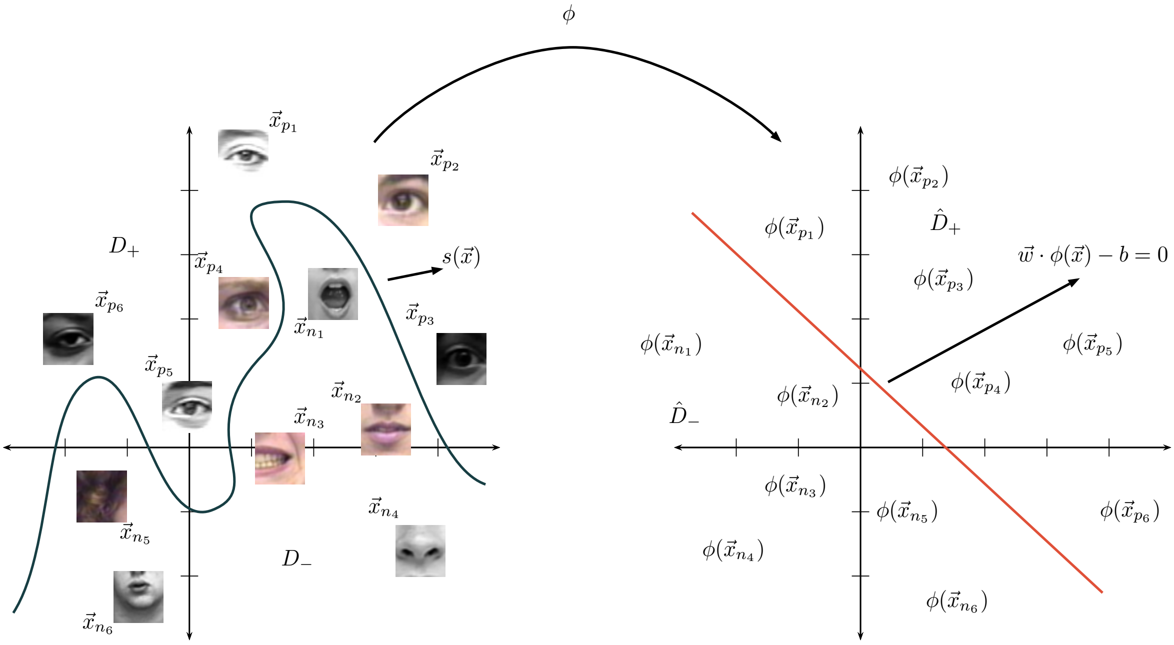


Figure 2.11: Non-linearly separability of $D = D_+ \cup D_-$ where $D_+ = \{(\vec{x}_{p_1}, y_+), \cdots, (\vec{x}_{p_6}, y_+)\}$ and $D_- = \{(\vec{x}_{n_1}, y_-), \cdots, (\vec{x}_{n_6}, y_-)\}$ is translated to a linearly separability of $\bar{D} = \bar{D}_+ \cup \bar{D}_-$ based on $E = \{\vec{x}_{p_1}, \cdots, \vec{x}_{p_6}, \vec{x}_{n_1}, \cdots, \vec{x}_{n_6}\}$. Left graph represent the input space and right graph represent $\mathcal{H}$ space where $\bar{D}$ is linearly separable based on D and ϕ.

can build a map Φ of the eigenfunctions decomposition of a definite positive function k called kernel or Mercer function. That is, if k is the continuous kernel of a integral operator $\mathcal{K}$,

$$\mathcal{K} : L_2 \to L_2$$
$$f \to \mathcal{K}f$$

$$(\mathcal{K}f)(y) = \int k(x,y)f(x)dx, \tag{2.15}$$

which is defined positive, i.e,

$$\int f(x)k(x,y)f(y)dxdy > 0 \text{ s.t. } f \neq 0, \tag{2.16}$$

then k can be expressed as a series

$$k(x,y) = \sum_{i=1}^{\infty} \alpha_i \psi_i(x)\psi_j(y), \tag{2.17}$$

with positive coefficients α_i, and

$$(\psi(x) \cdot \psi(y))_{L_2} = \delta_{ij}, \tag{2.18}$$

for i,$j \in \mathbb{N}$. By (2.17), can be extracted

$$\Phi(x) = \sum_{i=1}^{\infty} \sqrt{\alpha_i}\psi_i(x). \tag{2.19}$$

(2.19) is the map that leads the input data in E to a Hilbert space $\mathcal{H}$ with higher dimension where k has the properties of an inner product, i.e,

$$k(x, y) = \Phi(x) \cdot \Phi(y). \tag{2.20}$$

By (2.20) and the fact that Hilbert spaces preserve the geometric properties of Euclidean spaces it is possible to use the continuous kernel k to measure the similarity between the elements inside the Hilbert space $\mathcal{H}$ operating in the input space without knowing Φ. The existence of $\mathcal{H}$ was demonstrated in the work of Nachman Aronszajn [Aronszajn 1950] and Vladimir Vapnik [Vapnik and Cortes 1995].

So, replacing $\vec{x}_i \cdot \vec{x}_j$ with $\phi(\vec{x}_i) \cdot \phi(\vec{x}_j)$ in (DP) can be used the inner product properties (2.20) to solve the convex optimization problem in $\mathcal{H}$ where $\bar{D}$ is linearly separable operating in $E \subset \mathbb{R}^{m_0}$ (input space). There are many kernels known in the state-of-the-art to do that, as shown Table 2.1, and each of them can solve a problem well than other but in classification problems it is preferred to use Gaussian kernel because it usually represent better non-linearly data in the feature space according to the results obtained in [Bo, Ren and Fox 2010, Bo and Sminchisescu 2009].

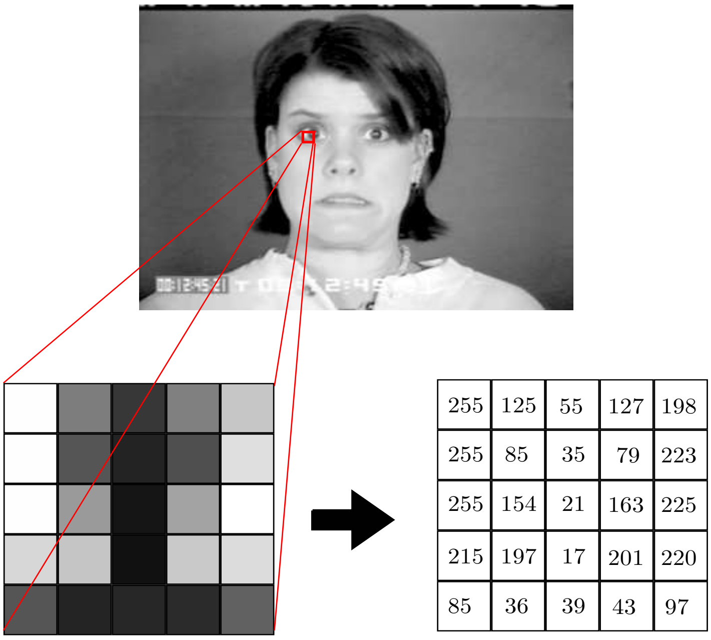
Kernel Trick can be used in other approaches related with supervised learning or unsupervised learning too. On the other hand, kernel theory is used in different way for object recognition to construct a novel type of descriptor that can used to extract different features. It details more than this in the next section.

Kernels known		
Kernel name	**Kernel**	**Parameters**
Linear kernel	$K(\vec{x}, \vec{y}) = \vec{x} \cdot \vec{y}$	none
Homogeneous polinomial kernel	$K(\vec{x}, \vec{y}) = (\vec{x} \cdot \vec{y})^d$	$d \geq 2$
Nonhomogeneous polinomial kernel	$K(\vec{x}, \vec{y}) = (\vec{x} \cdot \vec{y} + c)^d$	$d \geq 2, c > 0$
Gaussian kernel	$K(\vec{x}, \vec{y}) = e^{-\lambda \|\vec{x} - \vec{y}\|^2}$	where $\lambda = \frac{1}{2\theta^2}$ with $\theta > 0$

Table 2.1: Most common kernels used in the state-of-the-art in different learning tasks and machine learning approach. Here λ is called radial parameter and θ^2 is related with the variance in a normal distribution.

2.4.2 Learning with kernel methods in object recognition

It is well known that an object recognition problem can be seen as a supervised image classification problem and it can be solved through a machine learning algorithm as a SVM or an Artificial Neural Network. However, before train a model like them, it is important the image information characterization. This is because inside the image can be found information not related with the object of interest that introduce noise in the classification process. As shown Figure 2.12 only a part of image has the eye information and the rest consists of other parts of the person, background or other objects in general.



255	125	55	127	198
255	85	35	79	223
255	154	21	163	225
215	197	17	201	220
85	36	39	43	97

Figure 2.12: Eye information is not inside all the image only in a part of it. This is shown taking gray-scale image of CK+ database as a sample. The matrix values in right bottom are not the real values.

One of the most popular approaches to deal with this problem is Bag of Words methods or simply BOW. This method characterize an image as a local feature set. Formally, let $X = \{x_1, \cdots, x_p\}$ a set of local features (e.g. HOG features details in Section 2.3) of a image and $V = \{v_1, \cdots, v_n\}$[1] a set of features defining some object of interest, sometimes called dictionary. In BOW, each local feature is quantized in a n-dimensional binary feature vector

$$\mu(x) = [\mu_1(x), \cdots, \mu_n(x)], \tag{2.21}$$

where $\mu_i(x)$ is 1 if $x \in R(v_i)$ and 0 otherwise. Here $R(v_i) = \{x : ||x - v_i|| \leq ||x - v||, \forall v \in V\}$ is the area where x is similar to v_i using a suitable metric. The local features vectors for an image formed a normalized histogram $\bar{\mu}(X) = \frac{1}{|X|}\sum_{x\in X}\mu(x)$, where $|\cdot|$ is a cardinality of a set. Notice that $\bar{\mu}(\cdot)$ can be used to measure the similarity between two different images through of its features sets. That is, let X and Y feature sets of two different images, it is defined

$$\begin{aligned} K_{BOW}(X, Y) &= \bar{\mu}(X)^T\bar{\mu}(Y) \\ &= \frac{1}{|X||Y|}\sum_{x\in X}\sum_{y\in Y}\mu(x)^T\mu(y) \\ &= \frac{1}{|X||Y|}\sum_{x\in X}\sum_{y\in Y}f(x, y), \end{aligned} \tag{2.22}$$

as a measure of similarity. It is easy to show [Bo and Sminchisescu 2009] that $f(x, y)$ is a Mercer kernel and due to the closure property of the kernels, K_{BOW} is a Mercer kernel too. This approach

[1] X and V can be built extracting HOG features of a image.

allows measure the similarity of two images or patch of images with suitable local features selected (e.g. HOG or SIFT features) without use a type of machine learning approach.

Calculate the kernel K_{BOW} required a computational cost of $\mathcal{O}(|X||Y|)$ and when this is used with a kernel machine (e.g. SVM) has $\mathcal{O}(N^2)$ and $\mathcal{O}(N^2n^2d)$ for storage and calculate the total kernel matrix, respectively, where N is the number of images in the training set, and m is the average of cardinality of all sets. For image classification, m can be in thousands of units, so the computational cost quickly becomes of four grade when N tends to m (or tend to infinity). This make that K_{BOW} can not use for real-time detection systems in surveillance context because its high computational cost to calculate it. To address this, it can take a kernel $k(x, y)$ instead of $f(x, y)$ in (2.22) and not calculate it directly if not built a map Φ based in its inner product properties shown in (2.20) such that

$$K_s(X, Y) = \left\langle \bar{\Phi}(X), \bar{\Phi}(Y) \right\rangle, \tag{2.23}$$

where $\bar{\Phi}(X) = \frac{1}{|X|} \sum_{x \in X} \Phi$. If $\bar{\Phi}(X)$ is finite, it is possible to explicitly calculate and use it for mapping features and obtain finite descriptors that can be train trough a linear classifier (e.g. SVM). The train and test cost of this learning process is $\mathcal{O}(Nm\mathcal{D}d)$ and $\mathcal{O}(m\mathcal{D}d)$, respectively, where d is the dimension of the elements in some X and $\mathcal{D}$ is the dimension of mapping $\Phi(x)$. If the dimension of $\Phi(x)$ is low, the computational cost of this approach can be much lower than required through the calculate of the kernel functions. For example, the cost can be $\frac{1}{N}$ much lower when $\mathcal{D}$ belongs the same order than m. Note that for this method is not necessary to calculate the total kernel matrix but just the mapping $\bar{\Phi}(X)$. This type of feature descriptor is called kernel descriptor and there is only one problem to apply this approach related with the explicit construction of $\bar{\Phi}(X)$.

In practice is not easy to built a map Φ related with some kernel $k(x, y)$ because often it has a high dimension (possibly of infinite dimensional) and not all the maps lead to a significant similarity measure for determiner recognition problems. For this reason, in [Bo and Sminchisescu 2009] was proposed an approach to calculate an approximation of Φ with which is possible approximate features maps based in a set of basis vectors related with the object of interest. This approach is called Efficient Match Kernel (EMK) and we will explain more about it.

Let $\mathcal{H}$ a Hilbert space having as elements all the features vectors of the form $\psi_{high}(x)$, where *high* refer that the dimension of the space is very high. Due of this difficult, the idea for approximate features maps is project $\psi_{high}(x)$ to an features vector $\psi(x)$ within of an sub-space of $\mathcal{H}$ of low dimension based on a set of d_1 feature vectors, so build a local kernel through the inner product of the approximations. Formally, let $\{\psi(z_i)\}_{i=1}^{d_1}$, a set of mapped basis vectors z_i. Then, it can approximate a feature vector $\psi(x)$ of the form

$$\bar{v}_x = \underset{v_x}{\arg\min} \, ||\psi(x) - Hv_x||^2, \tag{2.24}$$

where $H = [\psi(z_1), \cdots, \psi(z_{d_1})]$ and $\bar{v}_x$ are coefficients of low dimensionality (space to projected).

The program (2.24) is really a convex optimization problem with unique solution

$$\bar{v}_x = (H^T H)^{-1}(H^T \psi(x)), \tag{2.25}$$

so the local kernel derived of the project vectors is

$$k_{low}(x,y) = \langle H\bar{v}_x, H\bar{v}_y \rangle = k_Z(x)^T K_{ZZ}^{-1} k_Z(y), \tag{2.26}$$

where k_Z is a vector $d_1 \times 1$ with $\{k_Z\}_i = k(x, z_i)$ and K_{ZZ} is a $d_1 \times d_1$ matrix with $\{K_{ZZ_{ij}} = k(z_i, z_j)\}$. For $G^T G = K_{ZZ}^{-1}$ (where K_{ZZ}^{-1} is positive define), the local features map is

$$\Phi(x) = G k_Z(x), \tag{2.27}$$

and the total resulting feature map is

$$\bar{\Phi}(X) = \frac{1}{|X|} G \left[\sum_{x \in X} k_Z(x) \right], \tag{2.28}$$

with a computational cost $\mathcal{O}(md_1 d + d_1^2)$ for a set of local features. The constructing map can be used with a linear SVM without kernel trick which allows optimize the computational cost of classification process.

In Chapter 3 is details the proposed approach to extract and train features that complement the results obtained by Viola and Jones detector. This approach is based in EMK and the linear version of SVM explained in this chapter.

2.5 Summary

In this chapter a brief review of the main works addressing the eye detection problem in multiple contexts and considerations that must be taken into account for real-time applications has been done. Especially, it is explained the approaches dealing better the challenges can be found in video surveillance context in uncontrolled environments.

First, we made a description of the work developed by Viola and Jones [Viola and Jones 2004] emphasizing its main contributions and disadvantages. Then, it was mentioned some current works based on Viola and Jones investigation to obtain better results in eye detection applications or systems. Due to of the low precision of the VJD in uncontrolled environments it made a brief review of the most important works that address the detection problem with more sophisticated methods where works as HOG [Dalal and Triggs 2005] and BOW [Felzenszwalb et al. 2010] are highlighted. Finally, it made an introduction of kernel theory to envelop the reader with the basis enough to understand research related with SVM [Vapnik and Cortes 1995] and EMK [Bo and Sminchisescu 2009].

In Chapter 3 we explain the methodology and design of the proposed method based in the Viola and Jones research and the approach of Efficient Match K ernel which never was used for eye detection problem before.

Chapter 3

The proposed methodology

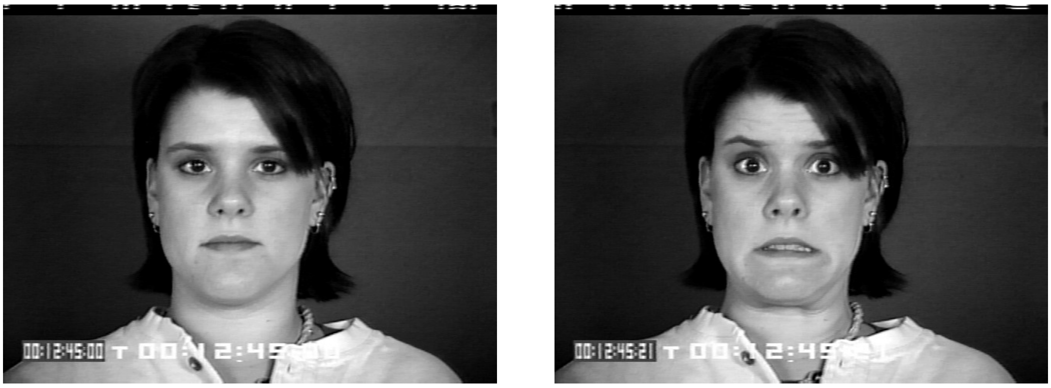
3.1 Introduction

In this chapter is described the methodology used to develop the eye detector proposed in this work. Section 3.2 exposes a sample problem to be solved using two types of database. Section 3.3 present details about features used in Viola and Jones work and why is important to complement it with other types of features. Section 3.4 describe all the methods that will be used to solve the problem of eye detection in unconstrained context. Finally, Section 3.5 details the proposed approach design using the resulting outputs of Viola and Jones detector as input data.

3.2 Eye detection problem in unconstrained settings

Many works about eye detection [Awais, Badruddin and Drieberg 2013, Chen et al. 2014, Choi, Han and Kim 2011, Hyunjun, Jinsu and Jaihie 2014, Oliveira et al. 2012] inside the object recognition context in Computer Vision have been doing currently. However, it is not common that these type of research deals the problem in a unconstrained settings. In this work is addressing the eye detection problem in unconstrained environments but without ignoring features such as performance and accuracy of detection. For do this, it is necessary introduce the reader inside both constrained and unconstrained contexts.

When we refer to a constrained or unconstrained context on eye detection refer to contexts in which the input information was collected for some detection problem. That is, in this work, the input images containing the object of interest. Many works about eye, face or human detection usually use collected images in environments with lighting, position and background constrained. For example, in [Lucey et al. 2010] was used CK+ database which contains people faces captured at the same distance, constant illumination and background bit variable. This make that a detection problem be more easy to solve because the environments features do not have additional information to analyze. Although CK+ database was designed for researches about detecting expressions also it is useful to experiments as we developed in this work for the eye detection. A samples of CK+ database is shown in Figure 3.1.



(a) Neutral expression.

(b) With expression.

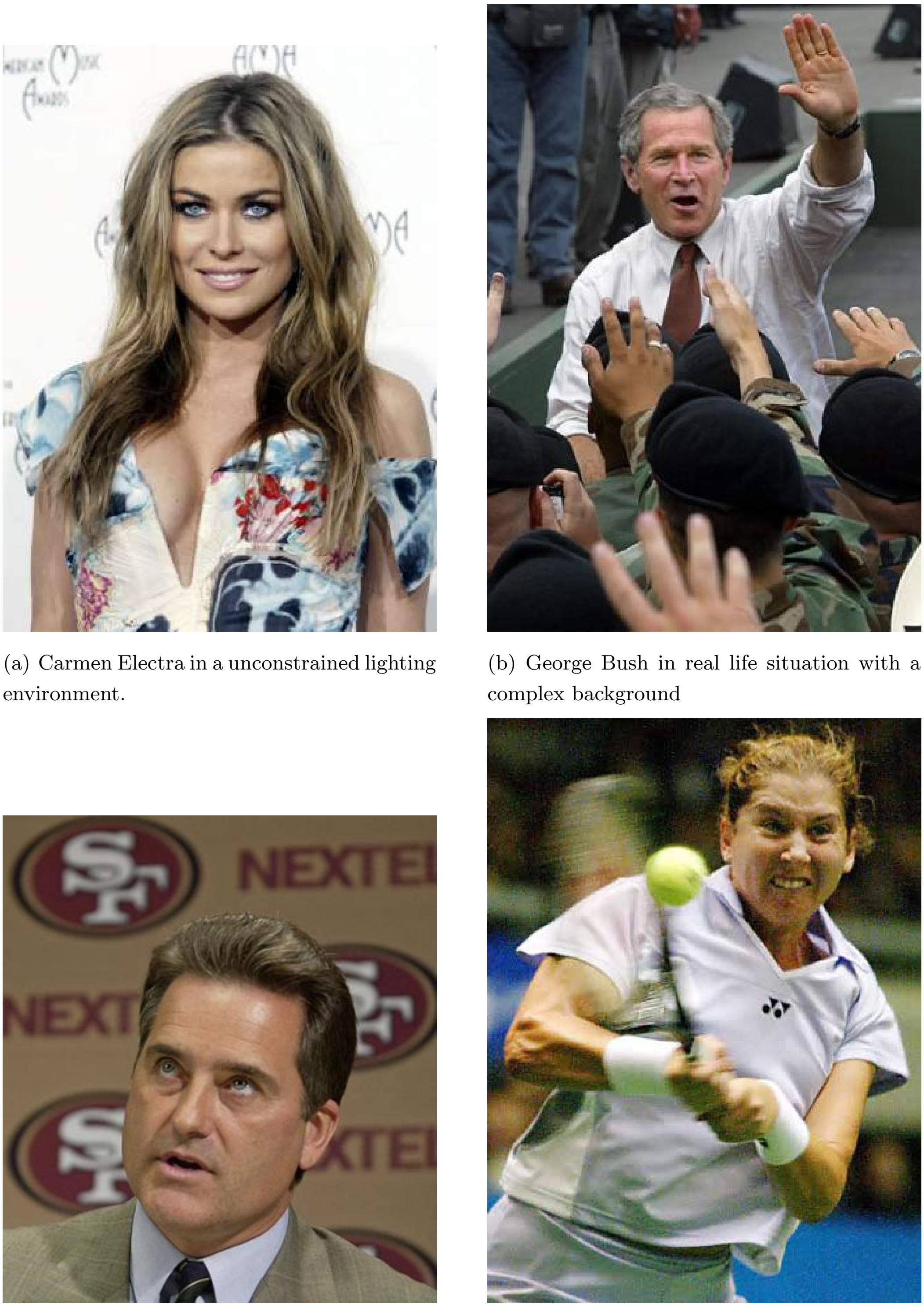
Figure 3.1: A samples of CK+ database [Lucey et al. 2010].

For detection problems in real life, for example in surveillance context, the collected images for detection usually are in unconstrained environments. FDDB database [Jain and Learned-Miller 2010] is not a database that contain images of surveillance environments but has an image collection of people captured in unconstrained environments. As shown Figure 3.2, it is not possible to know what type of environment we will have in a situation in real life and what type of detection challenge it will present. There are many challenges in this type of context that hindering detection task such as occlusion, non rigid variation, deformation, photometric, viewpoint and variable structure. To address these challenges in eye detection it is propose an approach based in EMK explained in 2.4.2 that can extract a complementary local features of the VJD resulting outputs in a fast way.

3.3 Viola and Jones detector as a first layer

The research made by Viola and Jones [Viola and Jones 2004] is one of the most popular in the object detection area. Currently, it is used in detection of several objects as humans, faces, eyes, cars or pets and it is used as support of many research in the state-of-the-art too. This is because the capacity of Haar-like features (explained in Chapter 2) to extract information patterns in the input image helping in final detection of the object of interest. In addition, the calculating necessary for each detection is not expensive than in other approaches because the integral image representation and the form to calculate it. However, the Viola and Jones detector has a disadvantage related with its results obtained in real life contexts. Due to the unconstrained environments several amount of false positives are obtained and this rules out its use in surveillance detection systems. A sample this is shown in Figure 3.3.

In this work is propose to use a second features class and a linear classifier to discriminate the resulting false positives of the Viola and Jones detector without adding a significant computational cost. Thus the advantages and benefits of that detector are preserved and only its results are optimized. The key is use the VJD as a generator of reduced ROIs and construct a new detector at the end of the Viola and Jones cascade classifier design as shown Figure 3.4. This detector takes the VJD resulting regions to build kernel features based on EMK. Then, a linear SVM classifier



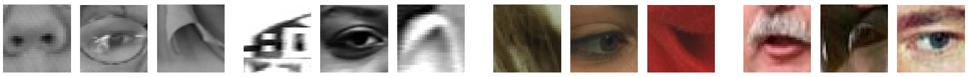
(a) Carmen Electra in a unconstrained lighting environment.

(b) George Bush in real life situation with a complex background

(c) A man with complex background.

(d) A tennis player in action.

Figure 3.2: A samples of FDDB database [Jain and Learned-Miller 2010] in unconstrained environments.



(a) BIOid. (b) CK+. (c) FERET. (d) FDDB.

Figure 3.3: Samples of resulting outputs of Viola and Jones detector.

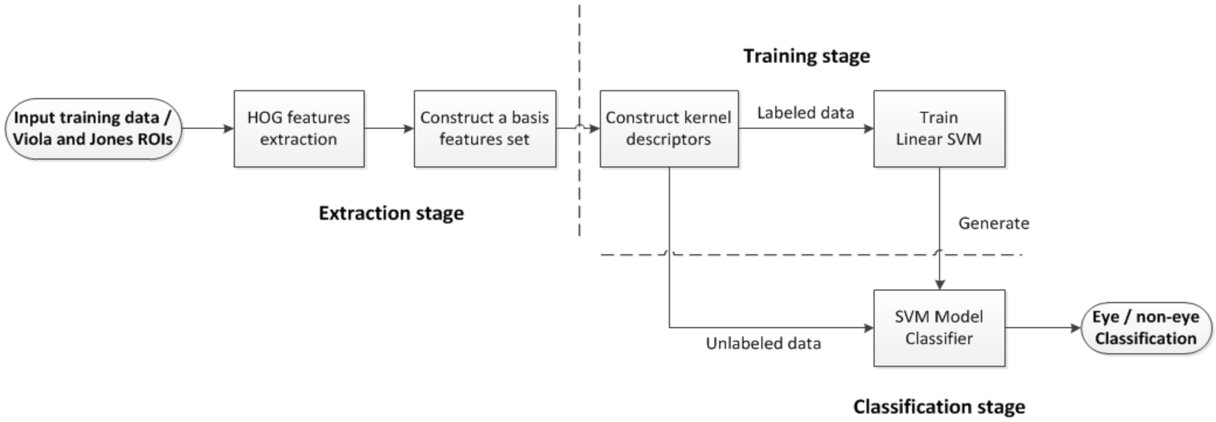


Figure 3.4: Flow diagram of cascade classifier with efficient kernel detector. Here, Viola and Jones detector is used to generate outputs and then an efficient kernel approach based on HOG features and a linear classifier are used to improve the results.

is used to discriminate the false positives of the true positives. It will be demonstrated that with this approach is possible to discard up to 98% of false positives of the Viola and Jones detector.

3.4 EMK and SVM for eye detection

The simplicity of Haar-like features used in the Viola and Jones detector (VJD) [Viola and Jones 2004] introduce a high amount of false positives in the resulting outputs in uncontrolled environments. This can be related to the object of interest and how is projected inside the image. In the case of the eye detection in a unconstrained environments, many of the patterns that represent to the eye through of the Haar-like features can be confused with the features found for other objects (or simply part of the background) in the image environment. In addition, it is not easy deals with eye recognition in unrestrained environments even for the human eye. For example, looking in Figure 3.3.d a mouth can be mistaken for an eye if you take a quick look. To addressing this problem in an eye detector it is possible use a second type of features that are capable to extract patterns with which possible difference between a false positive of a true positive region and address the usual challenges presented in this type of detection.

HOG features details in Section 2.3 can be used to characterize an VJD output as a set of features. These features can complement the patterns detected using Haar-like features. As shown Figure 3.6, it is possible to distinguish between false positives and true positives eye patterns inside VJD resulting outputs seeing the HOG features represented as a star structure in each sample. The HOG features set generated with each HOG feature descriptor can be used as input data to

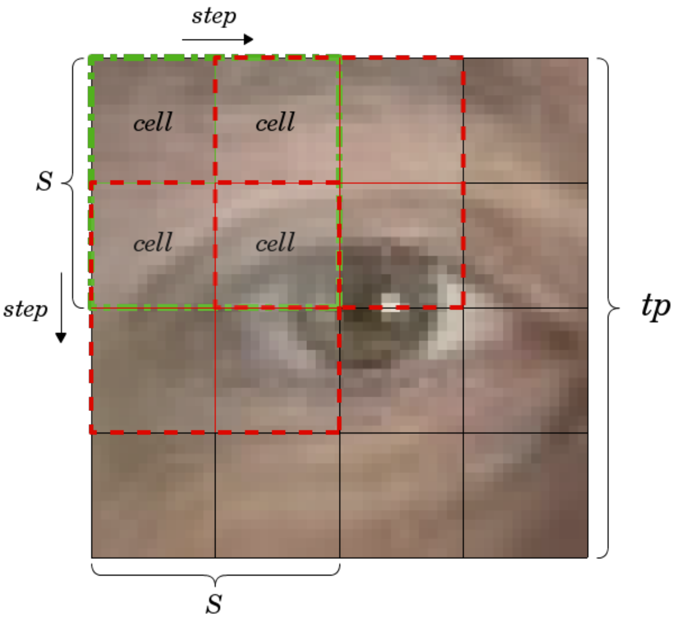


Figure 3.5: Green square in a output represent the structure of one HOG feature composed by 4 cell with 9 bins for each one. Based on the work of Dalal and Triggs [Dalal and Triggs 2005], it is enough consider the 9 oriented bins in $0° - 180°$. According to that, it will has 36 dimensions per features.

train a SVM classifier. That is, let $E = \{X_1, \cdots, X_L\}$ a set of VJD resulting outputs. Suppose that each output has $tp \times tp$ pixels and a block structure per output as shown in Figure 3.5. Then the amount of features c per output is defined as

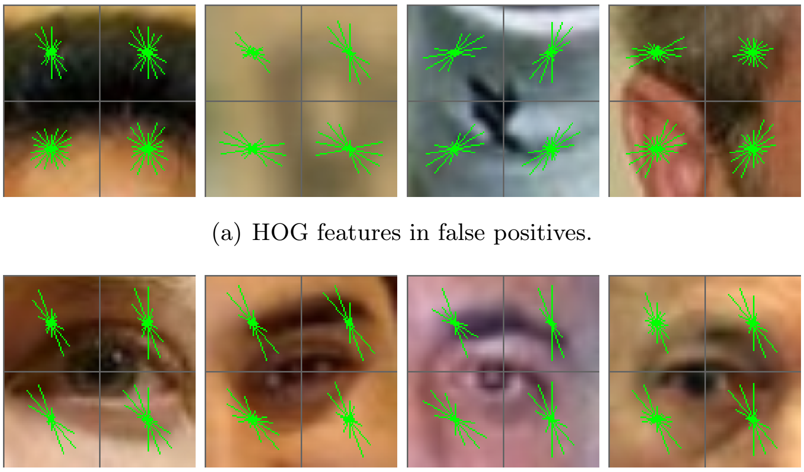
$$c = \left(\frac{tp}{S} + p \times \frac{tp}{S}\right) \times \left(\frac{tp}{S} + p \times \frac{tp}{S}\right), \tag{3.1}$$

where $S \times S$ represent the dimension per block and p is the step size between features. If we consider b bins and 4 cells per block, the HOG features representation for an output is defined as $X_s = (x_{s1}, \cdots, x_{sc}) \in \mathbb{R}^{b \times 4 \times c}$ with $s = 1, \cdots, L$. Here x_{sm} with $m = 1, \cdots, c$ represent a local features of the output X_s. So, it is possible to solve the optimization convex problem (DP) (2.11) based on E using this feature structure constructing a decision function like (2.12) of the form

$$f(X_s) = \begin{cases} +1, & \text{if } \vec{w}^* \cdot X_s - b^* > 0, \\ -1, & \text{if } \vec{w}^* \cdot X_s - b^* < 0, \end{cases} \tag{3.2}$$

where $X_s \notin E$ is an unlabelled output to classify, $+1$: represent an eye detected and -1 : represent an not eye detected. This approach is a good way improve the results of the detection. However, it is not guarantee that the training set D (based on E) is linearly separable. Although that could be solved using Kernel Trick, there is a better solution using Efficient Match Kernel (EMK).

Use Kernel Trick in SVM approach is an alternative to solve the linearly separability problem of a training set D but it can not control the dimensionality of feature space $\mathcal{H}$ which introduce an additional computational cost. Instead, it is possible to construct a features space $\mathcal{H}_Z$ based on a set of HOG local features Z where can be compared the mapped local features of the elements



(a) HOG features in false positives.

(b) HOG features in true positives.

Figure 3.6: HOG features samples extracted of the Viola and Jones resulting outputs using FDDB database. It is possible to see the difference between the false and true positives looking inside their HOG features.

on E through a special map related with the Mercer conditions 2.16. The key of this approach is based on EMK which need a set of mapped features vectors $\{\psi(z_i)\}_{i=1}^{d_1}$, with $z_i \in Z$, to construct a approximation of map $\bar{\Phi}$ as shown equation (2.27). In the case of eye detection, Z represent the basis vector set based on HOG local features that represent better one eye. To understand this, suppose that set Z is defined through three elements of a sample $X_1 = (x_{11}, x_{12}, x_{13}, x_{14})$ such that $Z = \{x_{11}, x_{12}, x_{13}\}$. This means that all the mapped elements of Z with Φ forms a generator set of the space $\mathcal{H}_Z$ where local features of any other mapped sample $X_s = (x_{s1}, x_{s2}, x_{s3}, x_{s4})$, with the same Φ, can be represented as weighted sum of the elements of Z as shown Figure 3.7. Also, it could be calculated the similarity of local mapped features of two different samples in $\mathcal{H}_Z$ through a suitable metric, norm or even a kernel defined inside this space.

Using this approach can be used the map $\bar{\Phi}$ to construct descriptors based on the samples of E generating a descriptors set of the form $\{\bar{\Phi}(X_1), \cdots, \bar{\Phi}(X_L)\}$ in a high dimension space where the training set $\bar{D}$ could be linearly separable. So, it is possible to train a learning machine like SVM to construct a linear classifier to identify if a sample $X_s \notin E$ is an eye or not. This method only introduce an additional step when the elements of Z are found through a unsupervised learning method as PCA [Barber 2012] or K-Means [James et al. 2013]. However, this does not affect the last detection performance.

3.5 Efficient kernel eye detector design

The proposed eye detector has 3 stages that one can call extraction, training, and classification stage, respectively. For these, we have selected random resulting outputs of the VJD [Viola and Jones 2004] as input data (shown in Figure 3.3) and these are divided in 25%, 50% and 25% respectively for each stage.

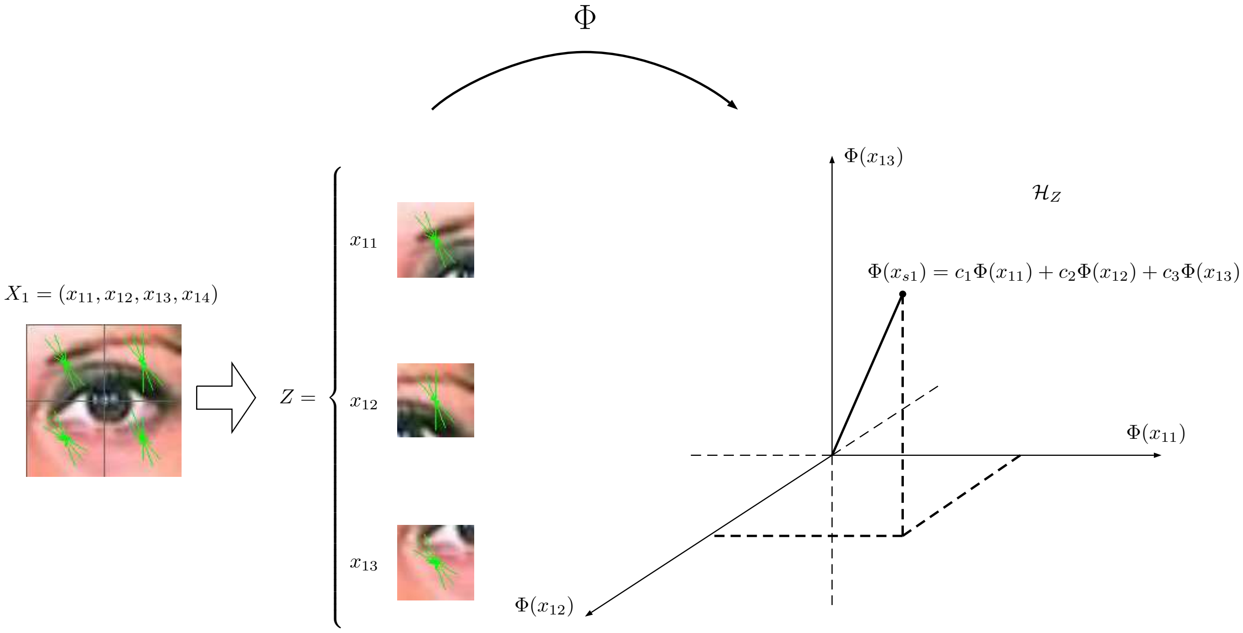
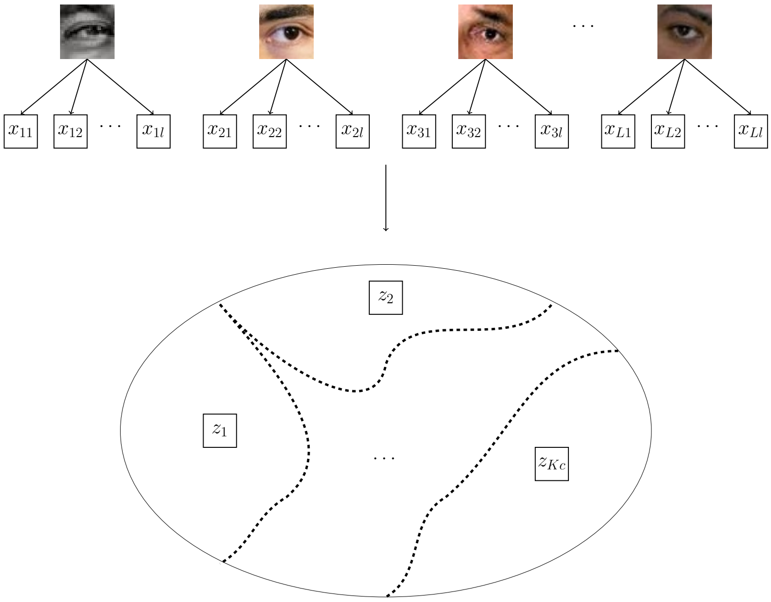

Φ
$X_1 = (x_{11}, x_{12}, x_{13}, x_{14})$
$Z =$
x_{11}
x_{12}
x_{13}
$\Phi(x_{13})$
$\mathcal{H}_Z$
$\Phi(x_{s1}) = c_1\Phi(x_{11}) + c_2\Phi(x_{12}) + c_3\Phi(x_{13})$
$\Phi(x_{11})$
$\Phi(x_{12})$

Figure 3.7: $\mathcal{H}_Z$ is constructed based on a set $Z = \{x_{11}, x_{12}, x_{13}\}$ of HOG local features generated using the outputs of VJD and map Φ. Let $X_s = (x_{s1}, x_{s2}, x_{s3}, x_{s4})$, a mapped local feature can be represented as $\Phi(x_{s1})$ in this space based on $\{\Phi(x_{11}), \Phi(x_{12}), \Phi(x_{13})\}$ where $c_1, c_2, c_3 \in \mathbb{R}$.

To simplify the implementation of the proposed method was used the version of the VJD in OpenCV Library [Bradski 2000]. This implementation consider multi-scale detection but for our propose method we only use a standard size per output as input data because the ROIs were considerably reduced for VJD. This size depends on the structure chosen for the HOG feature extraction (in each stage) and the average dimension of the outputs labeled as true positive in each database. As shown Figure 3.5 and based on the work done by Dalal and Triggs in [Dalal and Triggs 2005], we going to consider a $tp \times tp$ output and a $S \times S$ block composed of 4 cells with 9 bins per output for the construction of each descriptor HOG where $tp \in \{36, 44, 60, 72\}$. This selection is related with the average dimension of the outputs labeled as true positive per database and motivated for one of features found in [Oliveira et al. 2012]. It will be seen in Chapter 4 that this structure will take a good experimental results.

On extraction stage a HOG features framework is adopted to characterize each ROI. Then, K-Means [James et al. 2013] clustering is used to find a basis vector set Z, as shown Figure 3.8 where $z_1, z_2, \cdots, z_{Kc}$ are the centroids founded with the method. The idea is to extract the more important features defining one eye and differentiate them from other objects (e.g, part of the environment or part of other objects) on a subregion of the ROI. It forms a feature eye space where the similarity between parts of different ROIs can be measured. The algorithm that implement this stage is shown in Algorithm 1.

On training stage we need to supervise VJD outputs as input data, that is, choosing if a kernel descriptor characterized as HOG feature output is an eye or non-eye, and label it with a scalar value that parametrizes them (two classes). For that, HOG features descriptors [Felzenszwalb et al. 2010] are generated for each VJD output. Then, for each HOG descriptors are constructed a kernel



$$Z = [z_1, z_2, z_3, \cdots, z_{Kc}]$$

Figure 3.8: Extraction stage - In extraction stage the basis of features vectors is found and then these are used to construct the kernel descriptors in training stage. $X_1 = \{x_{11}, \ldots, x_{1l}\}$, $X_2 = \{x_{21}, \ldots, x_{2l}\}$, $\ldots$, $X_L = \{x_{L1}, \ldots, x_{Ll}\}$ are the L features sets of L true positives outputs of the VJD that will be the input data for an unsupervised learning method to find a basis set Z with Kc features vectors. In this work we will use a K-Means method using Kc as clusters number and we will analyze the behavior of this parameter in our experiments.

```
Data:
$\{X_1, X_2 \cdots, X_T\}$: A set of resulting ROIs of Viola and Jones detector;
$tp$: Size of a ROI side;
$S$: Size of a block;
Result:
$G$: A matrix to construct Efficient Kernel Descriptors as shown in (2.27);
$Z$: A matrix containing all the basis vectors;
$Kc$: Clusters number;
Parameters: Initialization
$K$: A matrix in (2.26);
$c$: Amount of features per ROI defined as $\left(\frac{tp}{S} + p \times \frac{tp}{S}\right) \times \left(\frac{tp}{S} + p \times \frac{tp}{S}\right)$;
descriptorValues: Array containing the HOG features extracted per ROI;
$Ps$: A matrix that containing the information of the HOG features of all ROIs where
 $Ps(i, \cdot)$ is a feature vector;
$p = 0.5$: size step per features;
$j = 1$;
for $i = 1$ ; $i \leq T$ do
    ToGrayScale($X_i$);
    Equalize($X_i$);
    Resize($X_i$,$tp$);
    descriptorValues[$i$] = HOGDescriptor($X_i$);
    for $k = 1$ ; $k \leq c$ do
        $Ps(j, k)$ = descriptorValues[$i$];
        $j++$;
    end
end
$Z$ = FindCentersKmeans($Ps$,$Kc$);
$K$ = ConstructK($Z$);
$K^{-1}$ = InvertMatrix($K$);
$G$ = CholeskyDescomposition($K^{-1}$);
return ($Z$,$G$)
```

Algorithm 1: Pseudo-code for extraction stage

```
Data:
${X_1, X_2 \cdots, X_{T_e}}$: A set of resulting ROIs of Viola and Jones detector;
$tp$: Size of a ROI side;
$S$: Size of a block;
$G$: A matrix to construct Efficient Kernel Descriptors as shown in (2.27);
$Z$: A matrix containing all the basis vectors;
Result:
$f$: Decision function from SVM model;
Parameters: Initialization
$c$: Amount of features per ROI defined as $\left(\frac{tp}{S} + p \times \frac{tp}{S}\right) \times \left(\frac{tp}{S} + p \times \frac{tp}{S}\right)$;
$p = 0.5$: size step per features;
descriptorValues: Array containing the features extracted per ROI;
$Fv$: A matrix that containing the information of the features of each ROIs;
$K_d$: Kernel descriptor;
for $i = 1$ ; $i \leq T_e$ do
    ToGrayScale($X_i$);
    Equalize($X_i$);
    Resize($X_i$,$tp$);
    descriptorValues[$i$] = HOGDescriptor($X_i$);
    for $k = 1$ ; $k \leq c$ do
        $Fv(k)$ = descriptorValues[$i$];
    end
    $K_d$ = ConstructEfficientKernelDescriptor($Fv$,$Z$,$G$);
    trainData($i$) = $K_d$;
    if $X_i$ is True positive then
        labelData($i$) = 1;
    else
        labelData($i$) = 0;
    end
end
$f$ = ConstructDesicionFunctionSVM(trainData,labelData);
return ($f$)
```

Algorithm 2: Pseudo-code for training stage

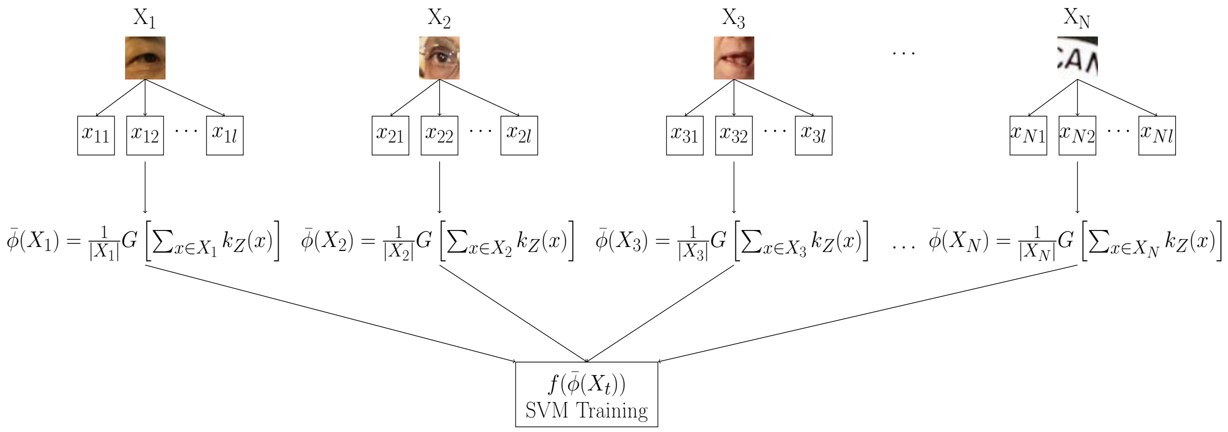

X1
X2
X3
XN
...
x_{11} x_{12} ··· x_{1l}
x_{21} x_{22} ··· x_{2l}
x_{31} x_{32} ··· x_{3l}
x_{N1} x_{N2} ··· x_{Nl}
$\bar{\phi}(X_1) = \frac{1}{|X_1|}G\left[\sum_{x\in X_1} k_Z(x)\right]$
$\bar{\phi}(X_2) = \frac{1}{|X_2|}G\left[\sum_{x\in X_2} k_Z(x)\right]$
$\bar{\phi}(X_3) = \frac{1}{|X_3|}G\left[\sum_{x\in X_3} k_Z(x)\right]$
...
$\bar{\phi}(X_N) = \frac{1}{|X_N|}G\left[\sum_{x\in X_N} k_Z(x)\right]$
$f(\bar{\phi}(X_t))$
SVM Training

Figure 3.9: Training stage - In training stage the kernel descriptors are trained in a linear Support Vector Machine. For this, it is necessary supervised input data, that is, we need to label kernel descriptors generated with the output of the VJD. Here, $X_1, \ldots, X_N$ are the sets of features which would be used to construct the kernel descriptors $\bar{\phi}(X_1), \ldots, \bar{\phi}(X_N)$. These descriptors are labelled as eye or other thing for the training. As a result, after training, a decision function $f(\cdot)$ is obtained that will be used to the eye detection.

descriptor which is labelled as eye or non-eye. Finally, these labelled descriptors are used to train a linear SVM to obtain a linear classifier $f(\cdot)$ as shown Figure 3.9. This classifier is constructed in a feature space where the mapped input data is probably linearly separable. The algorithm that implement this stage in shown in Algorithm 2.

Finally, on classification stage, an efficient kernel descriptor is constructed based on an unlabelled output and it is labelled by the linear classifier $f(\cdot)$ built on training stage. This classification is essentially the final detection task of the proposed method. This is show in Figure 3.10 and the reader can see the algorithm that implement this stage in Algorithm 3.

3.6 Conclusion

In this Chapter it was present a proposed methodology of an eye detector based on an Efficient Match Kernel approach and we will try to improve the precision of the well-know VJD using it. The idea is used this propose to complement security and surveillance systems where it is require a fast response and high precision to alert users.

Proposed eye detector takes ROIs resulting of Viola and Jones detector and extract local features based in kernel theory. This features can be used as a complement for Haar-like features used in the work of Viola and Jones without increase significantly its computational cost and improve its precision though of a linear SVM. It used a cascade architecture and a linear classifier to discriminate false positives such that the precision is improved in the final detection.

In Chapter 4 will be shown that the results obtained by the proposed method significantly improve the results obtained by the Viola and Jones detector.

```
**Data:**
$\{X_1, X_2 \cdots, X_{T_c}\}$: A set of resulting ROIs of Viola and Jones detector;
$tp$: Size of a ROI side;
$S$: Size of a block;
$G$: A matrix to construct Efficient Kernel Descriptors as shown in (2.27);
$Z$: A matrix containing all the basis vectors;
$f$: Decision function from training stage;
**Result:**
$p$: Define is a ROI is an eye or not;
**Parameters:** Initialization
$c$: Amount of features per ROI defined as $\left(\frac{tp}{S} + p \times \frac{tp}{S}\right) \times \left(\frac{tp}{S} + p \times \frac{tp}{S}\right)$;
$p = 0.5$: size step per features;
descriptorValues: Array containing the features extracted per ROI;
$Fv$: A matrix that containing the information of the features of each ROIs;
$K_d$: Kernel descriptor;
for $i = 1$ ; $i \leq T_c$ do
    ToGrayScale($X_i$);
    Equalize($X_i$);
    Resize($X_i$,$tp$);
    descriptorValues[$i$] = HOGDescriptor($X_i$);
    for $k = 1$ ; $k \leq c$ do
        $Fv(k)$ = descriptorValues[$i$];
    end
    $K_d$ = ConstructEfficientKernelDescriptor($Fv$,$Z$,$G$);
    $p = f(K_d)$;
    if $p = 1$ then
        $X_i$ is an eye;
    else
        $X_i$ is not an eye;
    end
end
return ($p$)
```

Algorithm 3: Pseudo-code for classification stage

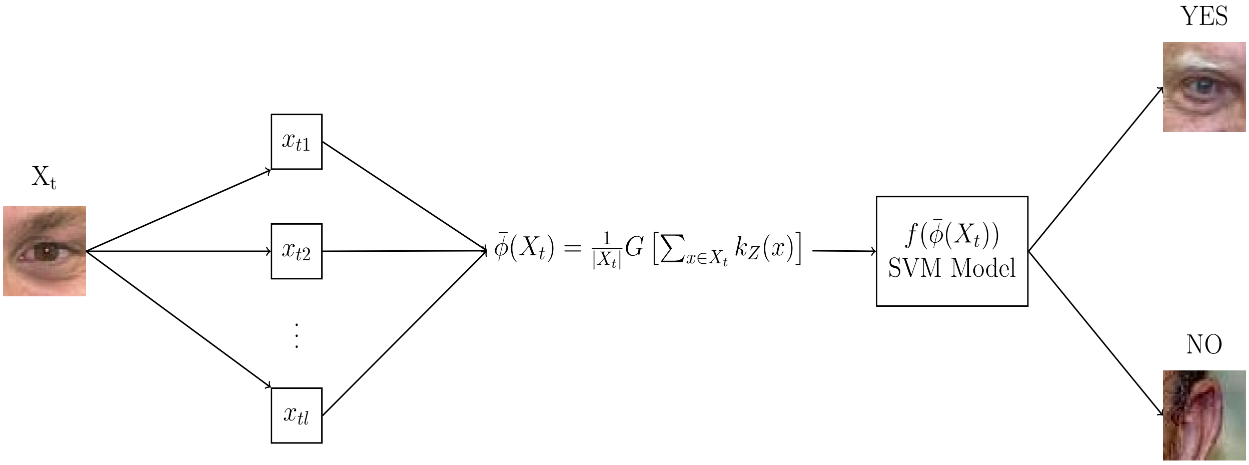

X_t
x_{t1}
x_{t2}
$\vdots$
x_{tl}
$\bar{\phi}(X_t) = \frac{1}{|X_t|} G \left[\sum_{x \in X_t} k_Z(x)\right]$
$f(\bar{\phi}(X_t))$
SVM Model
YES
NO

Figure 3.10: Classification stage - In classification stage, the detection is performed. Here, X_t is a set of features of an unknown output of the VJD used to construct a new kernel descriptor. This descriptor is evaluated using $f(\cdot)$ for the final detection.

Chapter 4

Experiments

4.1 Introduction

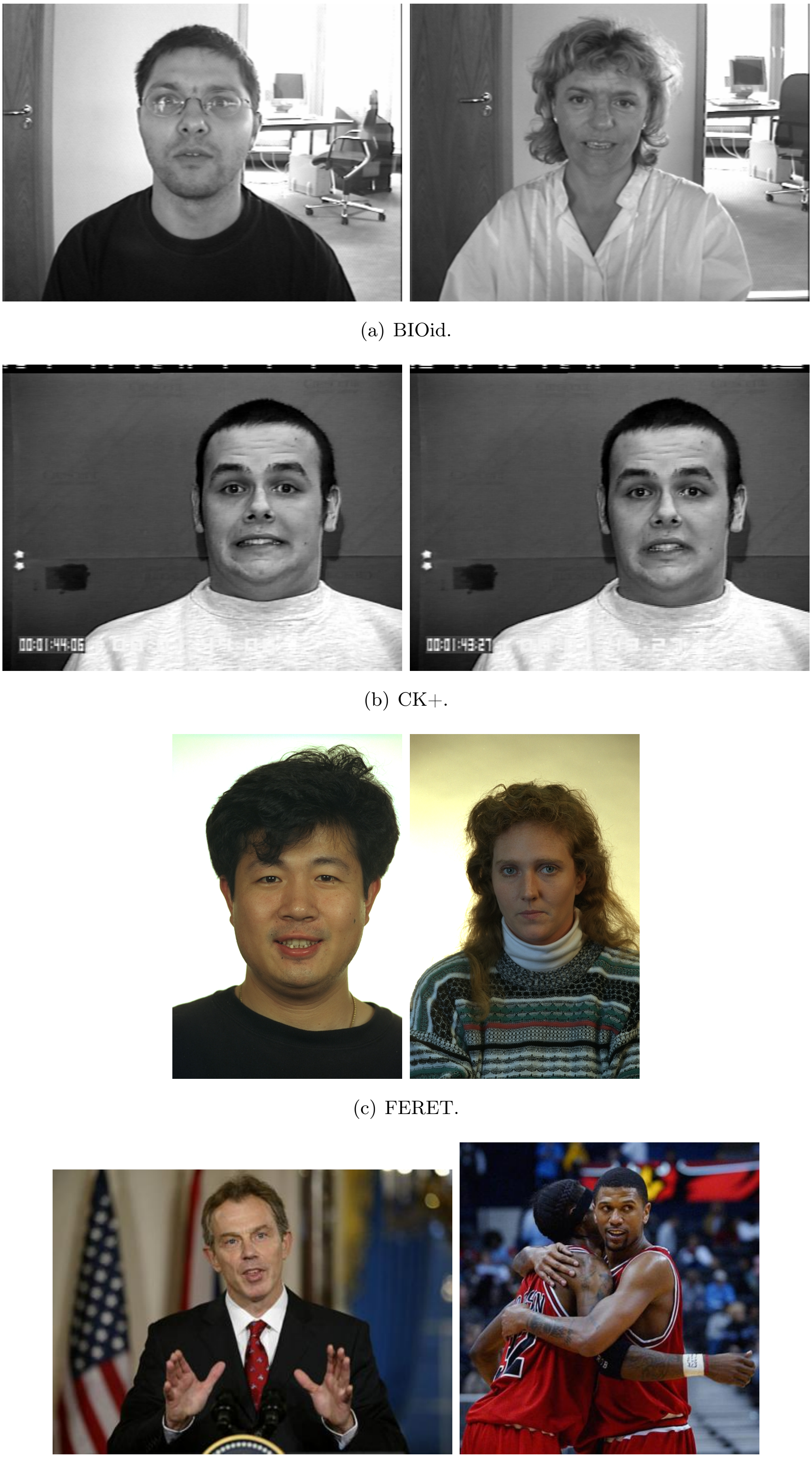
In this chapter we described the results obtained by the proposed detector in this work. First, it details the databases used for the tests and the parameters to use in each case. It known previously that the resulting outputs obtained by the Viola and Jones detector will be used for comparisons.

Section 4.2 details all databases to use. In Section 4.3 we explain the results obtained by the proposed detector highlighting two types of tests in particular.

4.2 Datasets

Five known databases are used here, for extraction, training and classification stages. They have been used by major research in Computer Vision area such as [Jesorsky, Kirchberg and Frischholz 2001, Wu and Ji 2014, Jain and Learned-Miller 2010]. They are the FDDB [Jain and Learned-Miller 2010], BioID [Jesorsky, Kirchberg and Frischholz 2001], CK+ [Lucey et al. 2010] and FERET databases [Phillips et al. 2000]. Among all of them, the FDDB dataset has the more complex environments containing uncontrolled settings detailed in Section 3.3. The other ones contain images in more controlled environments but with different formats. A samples of these is shown in Figure 4.1.

Our method takes the Viola and Jones detector as a generator of regions of interest. This means that the ROIs outputted by Viola and Jones detector are then used as input for building the kernel descriptors as was detailed in Section 3.3. For all stage is consider 4817 resulting ROIs for each database. Also, ROIs in each database were resized such that in BioID 36 × 36 ROIs, in CK+ 72 × 72 ROIs, in Feret DVD1 60 × 60, in Feret DVD2 72 × 72 and in FDDB 44 × 44 ROIs are considered according to the structure explained in Section 3.5.



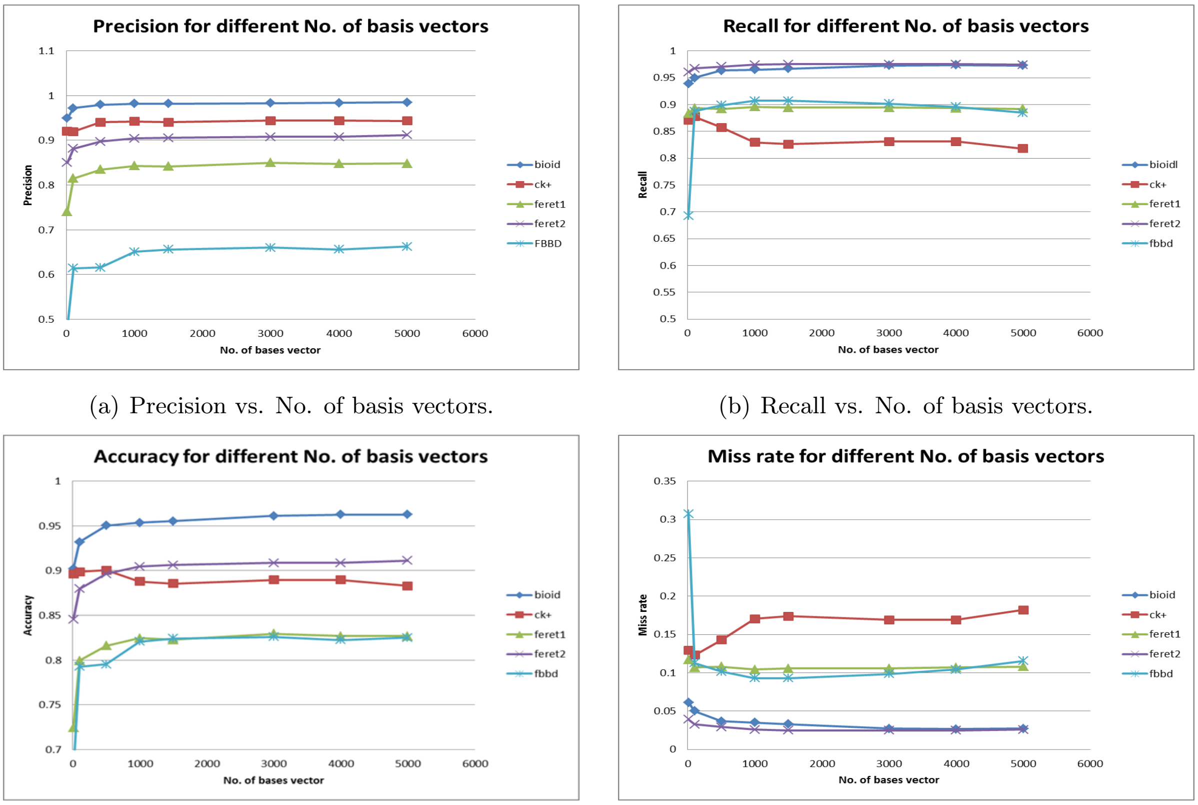
(a) BIOid.

(b) CK+.

(c) FERET.

(d) FDDB.

Figure 4.1: Samples of the databases used in each stage of the proposed method.



(a) Precision vs. No. of basis vectors.

(b) Recall vs. No. of basis vectors.

(c) Accuary vs. No. of basis vectors.

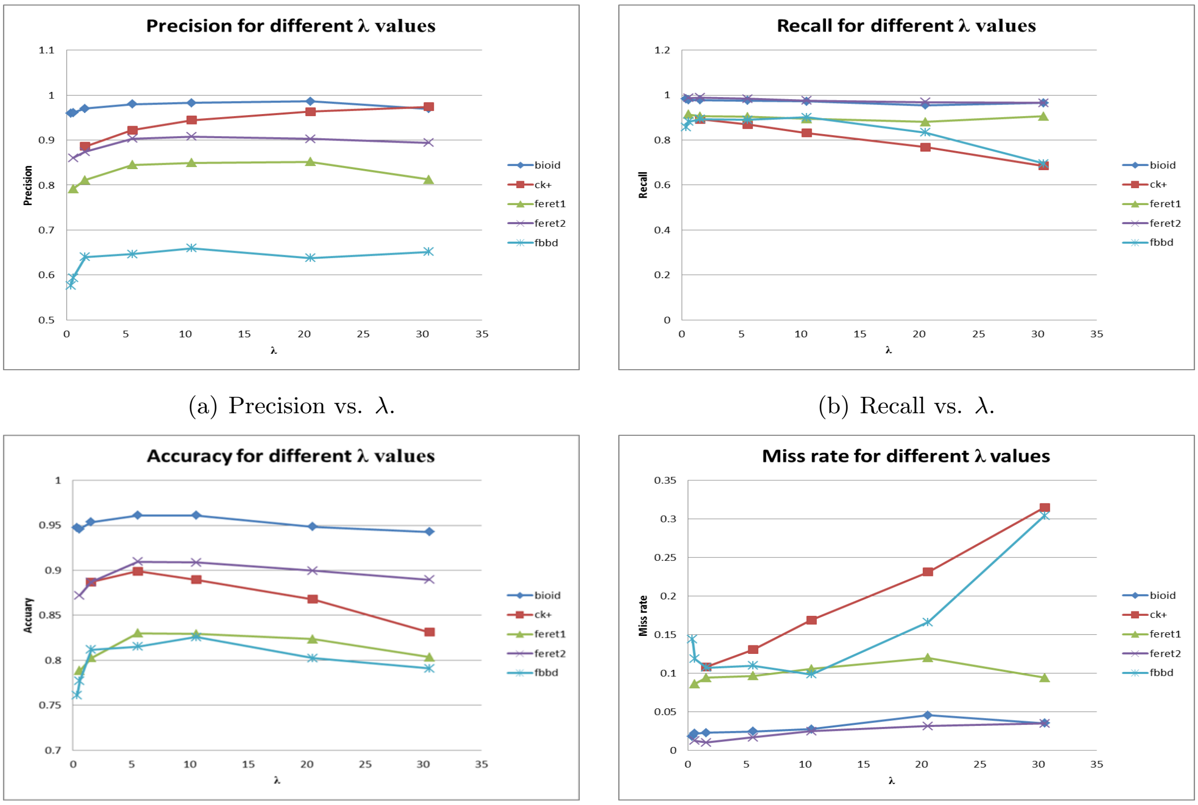
(d) Miss rate vs. No. of basis vectors.

Figure 4.2: Charts of the eye detector behavior by varying the among of basis vectors.

4.3 Results and discussion

In this section we present the results obtained testing the proposed eyes detector using as input data the VJD resulting ROIs taking into account BIOid [Jesorsky, Kirchberg and Frischholz 2001], CK+ [Lucey et al. 2010], FERET [Phillips et al. 2000] and FDDB [Jain and Learned-Miller 2010] databases. It is important to say that for the experiments we did not consider the color space information, only the luminance (i.e. gray-scale) is used in the three stages. Color images could be directly transformed to gray ones, and the detection of the eyes do not use color features. Some authors [Shuo and Chengjun 2011, Kumar, Raja and Ramakrishnan 2002] take into account color, however in the present work, for the nature of HOG features and the kernel descriptors, the color space information will only increase the data size and does not impact on improving the final detection.

Precision obtained by the VJD with all databases is shown on the Table 4.1 and we going to generate the same indicator testing the proposed method. First we made an analysis considering the indicators as recall, accuracy and miss rate when the amount of the basis vectors is changed on the extraction stage and setting the parameter $\lambda = 10.55$ on the Gaussian kernel used to construct the efficient kernel descriptors. The goal of this test is validate if there is an improvement on the detection when the amount of the basis vectors increasing or decreasing. According to that, it is possible to see on Figure 4.2.a that 90% of precision is obtained using BioID, CK+ and Feret



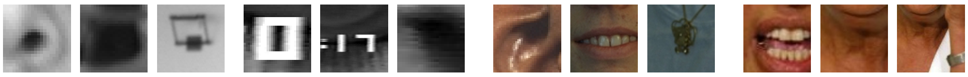
(a) Precision vs. λ.

(b) Recall vs. λ.

(c) Accuary vs. λ.

(d) Miss rate vs. λ.

Figure 4.3: Charts of the eye detector behavior by varying λ.



(a) BIOid.

(b) CK+.

(c) FERET.

(d) FDDB.

Figure 4.4: Samples of false positives ROIs discarded by the proposed method.

DVD2 database when the amount of basis vectors is over 1000. These results improve significantly the results of VJD.

On the specific case of the FDDB database, we obtained a 66% of precision, 20% more than in the case of VJD with a recall of 90% as shown Figura 4.2.b. Taking into account this database contain scenes in uncontrolled environments we obtained a good improvement. This means that the basis vector set based on HOG features could represent better some local features and complement the Haar-like features used in this context. Notice that the amount of basis vectors is related with the number of the clusters in the K-Means algorithm used in extraction stage and we do not necessarily get better results using more clusters as shown in our results. With this idea, it is possible to control the computational cost of our method using an optimum clusters number. Looking inside the results we can concluded that only 1000 clusters are necessary to obtained a better precision respect to VJD. It is important to notice for FDDB database the miss rate decreases considerably when the amount of basis vectors is less than 1000 as shown Figure 4.2.d

Table 4.1: Precision obtained by Viola and Jones detector for each database. We used the implementation of VJD available in OpenCV Library with scale factor 1.05 and taking into account ROIs larger than 3×3.

Database	**Scale factor**	**TP average size**	**Precision**
BIOid	1.05	36×36	72%
CK+	1.05	72×72	47%
FERET dvd1	1.05	60×60	65%
FERET dvd2	1.05	72×72	70%
FDDB	1.05	44×44	43%

Table 4.2: Precision obtained by the proposed method compared with Viola and Jones results. Nro. depicts the optimum value to amount of basis vector, λ depicts the optimum value to λ and V&J represent the results obtained by Viola and Jones detector.

	Our method			**V&J**
Database	*Nro.*	λ	Precision	Precision
BIOid	1500	10.55	0.98	0.72
CK+	1500	10.55	0.94	0.47
FERET dvd1	1500	10.55	0.84	0.65
FERET dvd2	1500	10.55	0.90	0.70
FDDB	1500	10.55	0.65	0.43

which support our results. Detailed results of the Figure 4.2 can be seen in Table 4.3.

On the second part of the experiments, it is taken into account the trends found by analyzing the variation in the amount of basis vectors in all databases. It took 1500 as optimum amount of basis vectors for all database and were analyzed the variations of λ used in the Gaussian kernel to construct the kernel descriptors. From the experimental results, it can be seen in Figure 4.3.a that it is obtained a precision over 90% for the database BioID, CK+ and FERET dvd2. These results improve considerably the obtained by VJD approximately more than 33%. On the other hand, 65% of precision is obtained for FDDB database, 22% more than the results obtained by VJD. This result together with 90% of recall obtained for the same database, as shown Figure 4.3.b, significantly improve results in Table 4.2.

Finally, notice that for BIOid, FERET DVD1, FERET DVD2 and FDDB database it is obtained an optimum miss rate when $\lambda < 10$. This behavior could be analyzed with the results of the first experiments for each database in order to try to reduce computational cost contained at the time of construction of the basis vectors. Detailed results of the Figure 4.2 can be seen in Table 4.4.

Our results shown that the proposed method significantly improves the results obtained by the Viola and Jones detector and reduce the amount of false positives generated by it. In Figure 4.4 it is shown samples of the false positives discarded by out detector using the efficient kernels

Table 4.3: Results obtained in all of databases when the amount of the basis vector be varying. Nro. is the amount of basis vector.

CK+ database					**BIOid database**				
Nro.	**Precision**	**Recall**	**Accuracy**	**Miss rate**	**Nro.**	**Precision**	**Recall**	**Accuracy**	**Miss rate**
10	0.92027	0.87049	0.89617	0.12950	10	0.94942	0.93867	0.90199	0.06132
100	0.91924	0.87704	0.89867	0.12295	100	0.97200	0.95000	0.93189	0.05000
500	0.94064	0.85737	0.90033	0.14262	500	0.97984	0.96320	0.95016	0.03679
1000	0.94227	0.82950	0.88787	0.17049	1000	0.98176	0.96509	0.95348	0.03490
1500	0.94029	0.82622	0.88538	0.17377	1500	0.98180	0.96698	0.95514	0.03301
3000	0.94413	0.83114	0.88953	0.16885	3000	0.98284	0.97264	0.96096	0.02735
4000	0.94413	0.83114	0.88953	0.16885	4000	0.98379	0.97358	0.96262	0.02641
5000	0.94328	0.81803	0.88289	0.18196	5000	0.98471	0.97264	0.96262	0.02735
FERET dvd1 database					**FERET dvd2 database**				
Nro.	**Precision**	**Recall**	**Accuracy**	**Miss rate**	**Nro.**	**Precision**	**Recall**	**Accuracy**	**Miss rate**
10	0.74082	0.88288	0.72439	0.11711	10	0.84990	0.96067	0.84551	0.03932
100	0.81455	0.89317	0.79933	0.10682	100	0.88126	0.96741	0.87956	0.03258
500	0.83493	0.89189	0.81598	0.10810	500	0.89719	0.97078	0.89617	0.02921
1000	0.84261	0.89575	0.82431	0.10424	1000	0.90406	0.97415	0.90448	0.02584
1500	0.84140	0.89446	0.82264	0.10553	1500	0.90510	0.97528	0.90614	0.02471
3000	0.84963	0.89446	0.82930	0.10553	3000	0.90794	0.97528	0.90863	0.02471
4000	0.84737	0.89317	0.82681	0.10682	4000	0.90794	0.97528	0.90863	0.02471
5000	0.84822	0.89189	0.82681	0.10810	5000	0.91167	0.97415	0.91112	0.02584
FDDB database									
Nro.	**Precision**	**Recall**	**Accuracy**	**Miss rate**					
10	0.48140	0.69295	0.67421	0.30704					
100	0.61403	0.88732	0.79268	0.11267					
500	0.61583	0.89859	0.79529	0.10140					
1000	0.65050	0.90704	0.82055	0.09295					
1500	0.65580	0.90704	0.82404	0.09295					
3000	0.65979	0.90140	0.82578	0.09859					
4000	0.65567	0.89577	0.82229	0.10422					
5000	0.66244	0.88450	0.82491	0.11549					

Table 4.4: Results obtained in all of databases when λ be varying.

CK+ database

Lambda	Precision	Recall	Accuracy	Miss rate
0.35	–	–	–	–
0.55	–	–	–	–
1.55	0.88599	0.89180	0.88704	0.10819
5.55	0.92181	0.86963	0.89889	0.13036
10.55	0.94413	0.83114	0.88953	0.16885
20.55	0.96303	0.76885	0.86794	0.23114
30.55	0.97435	0.68524	0.83139	0.31475

BIOid database

Lambda	Precision	Recall	Accuracy	Miss rate
0.35	0.95944	0.98207	0.94767	0.01792
0.55	0.96107	0.97830	0.94601	0.02169
1.55	0.97003	0.97735	0.95348	0.02264
5.55	0.98009	0.97547	0.96096	0.02452
10.55	0.98284	0.97264	0.96096	0.02735
20.55	0.98635	0.95471	0.94850	0.04528
30.55	0.96966	0.96509	0.94269	0.03490

FERET dvd1 database

Lambda	Precision	Recall	Accuracy	Miss rate
0.35	–	–	–	–
0.55	0.79152	0.91377	0.78850	0.08622
1.55	0.81105	0.90604	0.80266	0.09395
5.55	0.84476	0.90347	0.83014	0.09652
10.55	0.84963	0.89446	0.82930	0.10553
20.55	0.85180	0.88030	0.82348	0.11969
30.55	0.81199	0.90604	0.80349	0.09395

FERET dvd2 database

Lambda	Precision	Recall	Accuracy	Miss rate
0.35	–	–	–	–
0.55	0.86007	0.98764	0.87209	0.01235
1.55	0.87400	0.98988	0.88704	0.01011
5.55	0.90299	0.98314	0.90946	0.01685
10.55	0.90794	0.97528	0.90863	0.02471
20.55	0.90261	0.96853	0.89950	0.03146
30.55	0.89386	0.96516	0.88953	0.03483

FDDB database

Lambda	Precision	Recall	Accuracy	Miss rate
0.35	0.57685	0.85633	0.76132	0.14366
0.55	0.59392	0.88169	0.77700	0.11830
1.55	0.64040	0.89295	0.81184	0.10704
5.55	0.64621	0.89014	0.81533	0.10985
10.55	0.65979	0.90140	0.82578	0.09859
20.55	0.63793	0.83380	0.80226	0.16619
30.55	0.65171	0.69577	0.79094	0.30422

descriptors.

In Figures 4.5 and 4.6 we show some samples of the results obtained using a application developed to test our detector. It is possible to see that our detector improve significantly the results of VJD in unconstrained environments.

4.4 Conclusion

Many well known databases were used to test our efficient eye detector and its results were compared respect the results obtained through Viola and Jones detector. As it was explained in previous chapters kernel features were extracted of each ROI to construct a kernel descriptor that complement the eye variations represented by Haar-like features. This was shown through of the two types of experiments performed. First, the variation of the amount of basis vectors was analyzed. Later, the values of λ were changed such that it was found optimum parameters in the model for each databases.

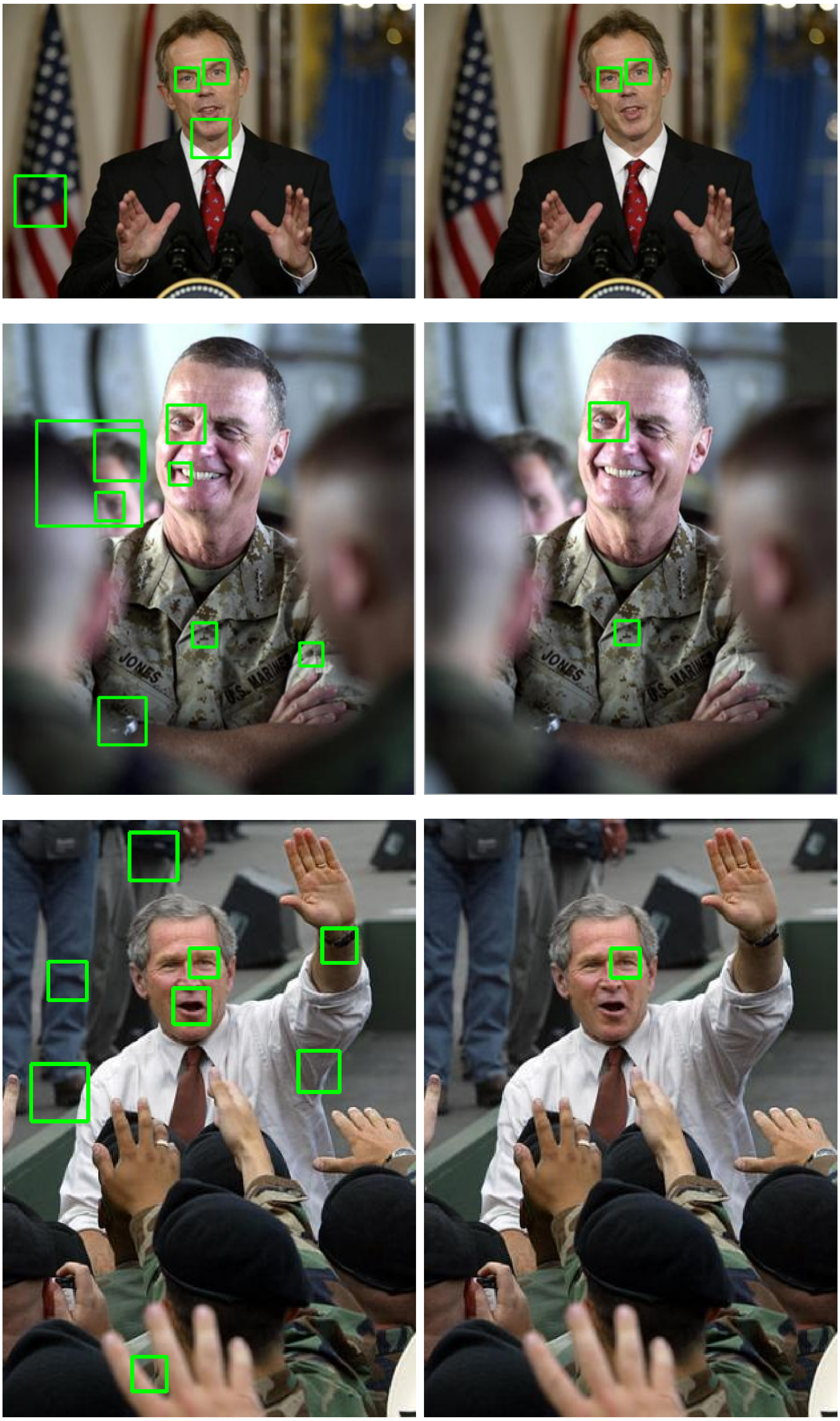


Figure 4.5: Viola and Jones detector vs. EMK detector.

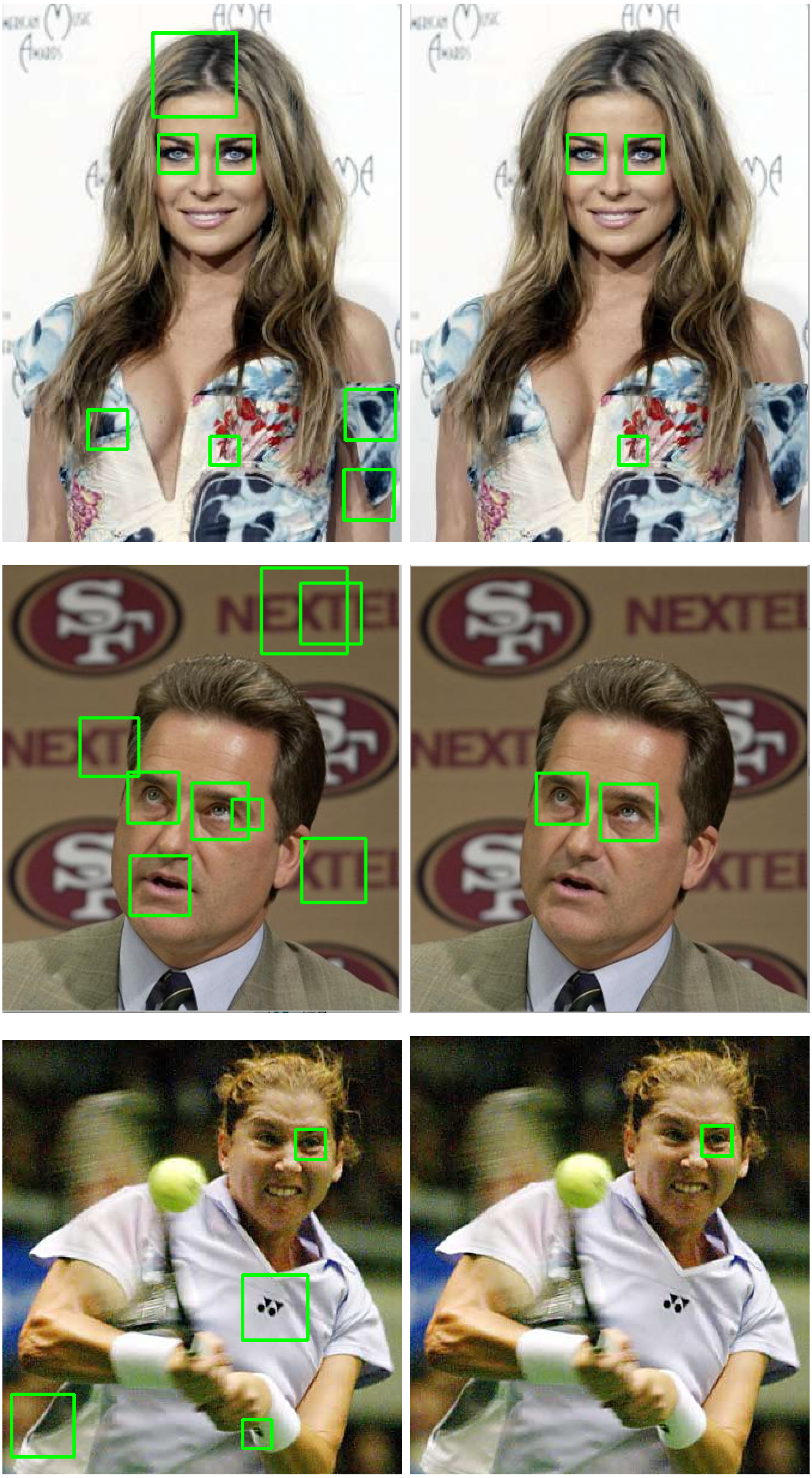


Figure 4.6: Viola and Jones detector vs. EMK detector.

With our method were obtained better precision than in the case of Viola and Jones detector only increasing the response time in approximately 0.0044 seconds using an Laptop with Intel(R) Core(TM) i7-4712HQ CPU @ 2.30GHz and 16G of RAM. Taking into account that the speed is one of the main advantages of this detector we can conclude that the proposed detector can be used in detection systems for security and video surveillance.

In Chapter 5 we will present our final observation about the work and will detailed possible futures lines of investigation related with kernel theory used for eye detection and object detection systems in general.

Chapter 5

Conclusions

We described an eye detection method that works in unrestrained settings and improves detection in various image databases. A map approximation was constructed from HOG features to a Hilbert subspace providing new features called efficient kernel descriptors.

The eye detector proposed improve the results of the well known Viola and Jones detector for eyes using kernel descriptors of local features, with at least 19% (FERET dvd1) more precision in the worst case, and 47% for best case (CK+).

These descriptors are projections of mapped feature vectors on a Hilbert space of high dimension and are built based in a finite set of basis features vectors and operations with a Gaussian kernel. We performed these kernel descriptors to improve the obtained results by Viola and Jones detector for eyes and we obtained a significant reduction in false positives without relevant reduction of true positives. That shows the kernel features take into account low level features that complement to which ones were detected through Viola and Jones framework without adding computational cost during all the detection process. As we have focused only on the detector outputs of Viola and Jones, the regions of interest are reduced. This reduces the spectrum of failed features found in the projection of the eye in an image.

In this chapter we detail the main contributions of the proposed detector and the future lines of investigation that may be exploited when the kernel theory is performed in object detection problems. Finally, we conclude this chapter mentioned details about the paper generated by this work and has been accepted at an international conference.

5.1 Main contributions

The efficient eye detector proposed in this work is an integral object detector that allow us to discriminate the false positives generated by the Viola and Jones detector. An implementation of the proposed detector was designed, developed and tested, which allowed us to achieve the objectives presented in Section 1.2:

1. It was shown that the kernel features are a good complement for Haar features in object

detection tasks. Specifically in the problem of eye detection in uncontrolled environments we show that kernel descriptors can be used to obtained a better precision than the Viola and Jones detector without increasing the time for each detection.

2. Based on HOG features an algorithm for extract and train kernel features was designed and developed. This algorithm can be easily used in other object recognition tasks.

3. An application to train and test our eye detector was developed. This application was made under C/C++ language and can be used to prove our detector with any database.

4. Automatic generation of indicators such as precision, accuracy, recall and miss rate through same application.

5.2 Future lines of investigation

For future work we will investigate other issues such that:

1. Low-level features, such as binary forms or color space and then move them to a larger dimension space for the detection.

2. Consider the spatial data in each local descriptor. Because each eye local feature has an specific position in an image.

3. Consider latent variable inside SVM model. This can increase the computational cost of the eye detector but improve the precision too.

Then, other appearance models in visual recognition could also be approached using our proposed framework.

5.3 Publications

The work in this manuscript allowed the development, directly, of the following paper accepted for presentation at conferences:

- Eye Detection in Unrestrained Settings using Efficient Match Kernels and SVM Classification [Benavides and Borges 2016].

Bibliography

[Aronszajn 1950]ARONSZAJN, N. Theory of reproducing kernels. *Trans. of the American Mathematical Society*, v. 68, n. 3, p. 337–404, May 1950.

[Awais, Badruddin and Drieberg 2013]AWAIS, M.; BADRUDDIN, N.; DRIEBERG, M. Automated eye blink detection and tracking using template matching. *IEEE Student Conf. on Research and Development (SCOReD)*, p. 16–17, December 2013.

[Barber 2012]BARBER, D. *Bayesian Reasoning and Machine Learning*. 1. ed. [S.l.]: Cambridge University Press, 2012.

[Benavides and Borges 2016]BENAVIDES, D.; BORGES, D. Eye detection in unrestrained settings using efficient match kernels and SVM classification. *IEEE Int. Conf. on Systems, Man, and Cybernetics (SMC)*, 2016.

[Bo, Ren and Fox 2010]BO, L.; REN, X.; FOX, D. Kernel descriptors for visual recognition. *Advances in Neural Information*, p. 1–9, 2010.

[Bo and Sminchisescu 2009]BO, L.; SMINCHISESCU, C. Efficient match kernel between sets of features for visual recognition. In: BENGIO, Y. et al. (Ed.). *Advances in Neural Information Processing Systems 22*. [S.l.: s.n.], 2009. p. 135–143.

[Bradski 2000]BRADSKI, G. The OpenCV library. *Dr. Dobb's Journal of Software Tools*, 2000.

[Chen et al. 2014]CHEN, Y.-L. et al. Real-time eye detection and event identification for human-computer interactive control for driver assistance. *IEEE Int. Conf. on Systems, Man, and Cybernetics (SMC)*, p. 2144–2149, 2014.

[Choi, Han and Kim 2011]CHOI, I.; HAN, S.; KIM, D. Eye detection and eye blink detection using AdaBoost learning and grouping. *Int. Conf. on Computer Communications and Networks (ICCCN)*, 2011.

[Cover 1965]COVER, T. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE Trans. on Electronic Computers*, v. 14, n. 3, p. 326–334, June 1965.

[Dalal and Triggs 2005]DALAL, N.; TRIGGS, B. Histograms of oriented gradients for human detection. *IEEE Computer Vision and Pattern Recognition Conf. (CVPR)*, v. 1, p. 886–893, June 2005.

[Felzenszwalb et al. 2010]FELZENSZWALB, P. et al. Object detection with discriminatively trained part-based models. *IEEE Trans. on PAMI*, v. 32, n. 9, p. 1627–1645, September 2010.

[Felzenszwalb and McAllester 2007]FELZENSZWALB, P.; MCALLESTER, D. The generalized A* architecture. *Journal of Artificial Intelligence Research*, v. 29, p. 153–190, 2007.

[Felzenszwalb and Huttenlocher 2005]FELZENSZWALB, P. F.; HUTTENLOCHER, D. P. Pictorial structures for object recognition. *International Journal of Computer Vision*, v. 61, p. 55–79, 2005.

[Fischler and Elschlager 1973]FISCHLER, M.; ELSCHLAGER, R. The representation and matching of pictorial structures. *IEEE Transactions on Computers*, C-22, p. 67–92, 1973.

[Freund and Schapire 1997]FREUND, Y.; SCHAPIRE, R. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, v. 55, p. 119–139, 1997.

[Froba and Ernst 2004]FROBA, B.; ERNST, A. Face detection with the modified census transform. *IEEE Int. Conf. Automatic Face and Gesture Recognition*, p. 91–96, May 2004.

[Hamel 2009]HAMEL, L. *Knowledge Discovery with Support Vector Machines*. 1. ed. [S.l.]: Wiley-Interscience, 2009.

[Hong et al. 2015]HONG, C. et al. Multi-view ensemble manifold regularization for 3D object recognition. *Information Sciences*, v. 320, p. 395–405, 2015.

[Hyunjun, Jinsu and Jaihie 2014]HYUNJUN, K.; JINSU, K.; JAIHIE, K. A pose adaptive eye detection method using 3D face information. *2014 Int. Conf. on Electronics, Information and Communications (ICEIC)*, p. 1–2, January 2014.

[Jain and Learned-Miller 2010]JAIN, V.; LEARNED-MILLER, E. *FDDB: A Benchmark for Face Detection in Unconstrained Settings*. [S.l.], 2010.

[James et al. 2013]JAMES, G. et al. *An Introduction to Statistical Learning*. 1. ed. [S.l.]: Springer-Verlag New York, 2013. (1431-875X, XIV).

[Jesorsky, Kirchberg and Frischholz 2001]JESORSKY, O.; KIRCHBERG, K.; FRISCHHOLZ, R. Robust face detection using the Hausdorff distance. *Int. Conf. on Audio and Video based Biometric Person Authentication*, p. 90–95, June 2001.

[Jin and Geman 2006]JIN, Y.; GEMAN, S. Context and hierarchy in a probabilistic image model. *IEEE Conference on Computer Vision and Pattern Recognition*, v. 2, p. 2145–2152, 2006.

[Kumar, Raja and Ramakrishnan 2002]KUMAR, R.; RAJA, S.; RAMAKRISHNAN, A. Eye detection using color cues and projection functions. *2002 Int. Conf. on Image Processing*, v. 3, p. III–337 – III–340 vol.3, 2002.

[Lowe 2004]LOWE, D. G. Distinctive image features from scale-invariant keypoints. *Int. Journal of Computer Vision*, v. 60, p. 91–110, November 2004.

[Lucey et al. 2010]LUCEY, P. et al. The extended Cohn-Kanade dataset (CK+): A complete dataset for action unit and emotion-specified expression. *IEEE Conf. on Computer Vision and Pattern Recognition - Workshops, (CVPRW)*, p. 94–101, July 2010.

[Mercer 1909]MERCER, J. Functions of positive and negative type, and their connection with the theory of integral equations. *Philosophical Trans. of the Royal Society of London*, v. 209, p. 415–446, 1909.

[Mitchell 1997]MITCHELL, T. M. *Machine Learning*. 1. ed. [S.l.]: McGraw-Hill Science/Engineering/Math, 1997.

[Oliveira et al. 2012]OLIVEIRA, L. S. et al. A fast eye localization and verification method to improve face matching in surveillance videos. *IEEE Int. Conf. on Systems, Man, and Cybernetics (SMC)*, p. 840–845, October 2012.

[Pandey and Lazebnik 2011]PANDEY, M.; LAZEBNIK, S. Scene recognition and weakly supervised object localization with deformable part-based models. *IEEE Int. Conf. on Computer Vision (ICCV)*, p. 1307–1314, 2011.

[Pedersoli and Leuven 2014]PEDERSOLI, M.; LEUVEN, K. A. A scalable 3D HOG model for fast object detection and viewpoint estimation. *Int. Conf. on 3D Vision (3DV)*, v. 1, p. 163–170, 2014.

[Peressini, Sullivan and Uhl 1988]PERESSINI, A.; SULLIVAN, F.; UHL, J. *The Mathematics of Nonlinear Programming*. 1. ed. [S.l.]: Springer-Verlag New York, 1988.

[Phillips et al. 2000]PHILLIPS, P. et al. The FERET evaluation methodology for face-recognition algorithms. *IEEE Trans. on PAMI*, v. 22, n. 10, p. 1090–1104, 2000.

[Rosenblatt 1958]ROSENBLATT, F. The Perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, p. 65–386, 1958.

[Shuo and Chengjun 2011]SHUO, C.; CHENGJUN, L. Fast eye detection using different color spaces. *IEEE Int. Conf. on Systems, Man, and Cybernetics (SMC)*, p. 521–526, Oct. 2011.

[Vapnik 1992]VAPNIK, V. Principles of risk minimization for learning theory. In: MOODY, J. E.; HANSON, S. J.; LIPPMANN, R. P. (Ed.). *Advances in Neural Information Processing Systems 4*. Morgan-Kaufmann, 1992. p. 831–838. Available from Internet: <http://papers.nips.cc/paper/506-principles-of-risk-minimization-for-learning-theory.pdf>.

[Vapnik, Boser and Guyon 1992]VAPNIK, V.; BOSER, B.; GUYON, I. A training algorithm for optimal classifier. *In Proc. 5th ACM Workshop on Computational Learning Theory.*, p. 144–152, 1992.

[Vapnik and Chervonenkis 1971]VAPNIK, V.; CHERVONENKIS, A. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, v. 16, n. 2, p. 264–280, 1971.

[Vapnik and Cortes 1995]VAPNIK, V.; CORTES, C. Support-vector networks. *Machine Learning*, v. 20, n. 3, p. 273–297, September 1995.

[Viola and Jones 2004]VIOLA, P.; JONES, M. J. Robust real-time face detection. *Int. Journal of Computer Vision*, v. 2, n. 57, p. 137–154, 2004.

[Wu and Ji 2014]WU, Y.; JI, Q. Learning the deep features for eye detection in uncontrolled conditions. *Int. Conf. on Pattern Recognition (ICPR)*, p. 455–459, 2014.

[Yang, Kriegman and Ahuja 2002]YANG, M.-H.; KRIEGMAN, D. J.; AHUJA, N. Detecting faces in images: A survey. *IEEE Trans. on PAMI*, v. 24, n. 1, p. 34–58, jan 2002.

[Yu and Joachims 2009]YU, C.; JOACHIMS, T. Learning structural SVMs with latent variables. *Int. Conf. on Machine Learning*, p. 1–8, 2009.

[Yu et al. 2014]YU, J. et al. High-order distance-based multiview stochastic learning in image classification. *IEEE Trans. on Cybernetics*, v. 44, p. 2431–2442, 2014.

[Zhu and Mumfordt 2007]ZHU, S.; MUMFORDT, D. A stochastic grammar of images. *Foundations and Trends in Computer Graphics and Vision*, v. 2, n. 4, p. 259–362, 2007.

Appendix

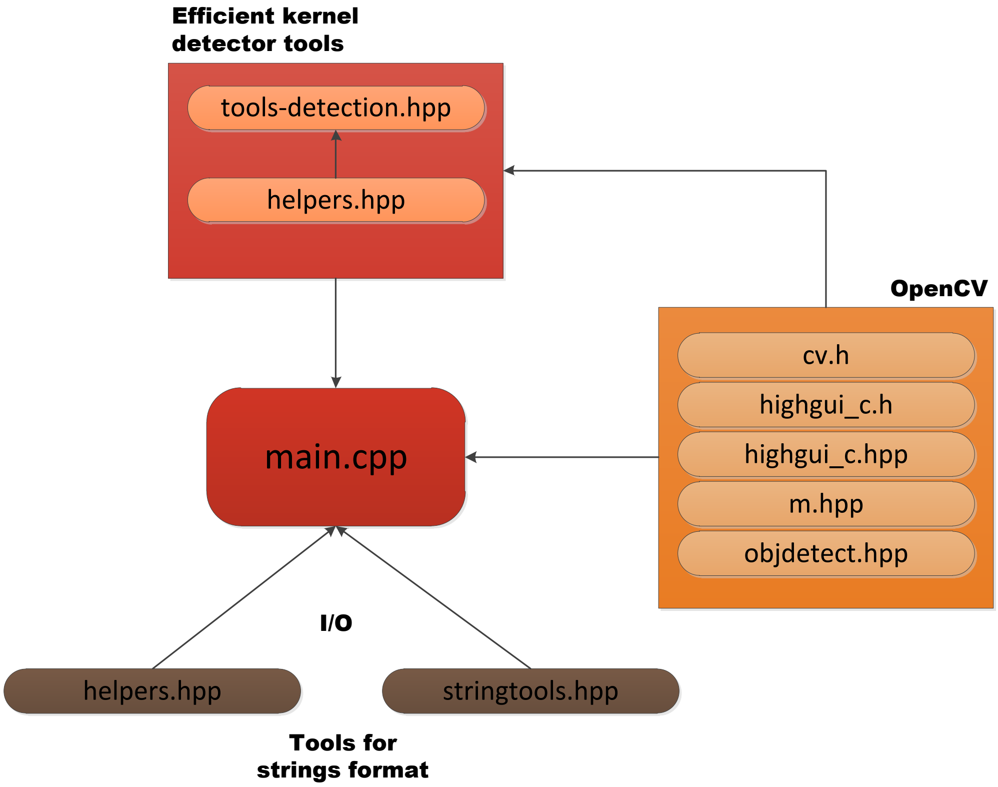


Figure I.1: Libraries architecture.

I. ARCHITECTURE OF THE PROGRAM

We programed some libraries in C/C++ Language for the efficient kernel eye detector , and the architecture of the software works as follows (Figure I.1):

- The main.cpp formats the names (inputs/outputs) and list files for each database for each stage detector and obtained the amount of each database (extraction database, training database and classification database) through helpers.cpp.
- The main.cpp obtain label information (t:true positive or f:false positive) of each sample file on training data through stringtools.cpp.
- The main.cpp construct the efficient kernel descriptors through tools_deteccion.cpp.
- The main.cpp use the OpenCV libraries to manipulate the image data and perform operations with matrix.

-The objdetect.hpp is used to obtain the Viola and Jones detector outputs and extract the HOG features to construct efficient kernel descriptor.

-The core.hpp is used to obtain the basis set through K-Means algorithm.

-The ml.hpp is used to training and test a linear SVM.